



HAL
open science

Reconstruction d'image : régularisation avec prise en compte des discontinuités.

Pierre Charbonnier

► **To cite this version:**

Pierre Charbonnier. Reconstruction d'image : régularisation avec prise en compte des discontinuités. : Thèse de doctorat. Traitement du signal et de l'image [eess.SP]. Université de Nice - Sophia Antipolis, 1994. Français. NNT : 1994NICE4763 . tel-01183977

HAL Id: tel-01183977

<https://hal.science/tel-01183977>

Submitted on 12 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Présentée à l'Université de Nice-Sophia Antipolis

Ecole Doctorale Sciences Pour l'Ingénieur

pour obtenir le titre de

Docteur en Sciences

mention

Sciences de l'ingénieur

par

Pierre CHARBONNIER

Reconstruction d'image : régularisation avec prise en compte des discontinuités.

Soutenue le 30 Septembre 1994 devant le Jury

<i>M. Jean-Marc CHASSERY</i>	Directeur de Recherche au C.N.R.S.	Président, rapporteur
<i>M. Henri MAITRE</i>	Professeur à l'E.N.S.T. Paris	Rapporteur
<i>M. Michel BARLAUD</i>	Professeur à l'U.N.S.A.	Directeur de thèse
<i>M. Gilles AUBERT</i>	Professeur à l'U.N.S.A.	Membre
<i>Mme Laure BLANC-FERAUD</i>	Chargé de Recherche au C.N.R.S.	Membre
<i>M. Pierre GRANGEAT</i>	Chef de projet au C.E.A/D.T.A. L.E.T.I.	Membre
<i>Mme Josiane ZERUBIA</i>	Chargé de Recherche à l'I.N.R.I.A.	Membre

THÈSE

Présentée à l'Université de Nice-Sophia Antipolis

Ecole Doctorale Sciences Pour l'Ingénieur

pour obtenir le titre de

Docteur en Sciences

mention

Sciences de l'ingénieur

par

Pierre CHARBONNIER

Reconstruction d'image : régularisation avec prise en compte des discontinuités.

Soutenue le 30 Septembre 1994 devant le Jury

<i>M. Jean-Marc CHASSERY</i>	Directeur de Recherche au C.N.R.S.	Président, rapporteur
<i>M. Henri MAITRE</i>	Professeur à l'E.N.S.T. Paris	Rapporteur
<i>M. Michel BARLAUD</i>	Professeur à l'U.N.S.A.	Directeur de thèse
<i>M. Gilles AUBERT</i>	Professeur à l'U.N.S.A.	Membre
<i>Mme Laure BLANC-FERAUD</i>	Chargé de Recherche au C.N.R.S.	Membre
<i>M. Pierre GRANGEAT</i>	Chef de projet au C.E.A/D.T.A. L.E.T.I.	Membre
<i>Mme Josiane ZERUBIA</i>	Chargé de Recherche à l'I.N.R.I.A.	Membre

A ma famille,

à mes amis,

à la mémoire de Robert Praud et d'André Bodin.

Cette thèse a été financée par une bourse de docteur ingénieur du GDR 134 Traitement du Signal et Image du CNRS. Je tiens à témoigner ma gratitude aux dirigeants du GDR pour la confiance qu'ils m'ont témoignée en m'octroyant cette bourse.

Les travaux présentés ici ont été menés d'Octobre 1991 à Septembre 1994 au sein du laboratoire I3S, URA 1376 du CNRS, Université de Nice-Sophia Antipolis. Je remercie son directeur, Mr Jean-Claude Bermond pour son accueil.

Je tiens à remercier Mr Jean-Marc Chassery, Directeur de Recherche au CNRS et directeur adjoint du GDR 134 pour l'attention qu'il porte à mes recherches. Je lui sais gré d'avoir accepté d'être rapporteur de cette thèse et de me faire l'honneur d'en présider le jury.

Je suis extrêmement reconnaissant envers Mr Henri Maître, Professeur à l'Ecole Nationale Supérieure des Télécommunications, pour l'intérêt qu'il a porté à mes travaux. Je lui sais gré d'avoir bien voulu être rapporteur de cette thèse.

Ces travaux ont été effectués sous la direction de Mr Michel Barlaud, Professeur à l'Université de Nice-Sophia Antipolis. Je tiens à le remercier pour son encadrement attentif, pour sa disponibilité et pour ses conseils précieux. Il a su me placer dans de très bonnes conditions de travail, en me faisant bénéficier de structures efficaces et de collaborations fructueuses. Je lui en suis très reconnaissant et le remercie pour son dévouement et son esprit d'entreprise.

Je veux ici exprimer ma sincère gratitude à Mme Laure Blanc-Féraud pour le soutien de tous les instants qu'elle m'a apporté au cours de ces trois années. Je n'oublierai jamais avec quelle rigueur scientifique elle a relu ce mémoire, ni la patience et le tact qu'elle a déployés à cette occasion. Je n'oublierai pas non plus l'aide qu'elle m'a apportée pendant certaines périodes difficiles. Merci "Mame Blanc".

Je remercie Mr Gilles Aubert, Professeur à l'Université de Nice-Sophia Antipolis avec qui nous avons beaucoup collaboré en ce qui concerne l'étude théorique des modèles et algorithmes présentés ici. J'ai particulièrement apprécié sa compétence scientifique et sa disponibilité. Aussi, c'est une joie pour moi que de le compter parmi les membres de mon jury.

Je remercie Mr Pierre Grangeat, chef du projet "Imagerie 3D" au CEA/DTA LETI, de m'avoir fait l'honneur de faire partie de mon jury de thèse.

Je remercie Mme Josiane Zérubia d'avoir accepté d'être membre du jury. Je la remercie également pour l'attention toute particulière et le temps qu'elle a consacrés à la lecture de ce mémoire, ainsi que pour ses conseils avisés à ce sujet.

Les travaux menés au sein de l'équipe IMAGE sur la reconstruction tomographique font depuis trois ans l'objet d'une collaboration particulièrement fructueuse avec l'équipe Pr. Bussière du Centre

Antoine Lacassagne du CHR de Nice. Je tiens spécialement à témoigner ma gratitude au Pr Jacques Darcourt en qui j'ai toujours trouvé un interlocuteur attentif, compétent et disponible. Merci également à Octave Migneco, Jacques Magné et José Benoliel pour leur sympathique accueil et pour avoir consacré une part de leur précieux temps à l'acquisition des données réelles qui nous étaient nécessaires.

Je remercie Mr Donald Geman, spécialiste de renommée internationale, de s'être intéressé à ces recherches et d'avoir accepté de donner un avis sur mes travaux.

Je tiens à remercier Mr Jean Céa, dont j'ai eu la chance d'être l'élève à l'Ecole Supérieure en Sciences Informatiques. Les discussions que nous avons eu avec lui au commencement de ces travaux ont été des plus profitables.

Je veux ici à exprimer ma reconnaissance et mon amitié à Pierre Malick “#8” Koulibaly et Ivan Laurette, depuis trois ans compagnons de cette “aventure tomographique”. Les moments partagés dans le travail et en dehors resteront d'excellents souvenirs. Je tiens également à remercier Jean-Marc Collin et Jean-Michel Bruneau, avec qui j'ai eu le plaisir de collaborer au début de cette thèse.

J'adresse un remerciement tout spécial à Pierre Lobel et Valérie Barraud pour l'aide que m'a apporté leur implantation des routines “sparse”. Je tiens également à associer à ces remerciements tous les autres stagiaires qui ont travaillé sur la reconstruction tomographique (Jean-Philippe Darmon, Virginie Amar, Jean-Luc Bernon, François Delbary, Vincent Garrouste, Franck Guyomar, Deirdre McGrath) et tous ceux et celles qui, d'une manière ou d'une autre, ont participé à l'élaboration de ce mémoire.

Il m'est particulièrement agréable de remercier tous les membres de l'équipe IMAGE non déjà cités: Marc Antonini, Patricia Gauthier, Pierre Mathieu, Thierry Gaidon, Dimitri Lebedeff, Jean-Marie Moureaux, Bruno Rouchouze. J'ai beaucoup apprécié l'ambiance de franche camaraderie qui règne dans cette équipe. J'ai eu beaucoup de plaisir à y travailler.

Je ne voudrais pas oublier de remercier le personnel administratif et technique de l'IS3, en particulier: Danielle, Micheline, Corinne, Kathy, Monique, Maurice, Lionel et Guy, pour l'aide qu'ils m'ont apportée tout au long de ces trois ans.

Je tiens à témoigner ma gratitude à tous les amis que je n'ai pas déjà cités: aux “montagnards” (Doulaye, Olivier, Jean-Louis, Monsieur Bataillou, Olivier, Sylvie...), aux membres de la “cafet” (Valérie, Patrick, Jean-François, Erol, Samuel, Valeria, Mario...), aux basketteurs (les “highlights”, Fédé et les gars des Mines, Manue et Clo...) et au Groupe d'Assise, pour les moments inoubliables qu'ils m'ont fait passer. Ces remerciements amicaux s'adressent aussi à Régis et Sophie, Yves et Sandrine, Ben, et à mes amis de La Gaubretière...

Merci enfin à ma famille et à toutes les personnes qui, par leur soutien moral et leurs encouragements, m'ont aidé pendant ces années de thèse.

Sommaire

Notations	1
INTRODUCTION	5

Chapitre I

Reconstruction d'image et régularisation	11
1. Introduction	11
2. Régularisation des problèmes inverses mal-posés: approche Bayésienne	12
2.1. Reconstruction Bayésienne.....	12
2.2. Estimation Bayésienne	13
2.3. Remarque.....	14
3. Modèle de formation des données.....	14
3.1. Modélisation des transformations déterministes	14
3.2. Modélisation des phénomènes aléatoires	15
4. Modèle a priori: utilisation des champs de Markov.....	16
4.1. Propriété de Markov	16
4.2. Théorème de Hammersley-Clifford: du local au global	17
4.3. Modèle utilisé	19
5. Optimisation.....	19
5.1. Algorithmes stochastiques.....	20
5.1.1. Analogie avec le processus de recuit.....	20
5.1.2. Recuit avec dynamique de Metropolis	21
5.1.3. Échantillonneur de Gibbs avec recuit.....	23
5.1.4. Conclusion.....	24
5.2. Algorithmes déterministes.....	25
5.2.1. Modes conditionnels itérés (ICM).....	25
5.2.2. Non Convexité Graduelle (GNC).....	26
5.2.3. Recuit par Champ Moyen (MFA)	27

6. Conclusion 27

Chapitre II

**Régularisation avec prise en compte des discontinuités, régularisation semi-
quadratique 29**

1. Introduction 29

2. Fonctions de potentiel et prise en compte des discontinuités 30

2.1. Modèle quadratique 30

2.2. Modèle avec processus de ligne 31

2.2.1 Processus de ligne Booléen 31

2.2.2 Processus de ligne Booléen implicite 35

2.2.3 Processus de ligne continu..... 36

2.2.4 Conclusion..... 37

2.3. Autres approches 38

2.3.1 Fonction d’Hebert et Leahy 38

2.3.2 Potentiels convexes 39

2.3.3 Les conditions de Lange 42

2.3.4 Conclusion..... 42

2.4. Etude heuristique des équations normales..... 43

2.4.1 Equations normales 44

2.4.2 Etude de cas: lissage et prise en compte des discontinuités 44

2.4.3 Conclusion..... 45

2.5. Application aux fonctions de potentiel précitées..... 46

2.5.1 Fonctions de pondération correspondantes 46

2.5.2 Fonctions de pondération utilisées 47

2.5.3 Qualité de la préservation des contours..... 48

2.5.4 Conclusion..... 49

2.6. Régularisation et prise en compte des discontinuités: conclusion..... 50

3. Régularisation semi-quadratique..... 51

3.1. Introduction 51

3.1.1 Utilisation du processus de ligne Booléen pour l’estimation du MAP 51

3.1.2 Utilisation de variables auxiliaires réelles pour l’estimation du MAP..... 52

3.1.3 Principe de la régularisation semi-quadratique 54

3.2. Définition de la variable auxiliaire 54

3.2.1 Première approche: “théorème de Geman & Reynolds étendu”..... 55

3.2.2 Deuxième approche: utilisation des paires de Legendre	57
3.2.3 Définition des énergies duales.....	58
3.3. Propriétés de la régularisation semi-quadratique.....	59
3.3.1 Propriété à b fixé	59
3.3.2 Propriété à f fixé	59
4. Conclusion	60

Chapitre III

Deux algorithmes déterministes de relaxation:.....	63
1. Introduction	63
2. Forme générale des algorithmes ARTUR et LEGEND	64
2.1. Stratégie de minimisation.....	64
2.2. Présentation des algorithmes	65
2.2.1. Expressions littérales de b^{n+1} à f^{n+1} fixé	65
2.2.2. Calcul de f^{n+1} à b^{n+1} fixé	66
2.2.3. Forme générale des algorithmes	67
2.3. Remarques sur les algorithmes.....	68
2.3.1. De l'aspect auto-adaptatif des algorithmes.....	68
2.3.2. De l'introduction progressive des discontinuités	69
3. Convergence des algorithmes ARTUR et LEGEND	70
3.1. Justification de la stratégie utilisée.....	70
3.2. Résultats de convergence	72
3.3. Perspectives: cas des potentiels non-convexe	73
4. Implantation des algorithmes ARTUR et LEGEND.....	73
4.1. Résolution des équations normales	73
4.1.1. Introduction	73
4.1.2. Cas de l'algorithme ARTUR.....	74
4.1.3. Cas de l'algorithme LEGEND.....	76
4.2. Paramétrisation du modèle	77
4.2.1. Harmonisation des fonctions de pondération	77
4.2.2. Paramètres des programmes	79
4.3. Utilisation des huit voisins	81
4.4. Tests d'arrêt.....	82
4.5. Conditions aux bords.....	82

4.5.1. Cas d'ARTUR	83
4.5.2. Cas de LEGEND	84
5. Comparaisons avec d'autres algorithmes.....	84
5.1. ARTUR et le GNC	84
5.1.1. Description rapide de l'algorithme GNC	84
5.1.2. ARTUR et le GNC	86
5.2. ARTUR et le MFA	87
5.2.1. Description rapide de l'algorithme MFA	87
5.2.2. ARTUR et le MFA	90
5.3. Potentiels variables et utilisation de variable auxiliaire	91
5.4. ARTUR et le MAP-EM OSL de Green.....	93
5.4.1. Algorithme ML-EM	93
5.4.2. Algorithme MAP-EM-OSL.....	94
5.4.3. ARTUR et le MAP-EM-OSL.....	95
5.4.4. Perspective: ARTUR dans le cas de la vraisemblance Poissonienne.....	96
6. Perspectives: autres stratégies possibles d'optimisation.....	98
6.1. Introduction	98
6.2. Schémas stationnaires.....	98
6.2.1. Schéma stationnaire implicite	98
6.2.2. Schéma stationnaire explicite.....	99
6.3. Schémas dynamiques.....	100
6.3.1. Schéma dynamique implicite	100
6.3.2. Schéma dynamique explicite.....	100
6.4. Conclusion.....	101
7. Conclusion	102

Chapitre IV

Application à la reconstruction tomographique	105
1. Introduction	105
2. La tomographie par émission de photon unique (SPECT).....	105
2.1. Modélisation du problème.....	105
2.1.1. Modélisation analytique de la formation des projections.....	106
2.1.2. Modélisation discrète de la formation des projections	107
2.1.3. Modèle de matrice de Radon utilisée	109

2.1.4. Quantification de la matrice de Radon	110
2.2. Reconstruction tomographique.....	112
2.2.1. Aspect mal-posé du problème de reconstruction.....	112
2.2.2. Effet de la quantification sur le conditionnement de la matrice	115
3. Description des objets utilisés pour les tests	115
3.1. Le fantôme numérique Nice	115
3.2. Le fantôme réel de Jaszczak	116
3.3. Les données réelles.....	117
4. Applications des algorithmes à la reconstruction tomographique	118
4.1. Etude de l'algorithme ARTUR.....	118
4.1.1. Fonctionnement de l'algorithme.....	118
4.1.2. Réglage des paramètres	127
4.1.3. Utilisation de différents potentiels	129
4.1.4. Influence de la condition initiale	133
4.1.5. Influence du voisinage utilisé.....	136
4.1.6. Application à des données réelles.....	137
4.2. Etude de l'algorithme LEGEND	139
4.2.1. Marche de l'algorithme	139
4.2.2. Influence du seuil d'arrêt.....	142
4.2.3. Utilisation d'un potentiel non convexe	143
4.2.4. Factorisation de Cholesky	143
4.3. Autres algorithmes.....	144
4.3.1. Moise.....	144
4.3.2. Artur explicite.....	145
6. Conclusion	146

Chapitre V

Résolution dans le domaine de la transformée en ondelettes 2D.....	149
1. Introduction	149
2. Présentation des transformées en ondelettes BCR et 2D	150
2.2. Transformée en ondelettes multirésolution	150
2.3. Représentation matricielle: transformée BCR.....	152
2.3.1. Transformée d'un vecteur.....	152
2.3.2. Transformée d'une matrice.....	153

2.4. Application aux images: transformée “2D”	155
2.4.1. Transformée d’un vecteur.....	155
2.4.2. Transformée d’une matrice.....	156
3. Résolution de systèmes linéaires de grande taille dans le domaine de la TO	157
3.1. Introduction	157
3.2. Exemple: cas de la régularisation de Tikhonov	158
3.3. Perspective: application aux algorithmes ARTUR et LEGEND	163
3.3.1. Cas d’ARTUR	163
3.3.2. Cas de LEGEND	165
4. Conclusion, perspectives de travail.....	166
CONCLUSION, PERSPECTIVES	167

Annexes

Annexe A: Calcul des équations normales associées à la minimisation du critère du MAP	174
Annexe B: Preuve du théorème 2 “de Geman et Reynolds étendu”	176
Annexe C: Preuve du théorème 3 “Paires de Legendre”	179
Annexe D: Preuve du théorème 4.....	184
Annexe E: Convergence d’ARTUR.....	190
Références Bibliographiques	197
Liste des figures et tableaux.....	205

Notations

Nous avons essayé d'utiliser dans ce mémoire des notations simples. Nous donnons ici les principales conventions et notations adoptées. Les notations peu usitées seront introduites dans le texte.

Représentation des images: conventions

Modèle déterministe:

- Dans ce cas, on considère une image comme une matrice dans laquelle sont rangées les valeurs des pixels. La plupart du temps, on considère une image carrée, de dimension $N \times N$:

$$f = [f_{i,j}] \text{ pour } i = 0 \cdots N-1 \text{ et } j = 0 \cdots N-1$$

L'indice i représente le numéro de ligne du pixel, et j son numéro de colonne.

- Nous aurons souvent à considérer l'image comme un vecteur de dimension $N^2 \times 1$:

$$f = [f_k] \text{ pour } k = 0 \cdots N^2 - 1$$

- Le passage de la représentation matricielle à la représentation vectorielle se fait de façon simple. On utilise l'ordonnancement lexicographique ("lexicographic ordering"). On écrit dans le vecteur les valeurs des pixels, dans l'ordre où on les lit dans la matrice (comme on lirait un texte: de gauche à droite et de bas en haut). Par exemple dans le cas 2x3:

$$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix} \text{devient} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

Le passage inverse se fait en écrivant dans la matrice les valeurs lues dans le vecteur (comme on écrit un texte).

- Nous utiliserons indifféremment la représentation matricielle ou la représentation vectorielle, sachant qu'il ne s'agit en fait que d'une transformation d'indice définie par:

$$\begin{cases} k = i.N + j \\ i = k / N \\ j = k \% N \end{cases}$$

Dans cette expression, / représente la division entière et % le reste de la division entière.

Modèle stochastique:

- Dans ce type de modèle, on considère l'image comme une valeur d'un champ aléatoire \mathcal{F} défini sur un réseau régulier de sites, S :

$$\mathcal{F} = \{f_s, s \in S\}$$

Chaque site correspond à l'emplacement d'un pixel et peut être repéré par ses coordonnées (i,j). Dans cette approche, f désigne la valeur de \mathcal{F} , et appartient à l'ensemble des configurations, Ω . Pour simplifier, on pose:

$$P(f) = P(\mathcal{F} = f)$$

- Le symbole f_s note la variable aléatoire associée à un site $s \in S$ donné. On note f_s la valeur du pixel, et on pose:

$$P(f_s) = P(\mathcal{F}_s = f_s).$$

Processus itératifs:

- Dans le cadre d'un algorithme itératif, nous noterons f^n l'estimée à l'itération n .
- Dans le cadre d'un algorithme itératif par étapes, nous noterons $f^{n,m}$ l'estimée à

l'itération n de l'étape m .

Principales notations employées

Divers

$\delta(i, j)$ Fonction indicatrice: $\delta(i, j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$.

Vecteurs-matrices

$f = [f_k]$ Vecteur colonne.

$f^t = [f_k^t]$ Vecteur ligne.

$\langle f, g \rangle$ Produit scalaire de f par g : $\langle f, g \rangle = f^t g = \sum_k f_k g_k$.

$\|f\|$ Norme de f : $\|f\|^2 = \langle f, f \rangle$.

RMS (abréviation de "Root Mean Square" en anglais). Racine de la moyenne des coefficients du vecteur au carré: $RMS(f) = \sqrt{\frac{1}{N^2} \sum_k (f_k)^2} = \frac{1}{N} \|f\|$

$A = [A_{i,j}]$ Matrice: i numero de ligne, j numéro de colonne de l'élément $A_{i,j}$.

A^t Transposée de A : $\left[(A^t)_{i,j} \right] = [A_{j,i}]$.

$\text{diag}\{a_k\}$ Matrice diagonale dont les éléments sont les coefficients a_k .

I Matrice identité: $I_{i,j} = \delta(i, j)$.

D_x Opérateur de gradient dans la direction horizontale: $(D_x f)_k = (D_x f)_{i,j} = f_{i,j} - f_{i,j-1}$.

D_y Opérateur de gradient dans la direction verticale: $(D_y f)_k = (D_y f)_{i,j} = f_{i,j} - f_{i-1,j}$.

Δ Laplacien discret: $(\Delta f)_k = (\Delta f)_{i,j} = (f_{i-1,j} - 2f_{i,j} + f_{i+1,j}) + (f_{i,j-1} - 2f_{i,j} + f_{i,j+1})$.

Cet opérateur correspond à la convolution par le filtre: $\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$.

- ⊗ Produit de Kronecker: si A est de taille $N_1 \times N_2$ et B de taille $M_1 \times M_2$ alors $A \otimes B$ est la matrice de taille $N_1 M_1 \times N_2 M_2$ définie par:

$$A \otimes B = [A_{i,j} B] = \begin{bmatrix} A_{0,0} B & \cdots & A_{0,N_2-1} B \\ \vdots & \ddots & \vdots \\ A_{N_1-1,0} B & \cdots & A_{N_1-1,N_2-1} B \end{bmatrix}$$

Mesures de qualité

SNR (abréviation de “Signal-to-Noise Ratio” en anglais). Rapport signal à bruit. Nous utilisons généralement le rapport des variances. Le SNR en variance entre une image g et une image de référence f est défini par:

$$10 \log_{10} \left[\frac{\sigma_f^2}{\sigma_{f-g}^2} \right]$$

où σ_f^2 est la variance de l’image de référence et σ_{f-g}^2 est la variance de l’image d’erreur $f - g$.

NMSE (abréviation de “Normalized Mean Square Error” en anglais). Erreur relative entre deux images. L’erreur relative entre g et l’image de référence f est définie par:

$$\frac{\|f - g\|^2}{\|f\|^2}$$

Notation des nombres réels

Pour les nombres réels, nous utiliserons la notation en virgule flottante: <mantisse>e<exposant>.

Ex : 1.2e2 = 120

INTRODUCTION

Cette thèse est consacrée à la *reconstruction d'image*. La reconstruction d'image numérique consiste, à partir de données mesurées par un capteur, à reconstruire une représentation d'un objet ou d'une scène.

Le mot "image" est ici à prendre au sens général de *représentation numérique* de l'objet. Il peut s'agir d'un champ d'intensité, analogue numérique de la photographie ou du film. Il peut également s'agir d'informations de nature très diverse. Citons par exemple la stéréovision, utilisée en robotique ou en imagerie satellitaire pour le calcul de modèles numériques de terrain (MNT). Dans ce dernier cas, les données sont constituées de deux photographies satellites, et il s'agit de reconstruire la carte des altitudes en chaque point de l'image. On peut également citer l'exemple de la détection du mouvement d'un objet à partir d'une séquence temporelle d'images de cet objet. Citons enfin l'exemple de la reconstruction tomographique qui consiste, à partir d'une série d'images collectées autour du patient par une caméra sensible aux rayonnements Gamma, de reconstruire des coupes d'organes. C'est l'application que nous avons plus particulièrement traitée au cours de cette thèse.

- On le voit sur ces exemples, les situations sont diverses. Les applications se différencient par la nature de l'objet étudié, du phénomène physique mesuré et des capteurs utilisés, de la géométrie du problème, de l'image à reconstruire, etc. Pour reconstruire l'image d'un objet à partir des données observées, il est tout d'abord nécessaire d'avoir un *modèle de la formation des données* à partir de l'objet. Ceci fait intervenir deux types de phénomènes: les phénomènes déterministes et les phénomènes aléatoires.

Les phénomènes déterministes sont liés à la géométrie du problème, à certains défauts connus du système imageur, etc. Ils se modélisent en général assez bien par des techniques d'algèbre linéaire ou des convolutions. Par exemple, en tomographie, la formation des données

mesurées à partir de l'objet fait intervenir une projection intégrale, qui se représente par une multiplication matrice-vecteur. Autre exemple, lorsque le miroir du télescope spatial Hubble présentait un défaut, il fournissait des images floues. On pouvait représenter la formation de ces données par une opération de convolution entre l'image originale des étoiles et une certaine fonction de transfert.

Les phénomènes aléatoires sont liés à la part de hasard (le "bruit") intervenant dans la formation des données. Ces phénomènes ne peuvent se modéliser que de manière statistique. Le modèle le plus classique est celui de la loi Gaussienne, mais on en utilise parfois d'autres, en fonction du problème considéré. Par exemple, lorsque le nombre de photons détectés est très faible (comme en astronomie ou en tomographie à très faible taux de comptage), une modélisation Poissonienne s'avère théoriquement plus adaptée.

- Une fois modélisée la formation des données, la reconstruction d'image devrait simplement consister à inverser ce processus. C'est en fait rarement possible de par la nature même des problèmes rencontrés et la présence de bruit. On parle alors de problème inverse mal posé. Il est nécessaire d'introduire une contrainte, un *a priori*, sur le type de solution recherchée: c'est la *régularisation*. Dans le cadre d'une approche probabiliste (Bayésienne) de la reconstruction, un très bon outil est la modélisation par champs Markoviens. Elle offre un cadre mathématique élégant, permettant de modéliser un grand nombre de contraintes, à travers la spécification de fonctions appelées *fonctions de potentiel*.

- Ayant défini le modèle de formation des données et le modèle d'image à reconstruire, il reste à mettre au point un "*bon*" *algorithme*. Un "bon" algorithme est un algorithme qui fournit, de la façon la plus rapide possible, une image conforme au modèle utilisé. La définition d'un tel algorithme est une tâche d'autant plus délicate que le modèle d'image à reconstruire est perfectionné. En effet, on peut alors se trouver en face d'un problème d'optimisation de fonctionnelles non quadratiques et pouvant être non convexes. Le modèle de formation des données utilisé a également une grande influence sur la définition de l'algorithme. Suivant le type de bruit considéré, les calculs seront plus ou moins complexes. Par exemple, une modélisation Gaussienne donnera des algorithmes beaucoup plus simples qu'une modélisation Poissonienne. D'autre part, certains modèles imposent la manipulation de matrices de très grandes tailles. C'est par exemple le cas en tomographie. Le volume d'information mis en jeu représente alors une difficulté très importante. Enfin, il peut être intéressant de définir une stratégie assez générale, permettant d'adapter l'algorithme à divers problèmes de reconstruction.

Pour définir un algorithme de reconstruction d'image, il existe deux approches. La première,

stochastique, est la plus satisfaisante du point de vue optimisation. Toutefois, le temps de calcul nécessaire est en général très important. De plus, ces méthodes sont peu adaptées à des problèmes tels que la reconstruction tomographique. La seconde approche, déterministe, est sous-optimale, mais permet de reconstruire des images de très bonne qualité en un temps beaucoup moins long. De plus, ces méthodes sont en général adaptables à un grand nombre d'applications.

• En résumé, pour reconstruire des images de bonne qualité à partir des données, il est nécessaire d'avoir défini:

- un "bon" modèle de formation des données,
- un "bon" modèle d'image à reconstruire,
- un "bon" algorithme de reconstruction.

Dans le cadre de cette thèse, nous utilisons des modèles "standard" de formation des données (le modèle de bruit Gaussien et le modèle Poissonien) et nous consacrons surtout aux deux derniers points.

En ce qui concerne la définition d'un modèle d'image satisfaisant, nous étudions particulièrement les modèles permettant la préservation des discontinuités, qui sont des éléments très importants pour la lecture de l'image.

Pour le dernier point, nos principales préoccupations sont d'obtenir la meilleure qualité d'image possible, de réduire au maximum le temps de calcul nécessaire, et de définir une stratégie adaptable à une grande variété d'applications. Pour cela, nous adoptons une stratégie déterministe car elle nous paraît offrir un bon moyen d'atteindre ces objectifs.

Ce mémoire est organisé de la manière suivante:

• Dans le premier chapitre, nous introduisons la notion de reconstruction d'image régularisée, dans le cadre mathématique de l'estimation Bayésienne. Nous évoquons les problèmes d'optimisation intervenant lorsqu'on veut prendre en compte les discontinuités et donnons une description rapide des principaux algorithmes existants.

• Au second chapitre, nous réfléchissons à la façon de construire un modèle de régularisation avec prise en compte des discontinuités. Dans ce cas, l'*a priori* posé sur l'image à reconstruire consiste à la considérer formée de zones homogènes, séparées par des bords francs. Quel type de fonction de potentiel faut-il utiliser pour assurer la préservation des discontinuités? On trouve principalement deux approches. La première travaille directement sur la forme de la fonction de

potentiel [HEBE89] [BESA86] [BOUM93] [GREE90A] [LANG90]. La seconde [GEMA84] [GEMA92] introduit une variable, appelée *processus de ligne*, permettant de marquer les discontinuités afin de les préserver du lissage. Les conditions imposées aux fonctions de potentiel ne sont pas les mêmes dans les deux cas. Nous proposons dans ce chapitre des conditions de prise en compte des discontinuités permettant d'unifier ces deux approches.

Dans la seconde partie du chapitre, nous présentons la régularisation *semi-quadratique* [GEMA93]. Ce type de régularisation repose sur l'utilisation d'une variable auxiliaire. Cette variable auxiliaire interagit avec la variable principale au niveau de la fonction de potentiel, soit de façon multiplicative comme dans [GEMA92] soit de façon additive [GEMA93]. La variable auxiliaire permet la prise en compte des discontinuités tout en offrant un moyen de simplifier le problème d'optimisation évoqué Chapitre I. En effet, la fonctionnelle *duale*, à deux variables, que l'on doit alors minimiser est *quadratique en la variable principale* lorsque l'on fixe la variable auxiliaire, et *convexe en la variable auxiliaire* lorsqu'on fixe la variable principale. Nous donnons de plus un résultat original important: l'expression de la variable auxiliaire minimisant la fonctionnelle duale, lorsqu'on fixe la variable principale.

- Dans le troisième chapitre, nous exploitons les propriétés de la régularisation semi-quadratique ainsi que l'expression littérale de la variable auxiliaire mise en évidence au Chapitre III. En utilisant une stratégie de *minimisations alternées* en chacune des variables, nous définissons deux algorithmes déterministes de reconstruction d'image avec prise en compte des discontinuités: ARTUR, dans le cas où la variable auxiliaire interagit de façon multiplicative avec la variable principale, et LEGEND, dans le cas d'une interaction additive. Ces algorithmes sont des algorithmes à *introduction progressive de discontinuités*. Nous proposons dans ce chapitre une étude des propriétés de convergence de ces algorithmes. Nous les situons par rapport aux principales approches déterministes utilisées: l'algorithme de non convexité graduelle ("Graduated Non Convexity", ou GNC, en anglais) et le recuit par champ moyen ("Mean Field Annealing", ou MFA, en anglais).

Les algorithmes ARTUR et LEGEND sont basés sur un modèle de bruit Gaussien. A titre de perspective, nous proposons une adaptation au cas où le modèle d'attache aux données n'est plus Gaussien mais Poissonien. Enfin, toujours à titre de perspective, nous passons en revue d'autres stratégies possibles de minimisation.

- L'application des algorithmes ARTUR et LEGEND à la tomographie d'émission 2D fait l'objet du quatrième chapitre. Après une description du problème de la reconstruction tomographique, nous proposons des résultats expérimentaux (sur données simulées et sur données

réelles) permettant de mieux comprendre le comportement et les propriétés des algorithmes.

- Dans certaines applications, en particulier en reconstruction tomographique, l'utilisation de méthodes algébriques de reconstruction telles qu'ARTUR et LEGEND, est problématique. En effet, elles impliquent la résolution de systèmes linéaires de très grandes tailles. Afin d'envisager des dimensions d'images plus importantes et le passage au cas 3D, il est indispensable de réduire la quantité d'information nécessaire aux algorithmes de reconstruction. Les matrices impliquées n'ont pas forcément de bonnes propriétés vis-à-vis de la transformée de Fourier. Au chapitre V, nous présentons une méthode, basée sur l'utilisation de la Transformée en Ondelettes, permettant de "creuser" ces matrices, en y introduisant un nombre important de zéros. Par le biais d'une algorithmique particulière, il est alors possible de diminuer le volume utile des matrices et d'accélérer la reconstruction, tout en gardant le contrôle de la qualité des images obtenues. A titre de perspective, nous étudions au Chapitre V l'adaptation des algorithmes ARTUR et LEGEND à ce type de technique.

Chapitre I

Reconstruction d'image et régularisation

1. Introduction

En traitement d'image, l'image d'origine (notée f) est reliée à l'image observée p , par une transformation qui fait intervenir deux types de phénomènes. Les premiers sont déterministes: projection intégrale (comme en tomographie), éventuels défauts du système imageur (dégradation par un flou, un bougé...). Les seconds, aléatoires, ne sont connus qu'en termes de statistiques, comme par exemple le bruit de mesure.

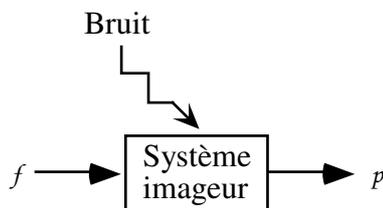


Figure 1-1: De l'image réelle à l'image observée

Le terme *problème inverse* désigne le problème consistant à "inverser" cette transformation, autrement dit: retrouver l'image de départ à partir de l'image observée. En fait, ce n'est pas toujours possible en pratique. En effet, la plupart des problèmes inverses en traitement d'images sont par nature *mal posés*.

Un problème inverse est dit mal posé (au sens d'Hadamard) si l'une au moins des trois conditions suivantes, n'est pas respectée:

“il existe une solution au problème, cette solution est unique et dépend continûment des données”.

C'est souvent la troisième condition qui pose problème: la solution ne dépend pas continûment des données et le processus de reconstruction est très sensible au bruit. La seule connaissance des données observées est insuffisante pour assurer l'existence, l'unicité et la stabilité d'une solution. Il est nécessaire d'introduire une contrainte supplémentaire, un *a priori* sur la solution. C'est la *régularisation*.

La théorie de la régularisation a été largement investiguée et a fourni une littérature abondante. Certains articles, comme [TITT85], [POGG85] ou [DEMO89], permettent d'aborder le sujet en synthétisant les différentes approches possibles de la régularisation.

Pour notre part, nous avons choisi d'introduire la régularisation dans le cadre de la reconstruction Bayésienne. Dans ce chapitre, nous introduisons les principales notions associées à la reconstruction Bayésienne et évoquons les problèmes d'optimisation qui interviennent.

2. Régularisation des problèmes inverses mal-posés: approche Bayésienne

2.1. Reconstruction Bayésienne

Une approche Bayésienne suppose que l'on soit capable d'établir la distribution de l'image de départ conditionnée par l'observation, $P(f|p)$, appelée distribution *a posteriori*. Pour cela, on utilise la formule de Bayes:

$$P(f|p) = \frac{P(p|f) P(f)}{P(p)} \quad (1-1)$$

Dans l'équation (1-1), p étant une donnée, $P(p)$ joue le rôle de constante de normalisation.

La distribution $P(p|f)$ mesure la vraisemblance des données par rapport à l'image originale (“likelihood” en anglais). Elle exprime la connaissance que nous avons du modèle de formation des données. Nous étudierons au paragraphe 3 les deux façons les plus courantes de spécifier cette distribution.

La distribution de probabilité *a priori*, $P(f)$, exprime les hypothèses faites sur la solution afin de régulariser le problème. Cette distribution est étudiée au paragraphe 4 et dans le Chapitre II.

2.2. Estimation Bayésienne

Supposant établie la distribution conditionnelle *a posteriori*, il nous faut maintenant choisir un estimateur statistique \hat{f} de l'image originale f . Cela peut se faire de diverses manières. A chaque estimateur, on peut associer une fonction de coût $c(e, f)$ qui mesure la qualité de l'estimation e par rapport à la solution exacte. L'estimateur est alors donné par:

$$\hat{f} = \arg \min_e \int c(e, f) P(f|p) df \quad (1-2)$$

Dans nos travaux, nous utilisons l'estimateur du Maximum a Posteriori (MAP). Dans ce cas, la fonction de coût associée est: $c(e, f) = 1 - \delta(e, f)$, et l'estimateur :

$$\hat{f} = \arg \max_f P(f|p) \quad (1-3)$$

L'estimateur du MAP peut se mettre en œuvre dans la plupart des cas, aussi est-il très utilisé en traitement d'images. De plus, il permet de faire le lien avec les approches déterministes de la régularisation ([BRUN93], [DEMO89]).

Nous remarquons que si l'on suppose que f a une distribution uniforme (toutes les configurations possibles ont la même probabilité), alors l'estimateur du MAP rejoint l'estimateur du maximum de vraisemblance ("Maximum Likelihood" ou ML en anglais):

$$\hat{f} = \arg \max_f P(p|f) \quad (1-4)$$

L'*a priori* uniforme est complètement neutre: il ne sert à rien vis-à-vis de la régularisation. Nous citons l'estimateur du maximum de vraisemblance car il est parfois utilisé (en particulier en imagerie médicale). Dans ce cas, si la méthode d'optimisation est itérative, la régularisation est simplement effectuée en arrêtant le processus "avant que les effets du bruits ne se fassent trop sentir", ce qui est très subjectif !

Il existe d'autres estimateurs, qui peuvent s'avérer être plus adaptés à certains problèmes (cf. [BOUT92]). Ainsi, l'estimateur du Mode de la Marginale a Posteriori (MPM), associé à la fonction de coût $c(e, f) = \sum_{i,j} [1 - \delta(e_{i,j}, f_{i,j})]$

$$\hat{f}_{i,j} = \arg \max_{f_{i,j}} P(f_{i,j}|p) \quad (1-5)$$

peut être utilisé en segmentation et classification. L'estimateur du champ moyen ("Mean

Field” en anglais), associé à la fonction de coût $c(e, f) = \sum_{i,j} (e_{i,j} - f_{i,j})^2$

$$\hat{f} = \int f_{i,j} P(f_{i,j}|p) df_{i,j} \quad (1-6)$$

peut être utilisé notamment en restauration d’images (comme par exemple dans [DINT90]).

2.3. Remarque

Nous notons que maximiser la distribution *a posteriori* est équivalent à minimiser $-\log[P(f|p)]$ (appelé log posterior en anglais), donné par:

$$-\log[P(f|p)] \propto -\log[P(p|f)] - \log[P(f)] \quad (1-7)$$

Dans les paragraphes suivant, nous étudions les différents termes de cette équation, et les problèmes d’optimisation associés à l’estimation au sens du MAP.

3. Modèle de formation des données

3.1. Modélisation des transformations déterministes

En dehors de tout aspect aléatoire, on peut dans bien des cas modéliser la transformation permettant de passer de l’image de départ f à l’image observée p , par un modèle linéaire de la forme suivante:

$$p = Rf \quad (1-8)$$

Dans cette équation, f et p sont mis sous forme de vecteurs ordonnés lexicographiquement. Notons que l’image de départ et l’image observée n’ont pas forcément la même taille. La matrice R n’est donc pas forcément carrée. L’aspect mal posé du problème inverse est lié au mauvais conditionnement de R .

La matrice R modélise la réponse impulsionnelle du système. La réponse impulsionnelle peut être invariante par translation (dans le cas d’une défocalisation, par exemple) et dans ce cas R est bloc-Toeplitz. La réponse impulsionnelle peut également dépendre de la position du pixel dans l’image, comme en tomographie (voir Chapitre IV). Dans ce dernier cas, R modélise une projection intégrale. C’est pourquoi nous emploierons parfois le terme de *projections* pour désigner p .

Il reste maintenant à introduire les phénomènes aléatoires intervenant dans la formation de

l'image observée.

3.2. Modélisation des phénomènes aléatoires

En tomographie comme en imagerie spatiale, on considère que les émissions du pixel k de l'image originale sont complètement aléatoires (cf. [GREE90A]). Le flot des photons venant de k et captés par l'élément l du détecteur¹ forme donc un processus de Poisson. Comme les émissions sont supposées indépendantes et que chaque photon est capté par au plus un dixel, ces processus sont indépendants pour tout k et tout l . Par propriété de superposition des processus de Poisson, il s'en suit que les données forment un champ de variables aléatoires ayant une distribution de Poisson et sont indépendantes entre elles:

$$p_l \approx \text{Poisson} \left(\sum_k R_{lk} f_k \right) \quad (1-9)$$

Dans ce cas, la distribution des données conditionnellement à l'image originale est donnée par:

$$P(p|f) = \prod_l \frac{\left(\sum_k R_{l,k} f_k \right)^{p_l} \exp\left(-\sum_k R_{l,k} f_k\right)}{p_l!} \quad (1-10)$$

Ce qui nous donne l'expression suivante pour la log-vraisemblance:

$$J_1(f) = -\log[P(p|f)] = \sum_l \left\{ \sum_k R_{l,k} f_k - p_l \log\left(\sum_k R_{l,k} f_k \right) \right\} + Cte \quad (1-11)$$

Cette expression n'est pas facile à exploiter. On introduit classiquement un champ de variables non-observées [DEMP77], les données complètes, afin de simplifier le problème de minimisation. Cela conduit à la famille des algorithmes EM ("Expectation-Maximization" en anglais, voir Chapitre III pour plus de détails). Pour l'estimateur du maximum de vraisemblance, c'est l'algorithme appelé ML-EM dans le domaine médical et algorithme de Richardson-Lucy en astronomie. Pour l'estimateur du MAP, c'est la famille des algorithmes MAP-EM.

Dans notre cas, nous utiliserons un modèle plus simple, même s'il est moins réaliste dans le cas de la tomographie. Il s'agit de supposer que les projections (1-8) sont corrompues par un bruit

¹ Par la suite, pour éviter toute confusion, nous désignerons un élément de détecteur par le terme "dixel", (contraction de "detector element").

additif. Ce bruit est considéré comme blanc, Gaussien et indépendant des données:

$$p = Rf + \eta \quad (1-12)$$

L'avantage de ce modèle est double. D'une part, il est très général. Les algorithmes développés dans ce cadre pourront donc être utilisés pour résoudre des problèmes très variés allant de la reconstruction tomographique à la restauration d'image. D'autre part, en considérant que la matrice d'autocorrélation du bruit se met sous la forme $\Gamma_{\eta\eta} = \sigma^2 \text{Id}$, l'expression de la log-vraisemblance est très simple:

$$-\log[P(p|f)] = \frac{1}{\sigma^2} \|p - Rf\|^2 + Cte \quad (1-13)$$

Dans le cadre de la reconstruction tomographique, le défaut de ce modèle est qu'il ne représente qu'une approximation du modèle Poissonien. Toutefois, dans les conditions cliniques habituelles, cette approximation est assez proche de la réalité [KOUL94].

On note que l'estimation au sens du ML dans le cas Gaussien revient à calculer une solution du type moindres carrés.

4. Modèle *a priori*: utilisation des champs de Markov

Afin de spécifier la distribution *a priori* nous utiliserons un modèle d'image Markovien. Ce type de modèle est basé sur une analogie avec la physique statistique. Les pixels de l'image sont considérés comme des atomes, arrangés sur un réseau régulier de sites, et pouvant se trouver dans plusieurs états énergétiques (les niveaux de gris). L'état de chaque atome ne dépend (en termes de statistiques) que de l'état de ses voisins. Une image f correspond à une configuration du système. Ce modèle permet de lier la distribution globale $P(f)$ à des attributs locaux de l'image. C'est pourquoi il s'avère particulièrement adapté à la prise en compte des discontinuités, comme nous le verrons au Chapitre II.

Nous donnons dans ce paragraphe quelques rappels sur les champs Markoviens (pour plus de détails, nous renvoyons le lecteur à l'article fondateur de S. et D. Geman [GEMA84]), et décrivons le modèle *a priori* que nous avons utilisé.

4.1. Propriété de Markov

Dans cette approche, nous considérons l'image f comme un champ aléatoire défini sur un ensemble de sites, noté S . Sur cet ensemble de sites, on définit un système de voisinage

$\mathcal{V} = \{\mathcal{V}_s, s \in S\}$, devant vérifier les deux conditions suivantes:

$$\begin{aligned} i) \quad & \forall s \in S, \quad s \notin \mathcal{V}_s \\ ii) \quad & \forall s, t \in S, \quad s \in \mathcal{V}_s \Leftrightarrow t \in \mathcal{V}_s \end{aligned} \tag{1-14}$$

Nous représentons sur la figure 1-2 les voisinages à quatre et huit voisins, qui sont les plus couramment utilisés.

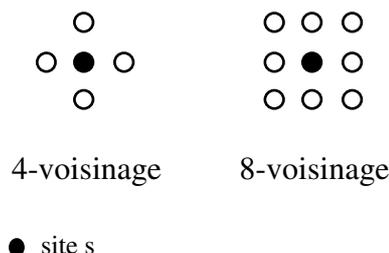


Figure 1-2: Voisinages d'ordre 1 et 2 (d'après [GEMA86])

On dit que f est un champ de Markov par rapport au système de voisinage \mathcal{V} si et seulement si il vérifie, pour tout $s \in S$:

$$\begin{aligned} i) \quad & P(f_s) > 0 \\ ii) \quad & P(f_s | f_t, s \neq t) = P(f_s | f_t, t \in \mathcal{V}_s) \end{aligned} \tag{1-15}$$

La taille du voisinage dépend de l'ordre du champ de Markov considéré (une relation entre ces deux quantités est établie dans [BRUN93]). Ainsi, les 4 et 8-voisinages représentés figure 1-2 correspondent aux champs de Markov d'ordre 1 et 2.

4.2. Théorème de Hammersley-Clifford: du local au global

La propriété (1-15) est très importante: elle permet de faire le lien entre les caractéristiques statistiques locales et globales de l'image. Toutefois, la loi conjointe locale en f_s n'étant pas facile à spécifier, ce résultat n'est pas directement exploitable. Ce n'est pas le cas du théorème suivant, dit "Théorème de Hammersley et Clifford", qui joue un rôle clef dans l'utilisation pratique des champs de Markov.

Le champ f est un champ de Markov par rapport au système de voisinage \mathcal{V} si et seulement si $P(f)$ a une distribution de Gibbs relativement à \mathcal{V} :

$$P(f) = \frac{1}{Z} e^{-J_2(f)} \tag{1-16}$$

où Z est une constante de normalisation (appelée fonction de partition) et l'énergie J_2 est définie comme la somme de fonctions, appelées fonctions de potentiel.

Les fonctions de potentiel sont définies localement sur les cliques correspondant à \mathcal{V} :

$$J_2(f) = \sum_{c \in \mathcal{C}} \varphi_c(f) \quad (1-17)$$

Les cliques associées à un système de voisinage comprennent les sites singletons et tous les sous-ensembles de S formés de sites voisins deux à deux. Les cliques associées aux systèmes de voisinage de la figure 1-2 sont représentées figure 1-3.

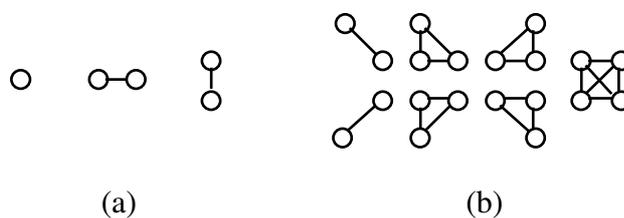


Figure 1-3: Cliques associées aux voisinages d'ordre 1 (a) et d'ordre 2 (a) et (b)

Le théorème de Hammersley et Clifford fournit donc un moyen simple de spécifier les caractéristiques d'un champ de Markov. C'est le choix des cliques utilisées et la forme des fonctions de potentiel qui permet de traduire l'*a priori* imposé à la solution.

Généralement, on écrit l'équation (1-17) sous la forme suivante [GEMA92]:

$$J_2(f) = \sum_{s \in S} w_m \sum_m \varphi \left[\left(D^{(m)} f \right)_s \right] \quad (1-18)$$

Dans l'équation (1-18), $D^{(m)}$ représente l'opérateur différentiel d'ordre m correspondant au type de clique utilisé: l'identité pour les cliques à un site, l'approximation discrète de la dérivée pour les cliques à deux sites, etc...

Le choix des cliques utilisées conditionne directement l'*a priori* imposé à la solution. Ainsi, en utilisant les sites singletons, on travaillera sur la dynamique de la solution. Si les potentiels sont fonctions des différences entre pixels voisins, on imposera une condition sur le gradient de la solution. En utilisant un opérateur différentiel d'ordre 3, on agira sur la dérivée seconde de la solution, etc... L'*a priori* imposé à la solution sera, bien sûr, également traduit dans la forme des fonctions de potentiel. Celle-ci sera discutée au Chapitre II dans le cadre de la régularisation avec

prise en compte des discontinuités.

4.3. Modèle utilisé

Dans nos travaux, nous utilisons des modèles de Markov d'ordre 1 et 2 correspondant à des systèmes à 4 et 8 voisins. Nous posons habituellement une condition sur le gradient de la solution et utilisons donc seulement les cliques à deux pixels de ces modèles (voir figure 1-4).



Figure 1-4: Cliques utilisées pour les voisinages d'ordres 1 (a) et 2 (b)

L'expression de l'énergie *a priori* utilisée dans le cadre de cette thèse est donc de la forme:

$$\begin{aligned}
 J_2(f) = & \sum_{i,j} \varphi(f_{i,j} - f_{i,j-1}) + \sum_{i,j} \varphi(f_{i,j} - f_{i-1,j}) \\
 & + \sum_{i,j} \varphi\left(\frac{f_{i,j} - f_{i-1,j-1}}{\sqrt{2}}\right) + \sum_{i,j} \varphi\left(\frac{f_{i,j} - f_{i+1,j-1}}{\sqrt{2}}\right)
 \end{aligned} \tag{1-19}$$

pour un modèle d'ordre 2. Dans un but de simplification, nous ne développerons par la suite que le cas du modèle d'ordre 1. Dans ce cas, l'expression du logarithme de la distribution *a priori* est:

$$-\log[P(f)] = \frac{1}{Z} J_2(f) + Cte \tag{1-20}$$

avec

$$J_2(f) = \sum_{i,j} \varphi(f_{i,j} - f_{i,j-1}) + \sum_{i,j} \varphi(f_{i,j} - f_{i-1,j}) \tag{1-21}$$

Nous avons maintenant tous les éléments en main pour former l'expression de la distribution *a posteriori*. Dans le paragraphe suivant, nous étudions la maximisation de cette distribution.

5. Optimisation

Comme nous l'avons remarqué au paragraphe 2.3, maximiser la distribution *a posteriori* est équivalent à minimiser $-\log[P(f|p)]$. Compte tenu de ce que nous venons de voir, cette dernière expression peut s'écrire en regroupant les constantes σ^2 et Z :

$$J(f) = J_1(f) + \lambda^2 J_2(f) \quad (1-22)$$

avec (dans le cas de la vraisemblance Gaussienne):

$$J_1(f) = \|p - Rf\|^2$$

$$J_2(f) = \sum_{i,j} \varphi(f_{i,j} - f_{i,j-1}) + \sum_{i,j} \varphi(f_{i,j} - f_{i-1,j}) \quad (1-23)$$

Dans (1-22), $J_1(f)$ est le terme d'attache aux données, $J_2(f)$ est le terme de régularisation ou énergie *a priori*, et $J(f)$ est appelée énergie ou critère du MAP.

La minimisation de $J(f)$ est problématique car la fonction φ peut être non convexe (cf. Chapitre II). En conséquence, le critère du MAP peut admettre des minima locaux. Dans ce cas, un algorithme déterministe de descente a toutes les chances de rester "coincé" dans un tel minimum, et donc de fournir une estimée éloignée de la solution du MAP (correspondant au minimum global de l'énergie du MAP). De plus, le minimum atteint, et donc l'estimée obtenue, dépend de l'initialisation de l'algorithme. Certains auteurs, comme Bouman et Sauer [BOUM93] n'hésitent pas donc à dire que le problème inverse reste mal posé lorsque l'énergie du MAP est non convexe, puisque il n'y a pas unicité de la solution. C'est pourquoi ils plaident pour l'utilisation de fonctionnelles convexes (cf. Chapitre II, paragraphe 1.3.2).

Dans ce paragraphe, nous présentons le principe des algorithmes les plus utilisés en traitement d'image. Ils se classent en deux catégories: les algorithmes stochastiques et les algorithmes déterministes. Les algorithmes stochastiques assurent (en probabilité) la convergence vers le minimum global de l'énergie du MAP, mais ont comme défaut d'être très lents. Les algorithmes déterministes sont plus rapides, mais peuvent rester piégés dans un minimum local du MAP.

5.1. Algorithmes stochastiques

5.1.1. Analogie avec le processus de recuit

Les algorithmes stochastiques d'optimisation sont basés sur une analogie avec le procédé de recuit, utilisé principalement en métallurgie et en verrerie. De façon à obtenir une cristallisation la plus parfaite possible (correspondant à un état moléculaire le plus ordonné possible), on porte le matériau à très haute température, puis on le laisse se refroidir très lentement. Le principe général des algorithmes est le suivant:

Porter le système à une température "assez élevée"

Répéter

| *Calculer un nouvel état du système*

| *Diminuer la température*

Jusqu'à atteindre une température "assez basse"

(1-24)

Il existe deux principales façons de calculer un nouvel état du système à une température donnée, définissant deux algorithmes: l'algorithme de recuit simulé avec dynamique de Metropolis et l'échantillonneur de Gibbs avec recuit.

5.1.2. Recuit avec dynamique de Metropolis

La dynamique de Metropolis [METR53] est la plus proche du processus physique du recuit. Dans ce cas, la variation de température est assez lente pour qu'on puisse considérer qu'à une température donnée, le système a le temps d'évoluer jusqu'à ce qu'il se trouve dans l'état le plus ordonné possible. Cet état correspond au point d'équilibre thermique à cette température. Cet équilibre est caractérisé par le fait que la probabilité pour que le système se trouve dans un état f est:

$$P(\mathcal{F} = f) = \frac{1}{Z} \exp(-\beta J(f)) \quad (1-25)$$

où $Z = \sum_f \exp(-\beta J(f))$ est appelée fonction de partition du système. $J(f)$ est l'énergie du système correspondant à la configuration f . La constante β est définie par:

$$\beta = \frac{1}{KT} \quad (1-26)$$

où K est la constante de Boltzmann et T la température absolue du système. On notera au passage la similitude de (1-25) avec l'équation (1-16).

On peut constater que lorsque la température est élevée ($T \rightarrow +\infty$), tous les états sont équiprobables. En conséquence, une baisse brutale de température risquerait de figer le matériau dans une configuration trop désordonnée (état métastable). Il faut donc faire baisser la température très lentement. On peut également constater que lorsque T tend vers 0, la configuration la plus

probable est celle correspondant à l'énergie minimale. Le matériau sera alors parfaitement cristallisé.

Dans le cas de l'algorithme de simulation du recuit utilisant la dynamique de Metropolis, on considère que la grille des pixels représente les atomes du matériau et les niveaux de gris, leurs états possibles. Une image représente donc une configuration f , à laquelle on associe l'énergie $J(f)$ et une probabilité d'apparition donnée par (1-25). On effectue à chaque température des petites perturbations du système, jusqu'à ce que celui-ci se trouve dans son état d'équilibre "thermique". Le plus souvent, la perturbation consiste à modifier légèrement la valeur d'un pixel en lui ajoutant une valeur aléatoire (appelée grain). Une autre version consiste à échanger la valeur de deux pixels.

La forme générale de l'algorithme suit (1-24). A une température donnée, l'algorithme de recherche du nouvel état du système est le suivant (on note f^k la configuration à l'itération k):

Répéter

| Calculer une configuration perturbée f

| Calculer $d = \exp\{-\beta[J(f) - J(f^k)]\}$

| Si $d > 1$ alors $f^{k+1} = f$

| Sinon

| tirer aléatoirement ξ dans $[0,1]$

| $f^{k+1} = f$ si $\xi \leq d$

| $f^{k+1} = f^k$ si $\xi > d$

Jusqu'à équilibre statistique approché

Si la configuration perturbée est d'énergie inférieure à la précédente, on l'accepte automatiquement. Dans le cas contraire, on l'accepte avec la probabilité d .

Le fait d'accepter des configurations d'énergie supérieure permet de s'affranchir du problème des minima locaux de la fonctionnelle J , comme l'illustre la figure 1-5.

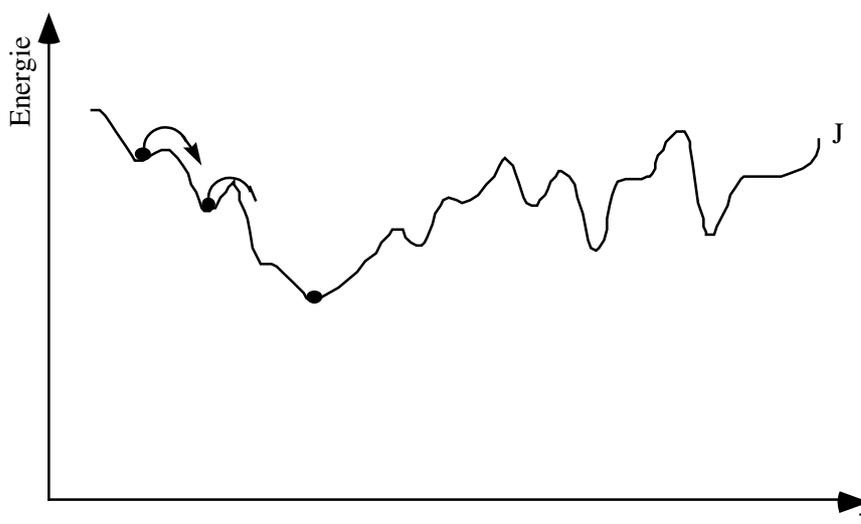


Figure 1-5: La dynamique de Metropolis permet de “sortir” des minima locaux (d’après [BLAKE])

5.1.3. Échantillonneur de Gibbs avec recuit

Dans le cas de l’échantillonneur de Gibbs [GEMA84], l’analogie avec le processus physique de recuit est moins directe. En effet, si l’algorithme général est bien de la forme (1-24), on ne laisse pas forcément le système se stabiliser à chaque température.

Le nouvel état du système est calculé de la façon suivante: on modifie la valeur de niveau de gris en un site s donné, en lui affectant la valeur ξ d’une variable aléatoire X obéissant à la loi conditionnelle suivante:

$$P(X = \xi) = \frac{1}{Z} \exp \left[-\frac{J(f_s = \xi | f_t \ t \neq s)}{T} \right] \quad (1-27)$$

Les sites sont la plupart du temps visités de façon cyclique. En considérant que le terme d’attache aux données $J_1(f)$ peut se mettre sous la forme d’une somme de potentiels locaux, et comme l’image est un champ Markovien, on peut écrire:

$$P(X = \xi) = \frac{1}{Z} \exp \left[-\frac{J(f_s = \xi | f_t \ t \text{ voisin de } s)}{T} \right] \quad (1-28)$$

Écrire $J_1(f)$ sous forme de somme de potentiels locaux suppose que l’opérateur R ait un support de petite taille. Une telle hypothèse est valable pour des applications telles que la restauration d’image. Par contre, ce n’est pas le cas en tomographie. Cela limite donc le champ

d'application de l'algorithme.

Le tirage de ξ s'effectue de la façon suivante. On calcule, pour toutes les valeurs possibles de f_s , l'énergie $J(f_s = \xi | f_t \text{ t voisin de } s)$. On construit ainsi la densité de probabilité (1-28) et la fonction de répartition associée. Il suffit alors de tirer une valeur aléatoire α comprise entre 0 et 1, et d'inverser la fonction de répartition, comme le montre le schéma 1-6.

Remarque: à température fixée, lorsque le nombre d'itérations effectuées tend vers l'infini, l'algorithme trouve un état stable du système, selon la loi (1-25). Cela permet de calculer des échantillons aléatoires (images) selon la loi de Gibbs à une température donnée (d'où le nom du procédé).

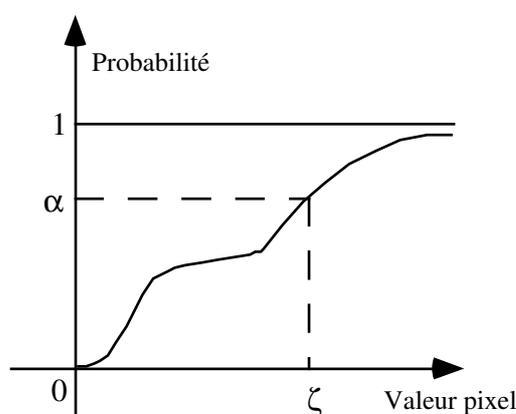


Figure 1-6: Tirage aléatoire selon une loi, par inversion de la fonction de répartition

Dans le cas de l'échantillonneur de Gibbs avec recuit, on fait décroître la température du système au fur et à mesure des itérations. Dans le cas du recuit homogène, on laisse le système se stabiliser avant de modifier la valeur de la température. Dans le cas du recuit inhomogène, on diminue la température après chaque balayage de l'image. Si la variation de température est assez lente, on montre que l'état stable calculé correspond au minimum global de l'énergie du MAP, quel que soit le point de départ [GEMA84].

5.1.4. Conclusion

Les algorithmes de type recuit convergent en probabilité vers un minimum global du critère du MAP, indépendamment de la configuration initiale. Dans le cas où l'énergie du MAP s'écrit comme une somme de termes locaux, on peut utiliser l'échantillonneur de Gibbs avec recuit. Dans le cas contraire, on utilise le recuit simulé avec dynamique de Metropolis.

Ces algorithmes sont très utilisés. De multiples adaptations (ex. [PELL94]) en ont été proposées. Les versions parallèles de l'échantillonneur de Gibbs avec recuit permettent un gain de temps appréciable [GEMA92]. Malgré tout, les algorithmes de recuit demeurent très lourds en coût calcul.

D'autre part, même si la convergence est prouvée en théorie, il n'est pas toujours possible en pratique de se placer dans les bonnes conditions. Par exemple, si l'on veut s'affranchir du problème des minima locaux, il est nécessaire d'adopter une décroissance logarithmique de la température, ce qui ne peut pas toujours être fait en pratique. C'est ce qui fait écrire à Donald Geman, à propos de l'échantillonneur de Gibbs [GEMA92]: "nous n'avons aucune garantie d'obtenir un véritable minimum avec une quantité finie de calculs".

5.2. Algorithmes déterministes

Le critère du MAP est la plupart du temps non convexe. Dans ce cas, les algorithmes déterministes sont tous sous-optimaux, en ce sens qu'ils ne permettent d'atteindre qu'un minimum local de l'énergie du MAP.

Les algorithmes de descente simples du type gradient, gradient conjugué, relaxation ("Successive Over Relaxation" ou SOR en anglais), modes conditionnels itérés ("Iterated Conditional Modes" ou ICM en anglais) risquent de rester piégés dans des minima locaux éloignés du minimum global. Les estimées obtenues risquent donc d'être de mauvaise qualité. Diverses stratégies ont été proposées afin d'obtenir des solutions de bonne qualité. Les plus connues sont le GNC et le MFA.

Nous proposons ici un "survol" des algorithmes ICM, GNC et MFA. Nous reviendrons plus en détail sur les deux dernières au Chapitre III.

5.2.1. Modes conditionnels itérés (ICM)

L'algorithme ICM a été proposé par J. Besag dans [BESA86]. Son principe est le suivant: on parcourt les sites de façon cyclique. A chaque itération, on modifie la valeur en un site uniquement. En chaque site s , on maximise de façon déterministe la probabilité conditionnelle $P(f_s | f_t, t \neq s, p)$. Autrement dit, on minimise le critère du MAP par rapport à f_s uniquement.

On montre qu'à chaque itération, la valeur de la probabilité a posteriori $P(f|p)$ augmente. L'algorithme converge vers un minimum local de l'énergie du MAP. Moyennant une "bonne" initialisation, l'estimée calculée est de bonne qualité. L'avantage de cet algorithme est qu'il est très

général et peut être utilisé dans un grand nombre d'applications. De plus, surtout si le nombre d'états possibles du système est petit (comme par exemple en segmentation), l'ICM est très rapide.

Par analogie avec le recuit simulé, on peut également introduire une variation de température dans l'algorithme [GUYO].

5.2.2. Non Convexité Graduelle (GNC)

Dans [BLAK87], A. Blake et A. Zisserman proposent un algorithme déterministe d'optimisation du critère non convexe du MAP. Le principe de l'algorithme est le suivant (voir figure 1-7): il s'agit de procéder par étapes. A la première étape ($n=0$), on minimise une approximation convexe du critère du MAP, notée $J^{(0)}$. Ce critère étant convexe, on trouve aisément son minimum global à l'aide d'un algorithme simple de descente (du type Gradient Conjugué ou SOR), quel que soit le point de départ.

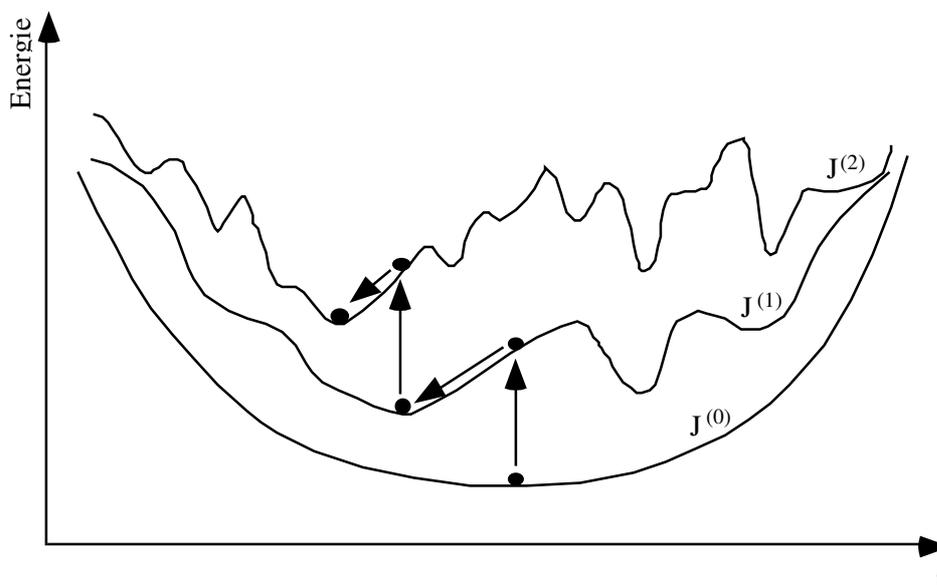


Figure 1-7: Principe de l'algorithme GNC (d'après [BLAKE])

Par la suite, on minimise à chaque étape une énergie non convexe $J^{(n)}$. On utilise à chaque fois un algorithme simple de descente, initialisé avec la solution calculée à l'étape précédente.

La suite des énergies minimisées à chaque étape est construite de façon à tendre vers l'énergie du MAP: $J^{(n)}(f) \xrightarrow{n \rightarrow \infty} J(f)$. Si la variation de forme du critère entre deux étapes est infime, on peut espérer trouver le minimum global du critère du MAP. Toutefois, aucune preuve formelle de convergence vers un minimum local ou global de l'énergie du MAP n'est donnée dans le cas général dans [BLAK87]. Notons que, comme $J^{(0)}$ est convexe, le résultat de l'algorithme ne dépend pas des

conditions initiales.

Ce type d'algorithme introduit donc graduellement la non convexité au niveau du critère minimisé. C'est pourquoi Blake et Zisserman le nomment "algorithme de Non Convexité Graduelle" (GNC en anglais) .

5.2.3. Recuit par Champ Moyen (MFA)

L'algorithme du recuit par champ moyen a été proposé par D. Geiger et F. Girosi. Le lecteur pourra se référer à [GEIG91] pour plus de détails (voir aussi Chapitre III).

L'algorithme de recuit par champ moyen met en œuvre une descente en température. A chaque température, on calcule de façon déterministe une estimée au sens du champ moyen. Lorsque la température s'annule, l'estimée calculée est une bonne approximation de la solution [ZERU92].

On peut considérer que, comme pour le GNC, on minimise une suite de fonctionnelles. Toutefois, la construction de cette suite est différente. Dans le cas du GNC, la suite était construite de façon *ad hoc* alors que dans le cas du MFA, elle est construite naturellement en fonction de la température du système. Une comparaison expérimentale des deux algorithmes est proposée dans [ZERU90]. Il apparaît que leurs performances sont proches.

6. Conclusion

Dans ce chapitre, nous avons décrit le concept de régularisation dans une approche Bayésienne. Nous proposons d'utiliser l'estimateur du Maximum A Posteriori. L'estimation de la solution du problème régularisé s'opère en minimisant une énergie (dite énergie du MAP). Cette énergie est formée de deux termes: un terme d'attache aux données et une énergie *a priori* qui permet d'exprimer les contraintes imposées à l'image. Nous avons étudié deux des principaux modèles d'attache aux données: la vraisemblance Poissonienne et la vraisemblance Gaussienne. Nous avons également spécifié d'une manière très générale le modèle d'image *a priori*. Pour cela, nous utilisons la théorie des champs Markoviens, qui permet de prendre en compte les attributs locaux de l'image. Ayant ainsi défini l'énergie du MAP, nous avons abordé le problème de sa minimisation.

Dans le cas où le critère du MAP est non convexe, deux types d'algorithmes peuvent être utilisés. D'une part, les algorithmes stochastiques permettent en principe d'atteindre le minimum global du critère. L'inconvénient de ces algorithmes est leur lourdeur d'utilisation. D'autre part, les algorithmes itératifs déterministes sont beaucoup plus rapides, mais n'atteignent qu'un minimum

local du critère. Certains algorithmes, comme le MFA ou le GNC, fournissent cependant des solutions de bonne qualité.

Dans le cas où le critère du MAP est convexe, il n'est bien entendu pas nécessaire d'utiliser un algorithme stochastique. Toutefois, même dans ce cas, il reste un problème à résoudre. En effet, le critère du MAP est la plupart du temps non quadratique. Les équations normales associées sont donc non linéaires. Dans le chapitre III, nous présentons deux algorithmes déterministes pour la minimisation du critère du MAP. Ces algorithmes permettent de résoudre le système d'équations normales non linéaires associées au critère du MAP en résolvant une suite de systèmes linéaires. Dans le cas où le critère du MAP est convexe, l'estimée calculée correspond à la solution unique du MAP. Dans le cas contraire, l'estimée obtenue est de très bonne qualité.

Chapitre II

Régularisation avec prise en compte des discontinuités, régularisation semi-quadratique

1. Introduction

Comme nous l'avons vu au Chapitre I, l'image estimée est obtenue par minimisation du critère du MAP. Ce critère est la somme d'un terme d'attache aux données, et d'un terme de régularisation:

$$J(f) = J_1(f) + \lambda^2 J_2(f) \quad (2-1)$$

Dans notre cas, le terme de régularisation est défini par:

$$J_2(f) = \sum_k \varphi[(D_x f)_k] + \sum_k \varphi[(D_y f)_k] \quad (2-2)$$

où D_x et D_y sont des opérateurs différentiels. Dans ce chapitre, nous nous intéressons à la régularisation avec prise en compte des discontinuités. Notre but est d'éliminer l'effet du bruit sur la solution obtenue, tout en préservant les contours, éléments essentiels à la "lecture" de l'image. Dans ce cas, l'*a priori* imposé à la solution est le suivant:

"la solution estimée doit être formée de zones homogènes, séparées par des bords francs".

Dans nos travaux, nous utilisons l'opérateur gradient. La solution recherchée est considérée

comme formée de zones constantes, séparées par des bords francs. On peut également utiliser des opérateurs d'ordres supérieurs [GEMA92], [GEMA93]. Par exemple, en utilisant un opérateur de dérivée seconde, on cherchera une solution formée de zones à pente constante. Notons que tout ce qui est développé dans ce paragraphe ne dépend pas de l'opérateur utilisé.

Dans un premier temps, nous nous intéressons aux fonctions de potentiel φ : quelles propriétés doivent vérifier ces fonctions de façon à assurer la prise en compte des discontinuités?

Ce type de régularisation implique l'emploi de fonctionnelles difficiles à manipuler d'un point de vue algorithmique. Dans la seconde partie de ce chapitre, nous présentons la régularisation semi-quadratique, qui permet de faciliter la minimisation de telles fonctionnelles.

2. Fonctions de potentiel et prise en compte des discontinuités

Remarques préliminaires: Les fonctions de potentiel utilisées pour la prise en compte des discontinuités sont des fonctions paires. En effet, il paraît logique de traiter de façon identique deux gradients de même valeur absolue mais de signe opposé. Nous limiterons donc notre étude à l'intervalle $[0, +\infty[$.

De même, comme on le voit dans l'équation (2-2), on traite de façon identique la composante horizontale et le composante verticale du gradient. Dans ce paragraphe, par abus de langage, le mot gradient désignera indifféremment l'une ou l'autre des composantes $(D_x f)_k$ ou $(D_y f)_k$.

Enfin, dans la pratique, on peut être amené à traiter des images de dynamiques très différentes. On introduit habituellement un paramètre δ qui permet la mise à l'échelle de l'*a priori* par rapport aux données. Ainsi, les potentiels ne sont plus fonction des différences entre pixels voisins $f_s - f_t$, mais de $(f_s - f_t) / \delta$. Afin d'alléger les notations, nous "oublierons" momentanément ce paramètre et nous noterons u les différences entre pixels voisins.

2.1. Modèle quadratique

Intéressons-nous tout d'abord au cas où la fonction de potentiel utilisée est quadratique:

$$\varphi_Q(u) = u^2 \quad (2-3)$$

Le critère du MAP se met alors sous la forme suivante:

$$J_{Tik}(f) = \|p - Rf\|^2 + \lambda^2 \|D_x f\|^2 + \lambda^2 \|D_y f\|^2 \quad (2-4)$$

Ce schéma correspond en fait à une régularisation de Tikhonov [TIKH63], l'opérateur considéré étant le gradient: on impose une contrainte *globale* de douceur à la solution. Ce type de modèle décourage la formation de forts gradients dans l'image. En effet, la minimisation de J_{Tik} passe par une diminution globale des valeurs du gradient, donc un lissage global de la solution. Les discontinuités sont excessivement pénalisées et l'image estimée a un aspect trop lisse.

Pour éviter cela, on peut choisir le coefficient de lissage λ faible. Ainsi, le terme d'attache aux données devient prépondérant et on pénalise moins les gradients correspondant aux discontinuités, qui se trouvent préservées. Toutefois, on ne décourage pas non plus les gradients correspondant au bruit. La solution obtenue contient donc des discontinuités, mais n'a pas un aspect homogène.

Ce modèle ne permet donc pas simultanément le lissage des zones homogènes et la préservation des discontinuités. Il faudrait en fait pouvoir interrompre le lissage aux emplacements des discontinuités.

2.2. Modèle avec processus de ligne

2.2.1 Processus de ligne Booléen

Dans leur article de 1984, Donald et Stuart Geman proposent d'introduire dans la modélisation du problème une variable auxiliaire Booléenne b . Le rôle de cette variable est de marquer les discontinuités: elle vaut 0 si une discontinuité existe entre deux pixels adjacents, et 1 sinon. Cette variable est appelée "processus de ligne". Le processus de ligne est localisé sur une grille *duale* à la grille de l'image, comme le montre la figure 2-1.

Remarques:

- Pour des raisons de cohérence (cf §2.2.3 et §3), nous adoptons une logique inverse de celle utilisée dans [GEMA84], où le processus vaut 0 en l'absence de discontinuité et 1 sinon. Nous garderons notre convention dans toute la suite de cet exposé.

- D'autre part, nous considérons que $b = (b_x, b_y)$. Le *processus de ligne horizontal* b_x marque les discontinuités entre pixels situés sur une même ligne, tandis que le *processus de ligne vertical* b_y marque les discontinuités entre pixels d'une même colonne.

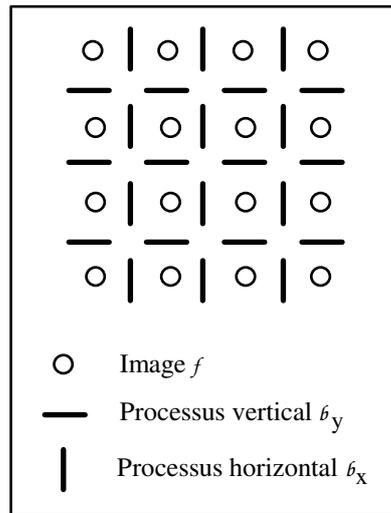


Figure 2-1: Localisation du processus de ligne

De même que pour f , on peut définir pour b un système de voisinage. Le système de voisinage associé au couple (f, b) est l'union des systèmes de voisinage de f et b .

Lorsqu'on utilise ce type de modèle, on considère non plus le problème de l'estimation du champ f , mais celui de l'estimation simultanée de f et de la variable non-observée b . En supposant que le couple (f, b) est un champ Markovien composé, son estimation au sens du MAP se ramène à la minimisation en f et en b d'une fonctionnelle *augmentée* de la forme:

$$J_B^*(f, b) = J_1(f) + \lambda^2 J_2^*(f, b) \quad (2-5)$$

où $J_2^*(f, b)$ s'écrit comme une somme de potentiels sur les cliques associées au système de voisinage du couple (f, b) . En utilisant les cliques à deux sites pour f et les cliques à un site pour b , on a:

$$J_2^*(f, b) = \sum_k \varphi^* [(D_x f)_k, (b_x)_k] + \sum_k \varphi^* [(D_y f)_k, (b_y)_k] \quad (2-6)$$

Ces équations sont comparables aux équations (2-1) et (2-2) à ceci près que le terme de régularisation est maintenant fonction du processus de ligne. Plus précisément, on peut écrire [GEMA84]:

$$J_2^*(f, b) = J_{Inter}^*(f|b) + J_{Pl}^*(b) \quad (2-7)$$

Le premier facteur est appelé *énergie d'interaction* entre f et b . Dans notre cas:

$$J_{Inter}^*(f|b) = \sum_k \varphi_{Inter}^* \left[(D_x f)_k | (b_x)_k \right] + \sum_k \varphi_{Inter}^* \left[(D_y f)_k | (b_y)_k \right] \quad (2-8)$$

Par un choix approprié du potentiel d'interaction φ_{Inter}^* , il est possible de “casser” le modèle aux endroits marqués par le processus de ligne. Ainsi, on peut poser:

$$\varphi_{Inter}^*(u|b) = bu^2 \quad (2-9)$$

Cette fonction est quadratique (cf §2.1) partout où $b = 1$, mais s'annule aux endroits des discontinuités, marquées par $b = 0$.

Le second facteur, $J_{Pl}^*(b)$, est l'énergie de régularisation du processus de ligne. C'est le coût associé à la création d'une discontinuité. Son absence conduit à des solutions incohérentes “avec des discontinuités partout” ([GEIG91]). Cette énergie s'écrit comme une somme de potentiels φ_{Pl}^* sur les cliques associées au système de voisinage de b . La fonction φ_{Pl}^* permet de traduire l'*a priori* imposé au processus de ligne. On trouve principalement deux types de modèle dans la littérature.

Dans le premier cas, on utilise les cliques à un site:

$$J_{Pl}^*(b) = \sum_k \varphi_{Pl}^* \left[(b_x)_k \right] + \sum_k \varphi_{Pl}^* \left[(b_y)_k \right] \quad (2-10)$$

et on associe à chaque discontinuité une pénalité de 1 en posant:

$$\varphi_{Pl}^*(b) = 1 - b \quad (2-11)$$

Ceci revient donc à imposer une contrainte sur la norme du processus de ligne. Ce type de modèle pénalise les discontinuités indépendamment les unes des autres. On ne tient pas compte d'éventuelles interactions entre éléments de contours.

Dans le second cas, on utilise le fait que les lignes de contours sont la plupart du temps structurées [GEMA93]. La fonction de potentiel peut, comme dans [GEIG91], être définie sur les cliques à trois voisins du système de voisinage de b par:

$$\varphi_{Pl}^* \left[(b_x)_{i,j} \right] = 1 - \varepsilon (h_x)_k \frac{(h_x)_{k+1} + (h_x)_{k-1}}{2} \quad \text{où } \varepsilon \in [0,1] \quad (2-12)$$

Pour respecter nos conventions, on a $(h_x)_k = 1 - (b_x)_k$. Une définition similaire à (2-12) est valable pour le processus de ligne vertical. Ce modèle décourage les discontinuité isolées et tend à prolonger les lignes de contour. Par contre, il ne traite pas tous les types d'interaction entre

processus de ligne. Un modèle plus complet utilise des cliques à quatre éléments. On affecte de façon arbitraire, ou par estimation, une valeur de potentiel à chaque cas de figure: pas de contour, terminaison, prolongement d'un contour, coin, té, croisement.

○ ○	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○
○ ○	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○
0	2,7	0,9	1,8	1,8	1,8	2,7
<i>0</i>	<i>2,0</i>	<i>0,25</i>	<i>0,8</i>	<i>1,2</i>	<i>1,2</i>	<i>2,0</i>

Figure 2-2: Valeurs de potentiel de régularisation de b définies dans [GEMA84] (en gras) et [MARR84] (en italique)

A titre d'exemple, nous donnons figure 2-2 les valeurs de potentiel définies dans [GEMA84] et [MARR84] (les configurations possibles sont données aux rotations près).

Par la suite, nous utiliserons le modèle le plus simple, sans interaction entre éléments du processus de lignes. Dans le cas du processus de ligne Booléen, nous écrivons donc le critère à minimiser comme:

$$J_B^*(f, b) = \|p - Rf\|^2 + \lambda^2 \sum_k \varphi_B^*[(D_x f)_k, (b_x)_k] + \lambda^2 \sum_k \varphi_B^*[(D_y f)_k, (b_y)_k] \quad (2-13)$$

avec

$$\varphi_B^*(u, b) = bu^2 + (1 - b) \quad (2-14)$$

En dehors des discontinuités (pour $b=1$), φ_B^* se comporte comme le potentiel quadratique présenté au §2.1 et favorise donc le lissage. Par contre, le coût associé aux discontinuités ($b=0$) est constant et ne dépend pas de u . Aux emplacements des contours, une diminution de la valeur de u est donc sans intérêt pour la minimisation de J_B^* . De cette façon, on interrompt le lissage aux sites marqués par b .

En conclusion, l'intérêt principal du modèle avec processus de ligne est le suivant:

il est possible de "casser" l'interaction entre pixels aux endroits marqués par b .

Remarque: dans [BLAK87], A. Blake et A. Zisserman préfèrent utiliser une modélisation déterministe de l'image. Toutefois, en se basant sur des analogies avec la mécanique, ils écrivent finalement l'énergie à minimiser sous une forme similaire à (2-13), faisant intervenir

explicitement le processus de ligne. Au paragraphe suivant, nous voyons comment ces auteurs font correspondre à ce modèle un processus de ligne *implicite*.

2.2.2 Processus de ligne Booléen implicite

Dans [BLAK87], A. Blake et A. Zisserman remarquent que dans (2-13), le terme d'attache aux données ne dépend pas de b . Ils proposent donc d'effectuer d'abord la minimisation en b , en écrivant:

$$\min_b \{\varphi_{GG}^*(u, b)\} = \varphi_{BZ}(u) \quad (2-15)$$

Un raisonnement simple permet de constater que la fonction $\varphi_{BZ}(u)$ est la quadratique tronquée définie par:

$$\varphi_{BZ}(u) = \min(u^2, 1) \quad (2-16)$$

Cette fonction est présentée sur la figure 2-3.

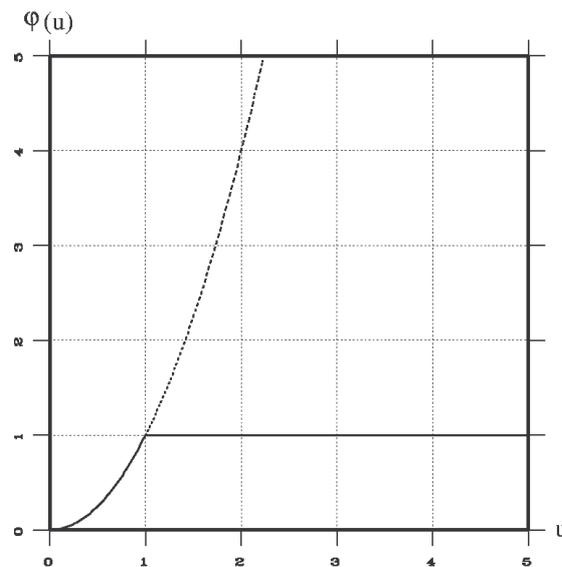


Figure 2-3: Fonction quadratique (pointillés) et quadratique tronquée (trait plein)

L'équation (2-15) est très importante. Elle signifie:

“il y a équivalence entre le modèle sans processus de ligne avec $\varphi = \varphi_{BZ}$ et le modèle faisant explicitement apparaître le processus de ligne Booléen”.

On dit que φ_{BZ} correspond à un modèle avec processus de ligne *implicite* Booléen. Notons que Blake et Zisserman n'utilisent pas le modèle explicite pour l'estimation de la solution, mais le modèle implicite. Pour ces auteurs, l'intérêt de l'équivalence entre les deux modèles est donc simplement de justifier le choix de la quadratique tronquée comme fonction de potentiel [GEMA93].

L'inconvénient de la fonction φ_{BZ} est qu'elle n'est pas dérivable en $u = 1$. On note donc une variation abrupte de son comportement pour $u = 1$ [BOUM93]. En conséquence, une petite variation de u autour de 1 peut provoquer une variation importante du potentiel, qui se répercute sur la solution estimée. Ceci va à l'encontre du but fixé: régulariser le problème.

2.2.3 Processus de ligne continu

Une façon de pallier l'inconvénient de la fonction φ_{BZ} est d'utiliser une fonction continue sur $[0, +\infty[$ et "ressemblant" à $\varphi_{BZ}(u)$. C'est le cas de la fonction proposée par S. Geman et D. McClure dans [GEMA85], tracée figure 2-4:

$$\varphi_{GM}(u) = \frac{u^2}{(1+u^2)} \quad (2-17)$$

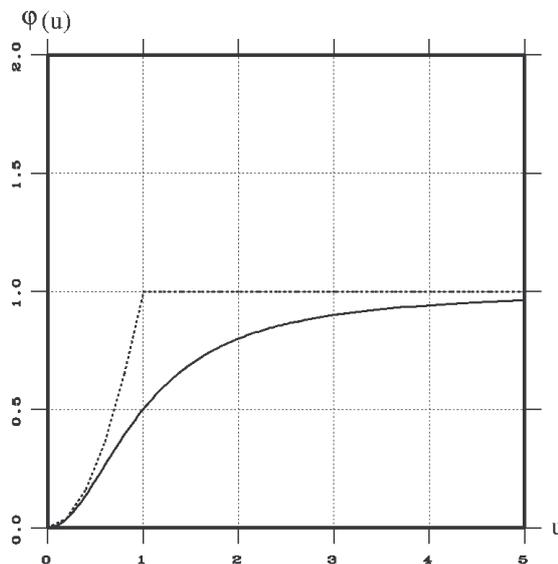


Figure 2-4: Quadratique tronquée (trait pointillé) et fonction de Geman et McClure (trait plein)

On peut montrer que cette fonction correspond à un modèle avec processus de ligne implicite *continu*. C'est en fait le cas de toutes les fonctions de potentiel satisfaisant aux conditions du théorème suivant, énoncé par D. Geman et G. Reynolds dans [GEMA92]:

Théorème 1:

Supposons que $\varphi(u)$ vérifie les propriétés suivantes sur $[0, +\infty[$:

- 1) $\varphi(0) = 0$
- 2) $\varphi(\sqrt{u})$ est concave
- 3) $\lim_{u \rightarrow +\infty} \varphi(u) = 1$

Alors, il existe une fonction $\psi(b)$ définie sur un intervalle $[0, M]$ telle que:

$$\varphi(u) = \inf_{0 \leq b \leq M} (bu^2 + \psi(b)) \quad (2-18)$$

et telle que $\psi(b)$ a les propriétés suivantes:

- a) $\psi(0) = 1$
- b) $\psi(b)$ est strictement décroissante
- c) $\psi(M) = 0$

Remarque: afin de rester fidèles à nos conventions concernant le processus de ligne, nous ne donnons pas ce théorème sous sa forme originale (où φ et ψ sont à valeurs négatives) mais sous la même forme que dans [BRUN93].

L'équation (2-18) et l'équation (2-15) procèdent de la même idée: assurer l'équivalence entre un modèle utilisant une fonction de potentiel donnée et un modèle avec processus de ligne explicite. D'après le théorème de Geman et Reynolds, la minimisation du critère du MAP est équivalente à la minimisation d'une fonctionnelle augmentée de la forme:

$$J_B^*(f, b) = J_1(f) + \lambda^2 [J_{Inter}^*(f|b) + J_{Pl}^*(b)] \quad (2-19)$$

où J_{Inter}^* est défini comme dans (2-8) avec $\varphi_{Inter}^*(u|b) = bu^2$ et J_{Pl}^* est défini comme dans (2-10) avec $\varphi_{Pl}^* \equiv \psi$.

On peut alors voir le modèle utilisant la quadratique tronquée comme le cas particulier où b est Booléen et $\psi(b) = (1-b)$.

2.2.4 Conclusion

L'intérêt des modèles incluant un processus de ligne est d'assurer de façon forte la prise en compte des discontinuités dans la régularisation. Le théorème de Geman & Reynolds montre que

les potentiels vérifiant les conditions 1), 2) et 3) correspondent à un modèle avec processus de ligne implicite continu. D'après ces auteurs, les propriétés essentielles à la préservation des discontinuités sont donc la concavité de $\varphi(\sqrt{u})$ et le comportement asymptotique de $\varphi(u)$ [GEMA92].

2.3. Autres approches

D'autres auteurs se sont penchés sur le problème de la régularisation avec prise en compte des discontinuités, sans utiliser de processus de ligne. En considérant (2-2), ils se sont attachés à trouver des fonctions de potentiel φ permettant de respecter au mieux les discontinuités. Toutefois, peu d'études théoriques ont été faites. Nous présentons ici les fonctions de potentiel les plus couramment utilisées. Cette présentation n'est bien sûr pas exhaustive.

2.3.1 Fonction d'Hebert et Leahy

Dans le contexte de la reconstruction tomographique régularisée, T. Hebert et R. Leahy proposent dans [HEBE89] d'utiliser la fonction suivante (tracée figure 2-5):

$$\varphi_{HL}(u) = \log(1 + u^2) \quad (2-20)$$

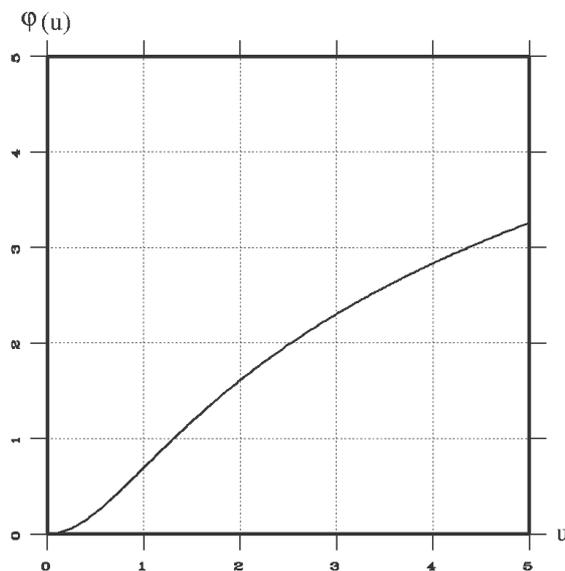


Figure 2-5: Fonction de Hebert et Leahy

Cette fonction non-convexe est présentée comme un compromis entre la quadratique et la fonction de Geman et McClure. Notons au passage que φ_{GM} est la dérivée de φ_{HL} par rapport à u^2 . Notons également que cette fonction ne satisfait pas les conditions du théorème de Geman et

Reynolds en ce qui concerne le comportement asymptotique de la fonction de potentiel.

2.3.2 Potentiels convexes

L'utilisation de potentiels non convexes rend l'estimation au sens du MAP difficile, puisque dans ce cas l'énergie à minimiser admet des minima locaux. C'est pourquoi plusieurs auteurs plaident pour l'utilisation de potentiels convexes, qui correspondent à des critères du MAP convexes.

Nous allons maintenant citer les principaux potentiels convexes connus. Dans [BESA86], J. Besag propose d'utiliser une fonction linéaire:

$$\varphi_{BE}(u) = |u| \quad (2-21)$$

Plus généralement, C. Bouman et K. Sauer [BOUM93] proposent une fonction de la forme suivante:

$$\varphi_{BS}(u) = |u|^\alpha \quad 1 \leq \alpha < 2 \quad (2-22)$$

Ces fonctions sont tracées sur la figure 2-6 pour α variant de 1 à 1,8 par pas de 0,2.

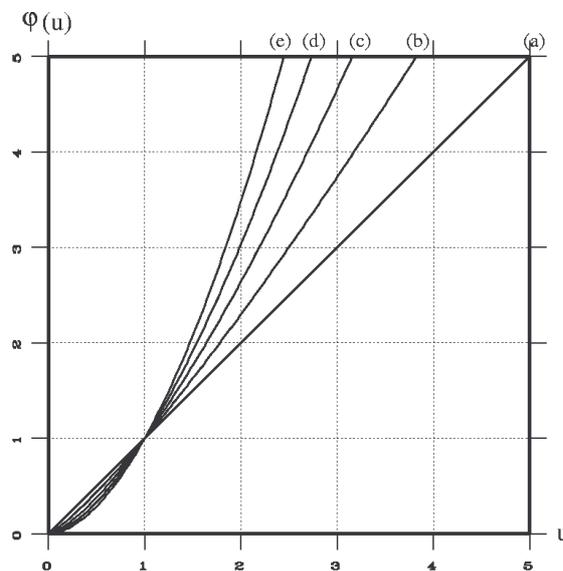


Figure 2-6: Fonction $|u|^\alpha$ pour: $\alpha = 1$ (a); $\alpha = 1,2$ (b); $\alpha = 1,4$ (c); $\alpha = 1,6$ (d); $\alpha = 1,8$ (e)

Dans ce modèle, Bouman et Sauer considèrent l'image comme un "Champ Markovien Gaussien Généralisé"[BOUM93]. Pour $\alpha = 1$, la distribution *a priori* est Laplacienne; pour $\alpha = 2$, elle

serait Gaussienne, mais dans ce cas, le potentiel est quadratique et les discontinuités ne sont plus prises en compte. C'est pourquoi nous interdisons la valeur $\alpha = 2$ dans (2-21).

Enfin, nous citons une dernière famille de fonctions. Ces fonctions (cf. figure 2-7) ont un comportement quadratique (ou plus généralement polynomial) en 0 et linéaire à l'infini.

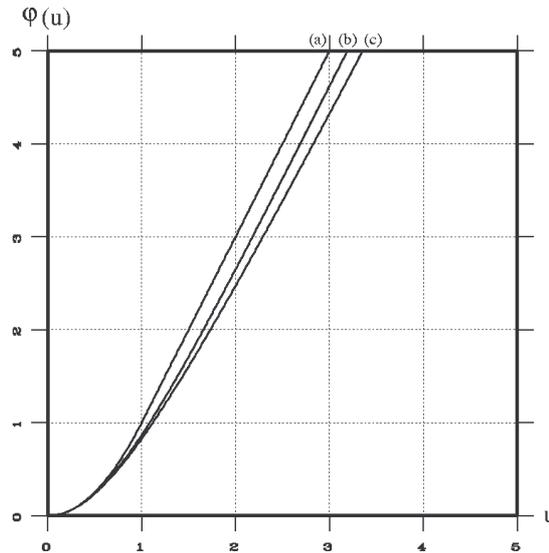


Figure 2-7: Fonction de Huber (a), fonction logcosh (b) et fonction des surfaces minimales (c)

C'est en particulier le cas de la fonction de Huber, utilisée par Stevenson et al (voir par exemple [SCHU93]):

$$\varphi_{HU}(u) = \min(u^2, 2u - 1) \quad (2-23)$$

et de la fonction log cosh proposée par Green dans [GREE90A]. Nous utiliserons la fonction de Green sous la forme:

$$\varphi_{GR}(u) = 2 \log \cosh(u) \quad (2-24)$$

Cette fonction peut être approximée par un polynôme en zéro, et devient très rapidement linéaire (en $2[u - \ln(2)]$) lorsque u augmente.

Pour notre part, nous proposons la fonction “des surfaces minimales”:

$$\varphi_{HS}(u) = 2\sqrt{1+u^2} - 2 \quad (2-25)$$

Comme on peut le constater sur la figure 2-7, la fonction φ_{HS} est très proche de la fonction log cosh. Par contre, elle est plus facile à manipuler du point de vue formel et c'est pourquoi nous en

recommandons l'usage.

D'autre part, la fonction φ_{HS} a une interprétation physique intéressante. Cette fonction intervient dans le calcul d'hyper-surfaces minimales. En effet, à une dimension, on constate que la longueur de la courbe $z = f(x)$ entre les points d'abscisses a et b est:

$$L = \int_a^b \sqrt{1 + [f'(x)]^2} dx \quad (2-26)$$

Ce résultat se généralise aux dimensions supérieures, la notion de longueur se généralisant à la notion d'hyper-surface.

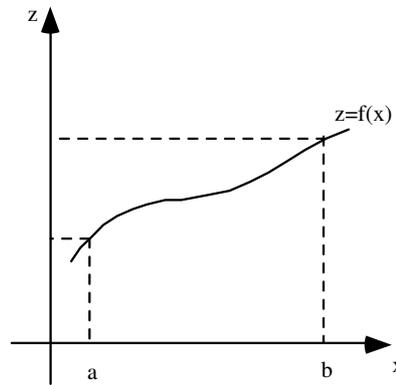


Figure 2-8: Courbe $z = f(x)$

Lorsque l'on utilise cette fonction de potentiel, le terme de régularisation intervenant dans le critère du MAP se met sous la forme:

$$J_2(f) = 2 \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sqrt{1 + (D_x f)_{i,j}^2} + 2 \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sqrt{1 + (D_y f)_{i,j}^2} - 4N^2 \quad (2-27)$$

Considérons la i -ème ligne de l'image, f_i . La valeur du pixel en fonction de l'indice de colonne j nous fournit une courbe (discrète) $z = f_{i,j}$. On peut alors interpréter le terme:

$$\sum_{j=0}^{N-1} \sqrt{1 + (D_x f)_{i,j}^2}$$

comme l'analogie discret de (2-26). Ce terme représente donc la longueur de la courbe $z = f_{i,j}$. On peut faire la même interprétation dans la direction y , en considérant une colonne de l'image.

L'utilisation de la fonction φ_{HS} revient donc à poser une contrainte sur la longueur totale des courbes décrites par les lignes et les colonnes de l'image.

2.3.3 Les conditions de Lange

Enfin, quant à déterminer de façon théorique les propriétés à vérifier par la fonction de potentiel, peu d'auteurs se sont posés explicitement la question. On trouve cependant une discussion sur ce sujet dans [LANG90]. K. Lange propose sept conditions¹, que nous pouvons regrouper sous la forme suivante:

- i) φ doit être paire, à valeurs positives,
- ii) φ doit être strictement croissante sur $[0, +\infty[$ et vérifier

$$\lim_{0^+} \varphi = 0 \text{ et } \lim_{+\infty} \varphi = +\infty,$$
- iii) la dérivée de φ doit être bornée, (2-28)
- iv) φ doit être strictement convexe.

Les motivations de cet auteur sont en fait légèrement différentes des nôtres, puisqu'il s'agit surtout d'assurer la convergence de l'algorithme MAP-EM OSL [GREE90A]. Toutefois, Lange fait remarquer que toutes ces conditions sont vérifiées par la fonction $\log \cosh$ dont le comportement linéaire à l'infini est censé permettre le respect des discontinuités ([BOUM93]). Il apparaît que les conditions ii) et iv) posées par l'auteur sont en contradiction avec les hypothèses du théorème de Geman et Reynolds (cf. §2.2.3). Notons que la condition iv) assurant l'unicité du minimum de l'énergie du MAP est également défendue par d'autres auteurs, par exemple Green [GREE90A], Bouman et Sauer [BOUM93].

2.3.4 Conclusion

Nous venons de voir dans les paragraphes 2.2 et 2.3 deux approches de la régularisation avec prise en compte des discontinuités:

- la première fait appel à la notion de processus de ligne, une variable permettant de marquer les discontinuités. Cette variable peut être implicitement définie à travers la fonction de potentiel si cette dernière vérifie certaines conditions, précisées dans le théorème de Geman et Reynolds.
- le second type d'approche ne fait pas appel à un processus de ligne. Il s'agit de trouver une "bonne" fonction de potentiel. Cette recherche est souvent empirique ([HEBE89], [LALU92]).

¹ Remarque: dans [LANGE90], K. Lange propose également de nouvelles fonctions de potentiel, que nous n'étudierons toutefois pas ici.

Nous notons que ces deux approches imposent des conditions différentes (et parfois contradictoires...) à la fonction de potentiel pour qu'elle assure la préservation des discontinuités.

Enfin, si toutes les fonctions citées précédemment sont censées permettre la prise en compte des discontinuités, elles ont des allures très différentes, comme le montre la figure 2-9.

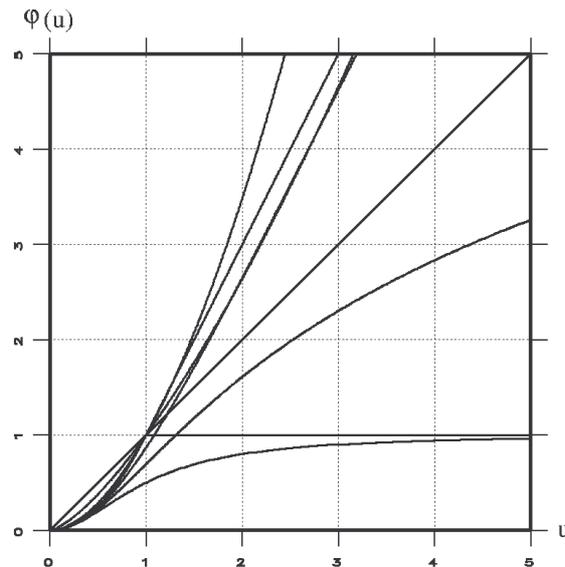


Figure 2-9: Diversité des fonctions de potentiel censées correspondre à une régularisation avec prise en compte des discontinuités

Certes, il est possible de paramétrer ces fonctions (comme par exemple dans [GREE90A] ou [LALU92]) de façon à les faire mieux “se ressembler”. Toutefois, il n’en demeure pas moins que les allures générales de ces fonctions sont très diverses. De plus, certaines sont convexes, ce qui assure l’existence d’un minimum unique du critère du MAP, d’autres non.

La question: “quelles propriétés doit vérifier une fonction de potentiel pour assurer la préservation des discontinuités?” reste donc posée. Afin d’y apporter une réponse, et dans une démarche unificatrice, nous proposons au paragraphe suivant une approche différente, basée sur l’étude des équations normales associées au critère du MAP.

2.4. Etude heuristique des équations normales

Nous supposerons par la suite que φ est continûment dérivable.

2.4.1 Equations normales

Les équations normales associées au critère du MAP (2-1) sont obtenues en annulant sa dérivée (le calcul complet est donné en annexe A):

$$\frac{\partial J}{\partial f} = 0 \Leftrightarrow (R^t R - \lambda^2 \Delta_{pond}) f = R^t p \quad (2-29)$$

Dans cette équation, la matrice Δ_{pond} est une approximation discrète d'un Laplacien pondéré. Au site (i,j), les coefficients du Laplacien pondéré sont donnés par:

$$\begin{pmatrix} 0 & \lambda^x & 0 \\ \lambda^w & -\Sigma & \lambda^e \\ 0 & \lambda^s & 0 \end{pmatrix} \quad (2-30)$$

les poids étant définis de la façon suivante:

$$\begin{array}{|c|c|} \hline \lambda^E = \frac{\varphi'(f_{i,j+1} - f_{i,j})}{2(f_{i,j+1} - f_{i,j})} & \lambda^W = \frac{\varphi'(f_{i,j} - f_{i,j-1})}{2(f_{i,j} - f_{i,j-1})} \\ \hline \lambda^S = \frac{\varphi'(f_{i+1,j} - f_{i,j})}{2(f_{i+1,j} - f_{i,j})} & \lambda^N = \frac{\varphi'(f_{i,j} - f_{i-1,j})}{2(f_{i,j} - f_{i-1,j})} \\ \hline \end{array} \quad (2-31)$$

La fonction $\varphi'(u)/2u$ définissant les poids du Laplacien est appelée *fonction de pondération*.

Nous allons maintenant étudier ces équations normales de façon à mettre en évidence des conditions assurant que la régularisation tient bien compte des discontinuités.

2.4.2 Etude de cas: lissage et prise en compte des discontinuités

Le premier de nos objectifs est de **lisser les zones homogènes** de l'image. Dans un premier temps, supposons donc qu'il n'y ait pas de discontinuités dans le voisinage de $f_{i,j}$, c'est-à-dire que tous les gradients autour de ce site soient faibles. Si φ vérifie:

$$\lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u} = 1 \quad (2-32)$$

alors les poids autour du pixel $f_{i,j}$ sont tous égaux à 1 et le Laplacien pondéré se comporte comme un Laplacien simple (cf. fig. 2-10(a)). L'équation normale a la forme d'une équation de diffusion. Il y aura donc lissage dans les quatre directions autour du site (i,j).

$$\begin{array}{cc} \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 & 0 \\ 0 & -3 & 1 \\ 0 & 1 & 0 \end{pmatrix} \\ (a) & (b) \end{array}$$

Figure 2-10: Pondérations du Laplacien au voisinage du site (i,j);

(a) pas de discontinuité, (b) discontinuité entre (i,j) et (i,j-1)

Le second objectif est de **préserv**er les **discontinuités** du lissage. Supposons qu'il y ait une discontinuité (c'est à dire un fort gradient) entre les sites (i,j) et (i,j-1). Si φ vérifie

$$\lim_{u \rightarrow +\infty} \frac{\varphi'(u)}{2u} = 0 \quad (2-33)$$

alors, le coefficient correspondant du Laplacien s'annule (cf. fig. 2-10(b)) et il n'y aura pas de lissage dans cette direction.

2.4.3 Conclusion

Nous venons de mettre en évidence deux conditions aux limites pour la fonction de pondération $\varphi'(u)/2u$. Afin d'assurer la continuité de la solution par rapport aux données, nous imposons de plus à cette fonction de varier continûment entre ses limites. En résumé, de façon à assurer la prise en compte des discontinuités, la fonction de potentiel utilisée doit être telle que:

$$\begin{array}{l} \text{i) } \lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u} = 1 \\ \text{ii) } \lim_{u \rightarrow +\infty} \frac{\varphi'(u)}{2u} = 0 \\ \text{iii) } \frac{\varphi'(u)}{2u} \text{ est continue et strictement décroissante sur } [0, +\infty[. \end{array} \quad (2-34)$$

Remarque: un des buts de la régularisation est d'éliminer les effets du bruit sur la solution estimée en effectuant un lissage. Dans notre cas, on veut également préserver les éléments importants de l'image que sont les discontinuités. Dans l'étude que nous venons de faire, on fait implicitement l'hypothèse que les forts gradients correspondent aux contours présents dans l'image originale, et qu'on doit donc les préserver. Ce n'est pas toujours vrai en pratique: de forts gradients peuvent aussi être dûs au bruit. De la même façon, on a supposé que les faibles gradients correspondent aux effets du bruit, et qu'on doit les lisser. Dans la réalité, les faibles gradients peuvent aussi correspondre à des contours inclus dans l'image.

Il apparaît donc important de pouvoir décider à partir de quelle valeur du gradient on introduit une discontinuité. C'est pourquoi on introduit dans la pratique un paramètre de hauteur de discontinuité (cf. Chapitre III). Ce paramètre permet de discriminer au mieux les attributs de l'image et les effets du bruit. Cette discrimination n'est pas forcément parfaite dans la réalité, en particulier s'il existe dans l'image des pics de bruit d'amplitude aussi importante que certains contours.

2.5. Application aux fonctions de potentiel précitées

2.5.1 Fonctions de pondération correspondantes

Le tableau 2-1 récapitule les expressions des fonctions de potentiel citées ci-dessus et donne les expressions des fonctions de pondération $\varphi'(u)/2u$ correspondantes. Certaines de ces fonctions respectent les conditions (2-34), d'autres non. Ainsi, il est clair que la fonction quadratique ne permet pas la prise en compte des discontinuités puisque la fonction de pondération associée vaut 1 quelle que soit la valeur du gradient (cf. figure 2-11).

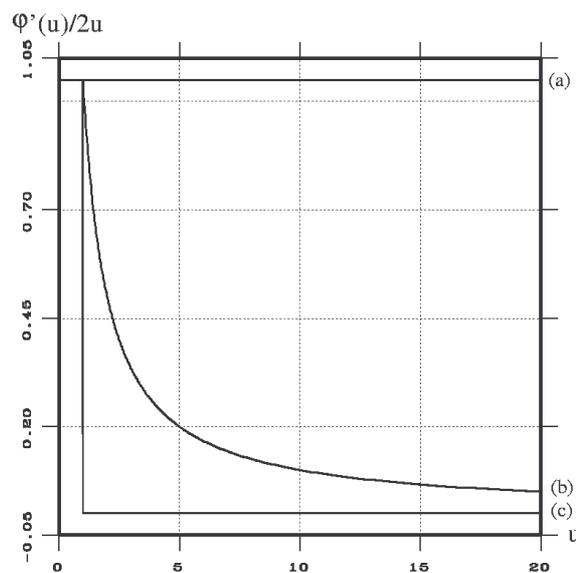


Figure 2-11: Fonction de pondération correspondant à la quadratique (a), la fonction de Huber (b) et la quadratique tronquée (c).

Potentiel	Expression de φ	Expression de $\varphi'(u)/2u$
φ_Q	u^2	1
φ_{BZ}	$\min(u^2, 1)$	$\begin{cases} 1 & u < 1 \\ 0 & u \geq 1 \end{cases}$
φ_{HU}	$\min(u^2, 2u - 1)$	$\begin{cases} 1 & u < 1 \\ 1/u & u \geq 1 \end{cases}$
φ_{BS}	$ u ^\alpha \quad 1 \leq u < 2$	$\frac{\alpha}{2} u ^{\alpha-2}$
φ_{GM}	$\frac{u^2}{1+u^2}$	$\frac{1}{(1+u^2)^2}$
φ_{HL}	$\log(1+u^2)$	$\frac{1}{1+u^2}$
φ_{HS}	$2\sqrt{1+u^2} - 2$	$\frac{1}{\sqrt{1+u^2}}$
φ_{GR}	$2\log[\cosh(u)]$	$\begin{cases} 1 & u = 0 \\ \tanh(u)/u & u \neq 0 \end{cases}$

Tableau 2-1 : fonctions de potentiel et fonctions de pondération

La fonction quadratique tronquée et la fonction de Huber fournissent, par contre, des fonctions de pondération à deux états. Toutefois, ces fonctions ne respectent pas la condition iii). Cela explique l’existence de “points critiques”. En de tels points, le passage d’un état à l’autre, autrement dit la décision de lisser ou non se fait de façon abrupte. Une petite variation des données peut entraîner un changement important de la solution, ce qui va à l’encontre du concept de régularisation.

2.5.2 Fonctions de pondération utilisées

Nous représentons sur la figure 2-12 les fonctions de pondération, vérifiant toutes les conditions (2-34), que nous utiliserons par la suite.

On remarque que les fonctions de Green et des surfaces minimales fournissent des fonctions de pondération très proches l’une de l’autre (les courbes sont quasiment confondues). D’autre part, alors que les fonctions de potentiel ont des allures très diverses, on peut remarquer que ces fonctions de pondération ont le même comportement général. Pour les faibles valeurs du gradients, elles tendent vers une constante tandis que pour de forts gradients, la pondération tend vers 0 . On peut donc dire que ces fonctions sont à deux

états, l'un correspondant à l'action de lissage, l'autre permettant la préservation des discontinuités.

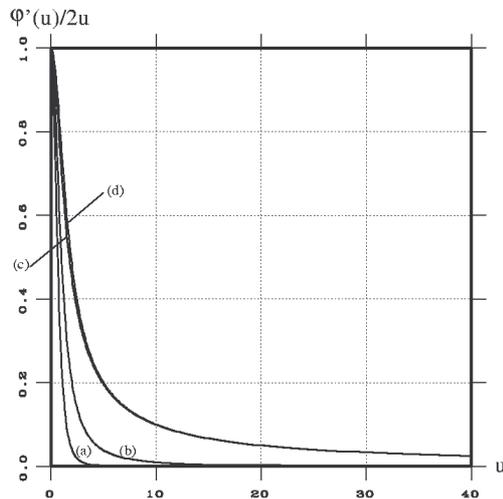


Figure 2-12: Fonction de pondération correspondant aux fonctions de Geman et McClure (a), d'Hebert et Leahy (b), des surfaces minimales (c) et de Green (d).

Cette figure confirme le fait que des fonctions telles que $\log(1+u^2)$, le log cosh ou la fonction des surfaces minimales permettent la prise en compte des discontinuités. Ceci est très important car dans le cas du log cosh et de la fonction des surfaces minimales, le critère du MAP est strictement convexe et n'a donc qu'un seul minimum. Il n'est alors pas nécessaire d'avoir recours aux coûteuses méthodes stochastiques pour l'estimation de la solution.

2.5.3 Qualité de la préservation des contours

Il faut noter que si les trois fonctions de pondération se ressemblent, elles ne décroissent pas à la même vitesse: la fonction associée au potentiel de Geman et McClure se comporte en $1/u^4$, celle associée au $\log(1+u^2)$ d'Hebert et Leahy en $1/u^2$ et celles associées au log cosh de Green et à la fonction des surfaces minimales, en $1/u$. En d'autres termes, le passage d'un état à l'autre (lissage ou non) se fait plus ou moins franchement. On peut s'attendre à obtenir des contours moins francs avec le logcosh ou la fonction des surfaces minimales qu'avec les deux autres fonctions - et c'est ce que nous observons en pratique (cf. Chapitre IV).

Ce phénomène de dégradation des contours est également observable dans l'article de Bouman et Sauer, [BOUM93], figure 7 page 303. Cette figure montre un résultat de restauration sur une impulsion carrée bruitée, en utilisant les fonction de potentiels de la forme $|u|^\alpha$. On constate que plus α se rapproche de 2, plus les contours estimés se dégradent, le résultat tendant vers une solution lisse "à la Tikhonov" pour $\alpha = 2 \dots$

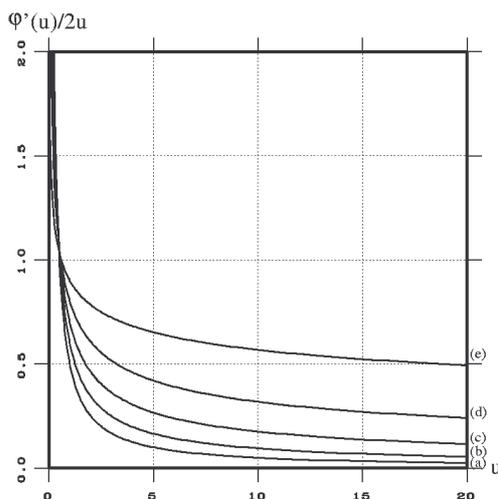


Figure 2-13: Fonction de pondération correspondant à la fonction $|u|^\alpha$ préconisée par Bouman et Sauer pour: $\alpha = 1$ (a); $\alpha = 1,2$ (b); $\alpha = 1,4$ (c); $\alpha = 1,6$ (d); $\alpha = 1,8$ (e).

Ce résultat s'explique aisément en regardant la figure 2-13 où nous représentons les fonctions de pondérations associées, pour plusieurs valeurs du paramètre α . On peut par exemple observer qu'un gradient de 20 fournira dans le cas $\alpha = 1$ une pondération quasiment nulle alors que pour $\alpha = 1,8$ on obtiendra la valeur non négligeable de 0,5. Dans le premier cas, le lissage sera interrompu, alors que dans le second cas il ne sera qu'atténué.

Remarque: nous n'utilisons pas les fonctions $|u|^\alpha$ dans la pratique, car la limite en 0 des fonctions de pondération est infinie, ce qui peut poser des problèmes pour le calcul numérique.

2.5.4 Conclusion

Ainsi, les potentiels vérifiant les trois conditions (2-34) fournissent des fonctions de pondération à deux états: l'un correspondant à l'action de lissage, l'autre à la préservation des discontinuités. Les fonctions de potentiel que nous utiliserons par la suite et les fonctions de pondération associées sont récapitulées dans le tableau 2-2. Ces fonctions

vérifient toutes les conditions (2-34). La prise en compte des discontinuités ne dépend pas de la nature convexe ou non de φ .

D'autre part, même s'il est possible de paramétrer les fonctions de pondération pour les faire mieux "se ressembler", certaines fonctions de pondérations permettent un passage plus rapide d'un état à l'autre.

Potentiel	Expression de φ	Nature de φ	Expression de $\varphi'(u)/2u$	Décroissance de $\varphi'(u)/2u$
φ_{GM}	$\frac{u^2}{1+u^2}$	Non convexe	$\frac{1}{(1+u^2)^2}$	$1/u^4$
φ_{HL}	$\log(1+u^2)$	Non convexe	$\frac{1}{1+u^2}$	$1/u^2$
φ_{HS}	$2\sqrt{1+u^2} - 2$	Convexe	$\frac{1}{\sqrt{1+u^2}}$	$1/u$
φ_{GR}	$2\log[\cosh(u)]$	Convexe	$\begin{cases} 1 & u = 0 \\ \tanh(u)/u & u \neq 0 \end{cases}$	$1/u$

Tableau 2-1 : Tableau récapitulatif des fonctions de potentiel et fonctions de pondération utilisées

On peut s'attendre à ce que certaines fonctions fournissent des estimées aux contours plus francs que d'autres. C'est en particulier le cas des fonctions non convexes telles que φ_{GM} ou φ_{HL} .

2.6. Régularisation et prise en compte des discontinuités: conclusion

A la lumière de l'étude que nous venons d'effectuer au paragraphe 1.4, il apparaît que:

“Afin d’assurer la prise en compte des discontinuités, les fonctions de potentiel doivent vérifier les conditions (2-34).”

Nous sommes en accord avec les deux approches communément utilisées: celle utilisant un processus de ligne et celles n'en utilisant pas. En effet, les conditions (2-34) sont compatibles avec les conditions citées aux paragraphes 2.2.3 et 2.3.3.

En particulier, dans le cas où φ est continûment dérivable, la condition 2 du théorème de Geman et Reynolds (concavité de $\varphi(\sqrt{u})$) est *équivalente* à la condition iii de (2-34) (continuité et décroissance stricte de la fonction de pondération).

Par contre, s'il est vrai que la condition 3 du théorème de Geman et Reynolds (existence d'une asymptote horizontale) *implique* la condition ii de (2-34), cette condition n'est pas *nécessaire*. L'hypothèse de Geman & Reynolds sur le comportement asymptotique des fonctions peut être relâchée.

Les conditions aux limites seront donc posées sur la fonction de pondération $\varphi'(u)/2u$ plutôt que sur la fonction φ elle-même. Cela étend la classe des fonctions de potentiel assurant la prise en compte des discontinuités. En particulier, nous justifions le fait suivant:

“Il est possible d'effectuer une régularisation avec prise en compte des discontinuités en utilisant des fonctionnelles *convexes*. “

Même si l'on peut s'attendre à obtenir des images aux contours moins francs en utilisant des fonctionnelles convexes, ce résultat est très important. En effet, si φ est convexe, l'énergie du MAP (2-1) n'a qu'un seul minimum. Dans ce cas, une approche déterministe se trouve pleinement justifiée. Toutefois, même dans ce cas, il reste encore un problème à résoudre: les équations normales (2-29) sont non-linéaires par rapport à f . C'est ce qui nous amène à utiliser le concept de régularisation semi-quadratique.

3. Régularisation semi-quadratique

Au paragraphe précédent, nous avons vu comment l'introduction de variables auxiliaires (le processus de ligne) permettait de justifier l'utilisation de certaines fonctions de potentiel pour la régularisation avec prise en compte des discontinuités.

Dans ce paragraphe, nous généralisons l'utilisation de telles variables afin de simplifier l'estimation de la solution du MAP.

3.1. Introduction

Le terme “régularisation semi-quadratique” a été introduit en 1993 dans [GEMA93]. Il est en fait la généralisation de travaux antérieurs, visant à faciliter l'estimation du MAP. Dans ce paragraphe, nous donnons un aperçu (non exhaustif) de ces travaux.

3.1.1 Utilisation du processus de ligne Booléen pour l'estimation du MAP

L'idée d'utiliser le processus de ligne pour l'estimation au sens du MAP date de 1984, année de parution de l'article des frères Geman [GEMA84] introduisant le processus de

ligne. Dès cette époque, certains auteurs (voir par exemple [MARR84]) font la constatation suivante:

lorsque le processus de ligne (Booléen) est fixé, le critère devient convexe par rapport à f .

Plutôt que d'opérer un recuit simulé sur la variable f , comme dans [GEMA84], Marroquin propose un algorithme dit "de recuit mixte". Cet algorithme opère de la façon suivante: à chaque température, on parcourt séquentiellement les sites d'intensité et les sites de processus de ligne. Quand un site d'intensité est visité, on minimise le critère en f en fixant les valeurs du processus de ligne. Quand un site de processus de ligne est visité, il est mis à jour selon l'algorithme de Métropolis. Aucune preuve de convergence n'est fournie, mais l'algorithme donne de bons résultats [MARR84].

En 1985, D. Terzopoulos (utilisant une modélisation déterministe où la notion de processus de ligne est remplacée par la notion de fonction de contrôle de continuité, voir aussi [MARC92]), propose un algorithme déterministe basé sur la même constatation [TERZ86]. A chaque étape, on minimise le critère en fixant la variable de ligne, puis on fixe f . Ensuite, on recalcule les nouvelles valeurs de la variable auxiliaire (aucun détail n'est donné sur ce dernier calcul). Là encore, il n'y a pas de preuve de convergence, mais l'auteur assure que les résultats obtenus sont proches du minimum global.

En 1989, Geiger et Girosi proposent l'algorithme MFA (voir [GEIG91]). Cet algorithme de recuit déterministe est basé sur des estimations au sens du champ moyen, mais fournit une estimée au sens du MAP lorsque la température tend vers 0. A chaque température, la nouvelle valeur du champ (moyen) d'intensité est calculée en fixant la valeur (moyenne) du processus de ligne. Ensuite, la nouvelle valeur du processus de ligne moyen est calculée à l'aide de la nouvelle valeur du processus d'intensité. En cours de calcul, le processus de ligne moyen est à valeurs continues, mais il tend vers le processus de ligne Booléen lorsque la température tend vers 0.

3.1.2 Utilisation de variables auxiliaires réelles pour l'estimation du MAP

Dans [GEMA92], D. Geman et G. Reynolds utilisent un processus de ligne continu. Comme dans le cas du processus de ligne Booléen, ils remarquent que le critère est quadratique en f lorsque b est fixé, ou de façon équivalente, que la probabilité $P(f|b)$ est Gaussienne. Cette propriété pourrait être utilisée pour définir un algorithme rapide d'échantillonneur de Gibbs avec recuit [GEMA93]. En effet, la lenteur de l'algorithme de

recuit sur f (décrit au Chapitre I) s'explique en partie par le fait que l'on doit modifier les valeurs de pixel une à une en faisant un tirage aléatoire selon la loi locale. Un algorithme effectuant le tirage d'un échantillon, c'est à dire de toute une image, en une seule opération serait donc plus rapide. Malheureusement, cela n'est pas possible dans le cas d'une distribution locale de Gibbs. Par contre, un tel algorithme peut être mis au point en utilisant le processus de ligne. L'algorithme de recuit rapide fonctionnerait de la façon suivante: à chaque balayage, on tire un échantillon en f à b fixé, puis un échantillon de b à f fixé.

Dans ce cas, la loi de probabilité $P(f|b)$ utilisée pour le tirage d'un échantillon de f à b fixé est Gaussienne. Or, une loi Gaussienne se déduit facilement d'une loi uniforme. Pour peu que la matrice de corrélation associée à la distribution de probabilité Gaussienne soit circulante, ce calcul peut s'effectuer facilement dans l'espace de Fourier. Le tirage d'un échantillon de f s'effectue donc en une seule opération à partir d'un échantillon de N^2 variables aléatoires uniformes et indépendantes.

D'autre part, lorsque f est fixé, les $(b_x)_k$ sont indépendants entre eux, et les $(b_y)_k$ également. Des tables de la fonction de répartition de $P(b|f)$ peuvent être calculées par avance. Un échantillon en b peut alors se calculer globalement à partir d'un tirage aléatoire de N^2 variables uniformes et indépendantes, en utilisant les tables précalculées.

L'algorithme de recuit rapide permettrait donc le calcul global des échantillons de f et de b , à partir d'échantillons de variables aléatoires uniformes et indépendantes (qui sont faciles à générer).

Malheureusement, dans le cas du processus de ligne, tel que défini dans [GEMA92], la matrice de corrélation de la loi Gaussienne $P(f|b)$ n'est pas circulante. Le tirage de l'échantillon de f ne peut s'effectuer de façon rapide dans le domaine de Fourier. C'est pourquoi cette méthode n'est pas viable *dans le cas du processus de ligne tel que défini dans [GEMA92]*.

Par contre, dans [GEMA93], D. Geman et C. Yang définissent une autre variable auxiliaire, marquant également les discontinuités, et telle que la matrice de corrélation associée à $P(f|b)$ soit bloc-circulante. Dans ce cas, le recuit simulé en f et b peut se mettre en œuvre sous forme d'algorithme rapide, chaque tirage d'échantillon se faisant de façon globale. Outre la méthode d'optimisation elle-même, le point fort de cet article est la généralisation du concept du processus de ligne à celui de variable auxiliaire.

Le rôle de cette variable n'est plus seulement de marquer les discontinuités, mais aussi de faciliter la minimisation du critère du MAP en rendant le problème *semi-quadratique*.

3.1.3 Principe de la régularisation semi-quadratique

L'expression "régularisation semi-quadratique" a été introduite en 1993 par D. Geman et C. Yang [GEMA93], qui la définissent de la façon suivante:

"introduire une nouvelle fonction d'énergie qui, bien que définie sur un domaine plus étendu, a le même minimum global en f que $[J_2]$, et peut être manipulée avec des méthodes d'algèbre linéaire."

L'idée est donc d'introduire des variables auxiliaires b_x et b_y de façon à rendre le critère plus facile à manipuler.

Sous certaines conditions (cf. §3.2), on peut écrire:

$$\varphi(u) = \inf_b \left\{ \varphi^*(u, b) \right\} \quad (2-35)$$

En utilisant ce résultat avec $u = (D_x f)_k$ et $b = (b_x)_k$ (respectivement avec $u = (D_y f)_k$ et $b = (b_y)_k$), le critère du MAP s'écrit alors:

$$J(f) = \inf_{b_x, b_y} J^*(f, b_x, b_y) \quad (2-36)$$

et la fonction d'énergie duale J^* est de la forme:

$$J^*(f, b_x, b_y) = \|p - Rf\|^2 + \lambda^2 \sum_k \varphi^*[(D_x f)_k, (b_x)_k] + \lambda^2 \sum_k \varphi^*[(D_y f)_k, (b_y)_k] \quad (2-37)$$

La transformation (2-35) s'effectue de façon à rendre le critère quadratique lorsque l'on fixe b_x et b_y . C'est ce qui donne son nom de "semi-quadratique" à ce type de régularisation. Par conséquent, lorsque l'on fixe les variables auxiliaires, les équations normales à résoudre sont linéaires. On peut énoncer cette propriété sous une autre forme en disant que la distribution $P(f|b)$ est Gaussienne [GEMA93].

3.2. Définition de la variable auxiliaire

Dans ce paragraphe, nous étudions le passage du problème non-quadratique de minimisation en f au problème semi-quadratique. Nous reprenons en partie les théorèmes de Geman et Reynolds [GEMA92] et de Geman et Yang [GEMA93]. Toutefois, nous posons des conditions plus strictes sur la fonction de potentiel. Il s'en suit que nous restreignons un peu le champ d'application de ces théorèmes. Par contre cette restriction

nous permet d'ajouter une deuxième partie à ces théorèmes, donnant l'expression de b à f fixé. Ce résultat a un rôle très important, c'est lui qui nous permet de définir les algorithmes ARTUR et LEGEND, proposés au Chapitre III.

3.2.1 Première approche: "théorème de Geman & Reynolds étendu"

La première approche, proposée dans [AUBE94] est basée sur le théorème 1 de Geman et Reynolds [GEMA92]. Nous relâchons l'hypothèse de l'existence d'une asymptote horizontale, ce qui permet d'utiliser une classe plus importante de fonctions incluant des potentiels convexes. Ce faisant, comme nous le remarquons dans l'introduction de ce paragraphe, nous restreignons un peu le théorème en supposant la stricte concavité de $\varphi(\sqrt{u})$.

Théorème 2:

Soit $\varphi:]0, +\infty[\rightarrow]0, +\infty[$, telle que $\varphi(\sqrt{u})$ soit strictement concave sur $]0, +\infty[$. Soient L et M définis comme:

$$L = \lim_{u \rightarrow +\infty} \frac{\varphi'(u)}{2u} \text{ et } M = \lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u}$$

Alors:

1) Il existe une fonction $\psi: [L, M] \rightarrow [\alpha, \beta]$ strictement convexe et décroissante telle que:

$$\varphi(u) = \inf_{L \leq b \leq M} (bu^2 + \psi(b)) \tag{2-38}$$

avec :

$$\alpha = \lim_{u \rightarrow +\infty} \left(\varphi(u) - u^2 \frac{\varphi'(u)}{2u} \right) \text{ et } \beta = \lim_{u \rightarrow 0^+} \varphi(u)$$

2) Pour tout $u \geq 0$ fixé, la valeur b_u pour laquelle l'inf est atteint:

$$\inf_{L \leq b \leq M} (bu^2 + \psi(b)) = (b_u u^2 + \psi(b_u))$$

est unique et donnée par:

$$b_u = \frac{\varphi'(u)}{2u}.$$

Une démonstration de ce théorème est donnée en annexe B. Au passage, on trouvera dans cette démonstration l'expression littérale de $\psi(b)$. Bien qu'elle ne soit pas nécessaire pour la définition des algorithmes présentés au chapitre suivant, nous donnons pour mémoire

(cf. tableau 2-3) l'expression obtenue pour les différents potentiels utilisés, sauf la fonction $\log \cosh$ pour laquelle on ne trouve pas d'expression analytique.

$\varphi(u)$	$\psi(b)$
$\log(1+u^2)$	$b-\log(b)-1$
$\frac{u^2}{1+u^2}$	$b-2\sqrt{b}+1$
$2\sqrt{1+u^2}-2$	$b+\frac{1}{b}-2$

Tableau 2-3 : expression $\psi(b)$ de pour les potentiels utilisés

La conséquence de l'hypothèse de stricte concavité de $\varphi(\sqrt{u})$ est double. D'une part, ce théorème ne peut pas s'appliquer à la fonction quadratique tronquée ni à la fonction de Huber qui ne satisfont pas les hypothèses du théorème. La variable auxiliaire ne peut être Booléenne et correspond au processus de ligne continu tel que défini au paragraphe 2.2.3.

D'autre part, à la différence du théorème 1 de Geman et Reynolds [GEMA92], on a ici une expression littérale du processus de ligne à u fixé. On remarque que cette expression correspond à celle de la fonction de pondération, définie au paragraphe 2.4.1.

Dans ces travaux, nous nous intéressons aux fonctions vérifiant les conditions (2-34). Il est intéressant de noter que les conditions du théorème 2, avec $L=0$ et $M=1$, sont équivalentes aux conditions (2-34). En effet, il y a équivalence entre la concavité d'une fonction et la stricte décroissance de sa dérivée. Or, la dérivée de $\varphi(\sqrt{u})$ est la fonction de pondération.

Le processus de ligne est donc à valeurs sur $[0,1]$ et tel que:

- en présence d'un fort gradient (discontinuité), b est proche de 0,
- en l'absence de discontinuité, la variable b tend vers 1,

ce qui correspond bien à nos conventions de départ. La minimisation en f et en b correspondra à l'estimation conjointe de la solution et de la carte du processus de ligne associée à cette image.

Remarque: dans le cas de la quadratique tronquée, l'expression (2-38) reste valable, avec $\psi(b)=1-b$. Le minimum en b est dégénéré et on voit facilement qu'à u fixé: $b=1$ pour $u < 1$ et que $b=0$ pour $u > 1$. Par contre, la valeur de b est indéfinie pour $u=1$.

3.2.2 Deuxième approche: utilisation des paires de Legendre

La seconde approche, plus récente, a été introduite dans [GEMA93]. Elle repose sur l'utilisation d'un outil classique d'analyse convexe [ROCK]: les paires de Legendre. Dans ce cas également, nous posons des conditions plus strictes que dans [GEMA93].

Théorème 3:

Soit $\varphi: [0, +\infty[\rightarrow [0, +\infty[$, telle que $\frac{1}{2}(u^2 - \varphi(u))$ soit strictement convexe sur $]0, +\infty[$. Alors:

1) Il existe une fonction $\zeta: [0, +\infty[\rightarrow [0, +\infty[$ telle que:

$$\varphi(u) = \inf_b \left((b-u)^2 + \zeta(b) \right) \tag{2-39}$$

2) Si $\lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u}$ existe, alors pour tout $u \geq 0$ fixé la valeur b_u pour laquelle l'inf est atteint:

$$\inf_b \left((b-u)^2 + \zeta(b) \right) = \left((b_u - u)^2 + \zeta(b_u) \right)$$

est unique et donnée par:

$$b_u = \left(1 - \frac{\varphi'(u)}{2u} \right) u.$$

Une démonstration de ce théorème est donnée en annexe C. On pourra remarquer que l'expression de la fonction ζ : $\zeta(b) = \sup_u \left[-(u-b)^2 + \varphi(u) \right]$ n'est pas directement exploitable analytiquement.

Comme dans le théorème précédent, c'est le fait de poser une hypothèse *stricte* sur $\frac{1}{2}(u^2 - \varphi(u))$ qui nous permet de donner l'expression littérale de b à u fixé. On peut montrer (voir Annexe C, paragraphe 4) que cette condition de convexité stricte et que la condition 2) sont automatiquement remplies, dès lors que la fonction de potentiel vérifie les conditions (2-34).

A la différence du cas précédent, les valeurs de la variable auxiliaire ne seront plus bornées sur l'intervalle $[0,1]$. Le comportement de b est le suivant:

- en présence d'un fort gradient, b se comporte comme u ,
- en l'absence de discontinuité, la variable b a une valeur proche de 0.

La minimisation en f et en b correspondra également à une estimation conjointe de la solution et de la carte des discontinuités associée à cette image.

3.2.3 Définition des énergies duales

Dans les deux paragraphes précédents, nous avons vu comment passer d'un critère non-quadratique à un critère semi-quadratique. Nous avons fait remarquer que lorsque les conditions (2-34) sont vérifiées, alors les conditions des théorèmes 2 et 3 le sont également. Autrement dit:

“lorsque la fonction de potentiel respecte les discontinuités alors on peut introduire, de façon additive ou multiplicative, une variable auxiliaire permettant de rendre le critère semi-quadratique”.

Ceci est résumé dans le tableau 2-4.

Conditions sur φ	i) $\lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u} = 1$	ii) $\lim_{u \rightarrow +\infty} \frac{\varphi'(u)}{2u} = 0$
	iii) $\frac{\varphi'(u)}{2u}$ est continue et strictement décroissante sur $[0, +\infty[$	
Définition de φ^*	$\varphi^*(u, b) = bu^2 + \psi(b)$ $b \in [0, 1[$	$\varphi^*(u, b) = (bu)^2 + \zeta(b)$ $b \in \mathfrak{R}^+$

Tableau 2-4 : Du non quadratique au semi-quadratique

En appliquant ces résultats au critère du MAP:

$$J(f) = \|p - Rf\|^2 + \lambda^2 \sum_k \varphi(D_x f)_k + \lambda^2 \sum_k \varphi(D_y f)_k \quad (2-40)$$

avec $u = (D_x f)_k$ et $b = (b_x)_k$ (respectivement avec $u = (D_y f)_k$ et $b = (b_y)_k$) nous obtenons dans le premier cas le critère semi-quadratique suivant:

$$J^*(f, b_x, b_y) = \|p - Rf\|^2 + \lambda^2 \sum_k \left\{ (b_x)_k (D_x f)_k^2 + \psi[(b_x)_k] \right\} + \lambda^2 \sum_k \left\{ (b_y)_k (D_y f)_k^2 + \psi[(b_y)_k] \right\} \quad (2-41)$$

et dans le second cas:

$$J^*(f, b_x, b_y) = \|p - Rf\|^2 + \lambda^2 \sum_k \left\{ (b_x - D_x f)_k^2 + \zeta[(b_x)_k] \right\} + \lambda^2 \sum_k \left\{ (b_y - D_y f)_k^2 + \zeta[(b_y)_k] \right\} \quad (2-42)$$

3.3. Propriétés de la régularisation semi-quadratique

Dans ce paragraphe, nous rappelons les deux types de propriétés très importantes de la régularisation semi-quadratique telle que nous venons de la définir.

3.3.1 Propriété à b fixé

Comme annoncé au §3.1.3, on constate bien, dans (2-41) et (2-42), que si b_x et b_y sont fixés, le critère $J^*(f, b_x, b_y)$ est quadratique en f . La conséquence de cette propriété est que la minimisation en f à b fixé est particulièrement simple: les équations normales à résoudre sont linéaires.

On peut remarquer d'une façon équivalente (cf. [GEMA92] et [GEMA93]) que la distribution a posteriori $P(f|b)$ est Gaussienne.

3.3.2 Propriété à f fixé

Dans le premier cas, la fonction ψ est convexe en b . Lorsque l'on fixe f , le critère est donc convexe par rapport aux variables auxiliaires.

Théorème	Expression de b pour f fixé
« Geman et Reynolds étendu »	$b = \frac{\varphi'(u)}{2u}$
« Paires de Legendre »	$b = \left(1 - \frac{\varphi'(u)}{2u}\right)u$

Tableau 2-5 : Expression de b pour f fixé

Dans le second cas, la fonction ζ n'est pas forcément convexe. Toutefois, on montre tout de même (voir annexe C 2) que le critère est convexe en (b_x, b_y) à f fixé.

Dans les deux cas, le couple minimisant $J^*(f, b_x, b_y)$ à f fixé est donc unique et on a une expression littérale permettant de calculer directement ses valeurs (cf. tableau 2-5). Cette

propriété est très importante. Elle est à la base des algorithmes ARTUR et LEGEND qui seront présentés au Chapitre III.

4. Conclusion

Dans la première partie de ce chapitre, nous nous sommes intéressés à la régularisation avec prise en compte des discontinuités, en nous demandant à quelles conditions doit obéir la fonction de potentiel. Nous avons présenté les différentes approches existantes et proposé une étude heuristique des équations normales. Cette étude nous a permis d'exhiber les trois propriétés (2-34) garantissant la préservation des discontinuités. Ces conditions ne portent en fait pas sur la fonction de potentiel $\varphi(u)$, mais sur la *fonction de pondération* $\varphi'(u)/2u$. Elles sont compatibles avec toutes les approches existantes. Un point important est que ces conditions ne préjugent pas de la nature convexe ou non de la fonction de potentiel. Même si on peut s'attendre à ce que les discontinuités obtenues soient moins franches lorsqu'on utilise un potentiel convexe, le minimum du MAP est unique, ce qui justifie pleinement l'usage d'algorithmes déterministes.

Comme nous l'avons fait remarquer à plusieurs reprises, le problème de l'estimation du MAP dans le cadre de la régularisation avec prise en compte des discontinuités est difficile à résoudre. En effet, même s'il peut être convexe, le critère à minimiser est dans tous les cas non quadratique. Nous avons présenté dans ce paragraphe la régularisation *semi-quadratique* (selon le terme introduit dans [GEMA93]).

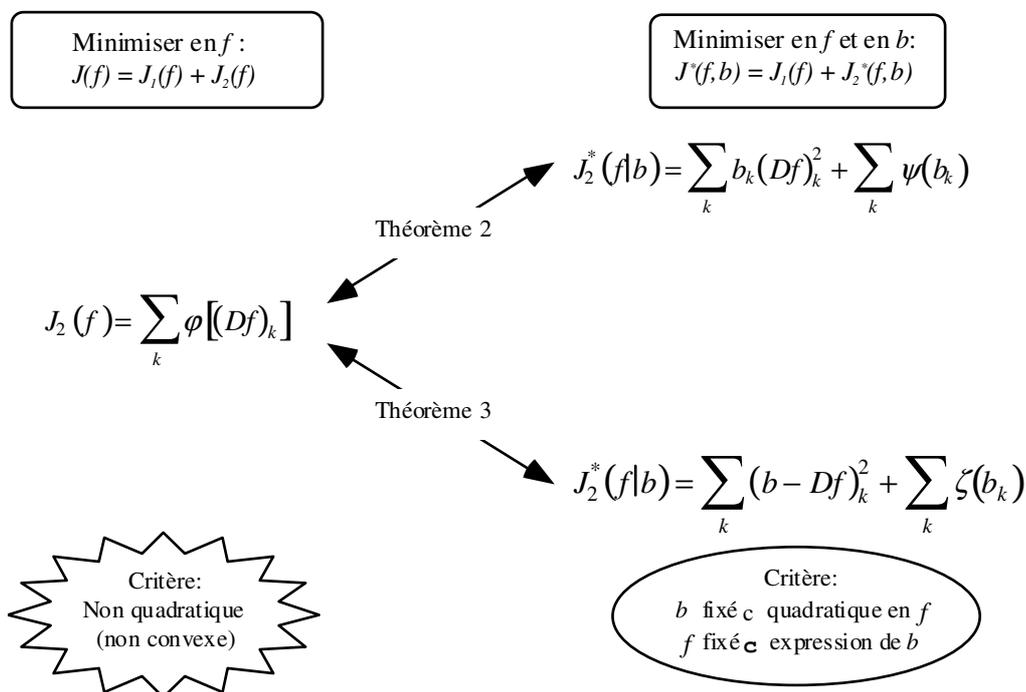


Figure 2-14: La régularisation semi-quadratique

Lorsque les conditions (2-34) sont vérifiées, on peut découpler le problème en introduisant soit de manière additive, soit de manière multiplicative, une variable auxiliaire b . Le rôle de cette variable est double: *marquer les discontinuités* présentes dans l'image de manière à les préserver du lissage, et *simplifier l'estimation* de la solution. En effet, en fixant b , le critère devient quadratique en f . D'autre part, à f fixé, nous remarquons que le critère est convexe en b et nous donnons l'expression littérale permettant de calculer son minimum. Ainsi, bien que l'on ait apparemment augmenté la difficulté du problème en introduisant de nouvelles inconnues, le nouveau critère à minimiser a des propriétés intéressantes. La question est maintenant de savoir comment on peut exploiter ces propriétés.

Pour notre part, nous adoptons un approche déterministe. D'une part, les applications visées, notamment la reconstruction tomographique, nécessitent un traitement rapide des données. D'autre part, comme nous l'avons fait remarquer précédemment, la régularisation avec prise en compte des discontinuités peut se mettre en œuvre à l'aide de fonctions convexes. Dans ce cas, le minimum du critère du MAP est unique, et une approche déterministe se trouve pleinement justifiée. Mettant à profit le fait que la minimisation du critère semi-quadratique est facile lorsque l'on fixe soit la variable principale, soit la variable auxiliaire, nous proposons au chapitre suivant deux algorithmes déterministes basés sur des minimisation alternées en f et en b .

Chapitre III

Deux algorithmes déterministes de relaxation: ARTUR et LEGEND

1. Introduction

Au Chapitre II, nous nous sommes intéressés dans un premier temps à la définition d'un modèle *a priori* pour la régularisation avec prise en compte des discontinuités. Ce type de régularisation implique la minimisation d'une fonctionnelle d'énergie $J(f)$ non quadratique et éventuellement non convexe. Nous avons ensuite vu comment on peut introduire une variable auxiliaire $b = (b_x, b_y)$, soit de façon multiplicative (théorème 2), soit de façon additive (théorème 3). Le rôle de cette variable est double: marquer les discontinuités et rendre le problème plus facile à résoudre. On doit alors minimiser la fonctionnelle duale $J^*(f, b_x, b_y)$. Cette énergie possède deux propriétés importantes:

- elle est quadratique en f à b fixé (c'est ce qui donne son nom de semi-quadratique à ce type de régularisation)
- elle est convexe en b à f fixé (et on a même une expression littérale du b réalisant le minimum).

Il reste maintenant à savoir comment on peut exploiter ces propriétés de façon algorithmique. Dans [GEMA93], Geman et Yang ont proposé un algorithme stochastique de reconstruction. Cet algorithme utilise la régularisation semi-quadratique (dans le cas additif) pour accélérer notablement un algorithme de type échantillonneur de Gibbs avec recuit. Toutefois, on ne peut pas utiliser la variable auxiliaire dans le cas multiplicatif avec des algorithmes stochastiques

[GEMA92]. D'autre part, les algorithmes stochastiques demeurent tout de même très lourds d'utilisation, et leur usage ne se justifie pas dans le cas d'énergies convexes.

Pour notre part, nous avons choisi d'utiliser une approche déterministe. Nous présentons dans ce chapitre deux algorithmes de reconstruction, ARTUR [CHAR93A] et LEGEND [CHAR94B], basés sur le même principe. La différence entre ces deux algorithmes tient à la façon dont la variable auxiliaire est utilisée: de façon *multiplicative* pour ARTUR, de façon *additive* pour LEGEND.

Nous donnons dans un premier temps la forme générale de ces algorithmes, puis en étudions la convergence, et donnons des détails utiles pour leur mise en œuvre. Ensuite, nous situons ARTUR et LEGEND par rapport à d'autres algorithmes déterministes existant. Enfin, nous élargissons notre réflexion à d'autres techniques possibles de minimisation de l'énergie duale.

2. Forme générale des algorithmes ARTUR et LEGEND

2.1. Stratégie de minimisation

La minimisation de l'énergie duale $J^*(f, b_x, b_y)$ est un problème délicat du fait que ce critère n'est pas quadratique et peut de plus ne pas être convexe. La stratégie de minimisation que nous adoptons est la suivante: plutôt que de minimiser directement J^* , on minimise par étapes une suite d'énergies plus faciles à manipuler. Ces énergies sont définies en fixant alternativement la valeur de la variable principale f , puis de la variable auxiliaire (b_x, b_y) . On obtient alors le schéma présenté figure 3-1.

$$f^0 \equiv 0$$

Répéter

1)	$(b_x^{n+1}, b_y^{n+1}) = \arg \min_{b_x, b_y} [J^*(f^n, b_x, b_y)]$
2)	$f^{n+1} = \arg \min_f [J^*(f, b_x^{n+1}, b_y^{n+1})]$

Jusqu'à convergence

Figure 3-1: Principe des minimisations alternées

A chaque étape, les pas 1) et 2) sont assez faciles à effectuer car ils mettent en jeu des minimisations de critères convexes. Une stratégie de minimisation semblable est utilisée dans [TERZ86] (aucun détail d'implantation n'est cependant fourni) et dans [DINT90].

Nous remarquons qu'à chaque étape, f^{n+1} n'est pas obtenue explicitement en fonction de f^n et de (b_x^{n+1}, b_y^{n+1}) , mais par minimisation d'une énergie. En d'autres termes, il est nécessaire de résoudre un système d'équations (les équations normales associées à $J^*(f, b_x^{n+1}, b_y^{n+1})$) pour obtenir la nouvelle estimée. C'est pourquoi ce schéma est dit *implicite*¹.

Étant données les propriétés mises en évidence au paragraphe 2.3 du Chapitre II, l'application de ce schéma nous conduit naturellement à deux algorithmes déterministes pour l'estimation du MAP.

2.2. Présentation des algorithmes

Les algorithmes ARTUR et LEGEND que nous proposons utilisent le principe des minimisations alternées présenté sur la figure 3-1. Appliquons maintenant les propriétés de la régularisation semi-quadratique, introduites au Chapitre II.

2.2.1. Expressions littérales de b^{n+1} à f^{n+1} fixé

D'après le Chapitre II, on sait qu'à u fixé, la valeur de b recherchée est donnée par une expression littérale (cf. théorèmes 2 et 3). En appliquant ces résultats en chaque pixel, c'est à dire:

$$\begin{aligned} & \text{- d'une part avec } b = (b_x^{n+1})_k \text{ et } u = (D_x f^n)_k \\ & \text{- d'autre part avec } b = (b_y^{n+1})_k \text{ et } u = (D_y f^n)_k, \end{aligned}$$

nous obtenons les expressions résumées dans le tableau 3-1.

¹ Ne pas confondre le terme *schéma implicite* avec le terme *modèle implicite* employé au Chapitre II. Cette dernière expression fait référence à l'existence d'un processus de ligne n'apparaissant pas explicitement dans le *modèle d'image* considéré, et n'a rien à voir avec l'algorithme d'optimisation utilisé.

Algorithme	Expression de b_x^{n+1}	Expression de b_y^{n+1}
ARTUR	$(b_x^{n+1})_k = \frac{\varphi'[(D_x f^n)_k]}{2(D_x f^n)_k}$	$(b_y^{n+1})_k = \frac{\varphi'[(D_y f^n)_k]}{2(D_y f^n)_k}$
LEGEND	$(b_x^{n+1})_k = \left\{ 1 - \frac{\varphi'[(D_x f^n)_k]}{2(D_x f^n)_k} \right\} (D_x f^n)_k$	$(b_y^{n+1})_k = \left\{ 1 - \frac{\varphi'[(D_y f^n)_k]}{2(D_y f^n)_k} \right\} (D_y f^n)_k$

Tableau 3-1: Calcul de b^{n+1} (à f^n fixé)

Ainsi, à l'étape n+1, les valeurs des variables auxiliaires minimisant $J^*(f^n, \bullet)$ se calculent directement à partir d'une expression littérale.

2.2.2. Calcul de f^{n+1} à b^{n+1} fixé

Tout d'abord, nous réécrivons les expressions des énergies duales à b^{n+1} fixée. Dans le cas d'ARTUR, l'énergie duale peut être mise sous la forme:

$$J^*(f, b_x^{n+1}, b_y^{n+1}) = \|p - Rf\|^2 + \lambda^2 \|D_x f\|_{B_x^{n+1}}^2 + \lambda^2 \|D_y f\|_{B_y^{n+1}}^2 + Cte(b^{n+1}) \quad (3-1)$$

avec:

$$B_x = \text{diag}[(b_x)_k] \text{ et } B_y = \text{diag}[(b_y)_k]$$

$$Cte(b^{n+1}) = \lambda^2 \sum_k \left\{ \psi[(b_x^{n+1})_k] + \psi[(b_y^{n+1})_k] \right\}$$

et:

On posera $\Delta_A^{n+1} = -D_x^t B_x D_x - D_y^t B_y D_y$. La matrice Δ_A^{n+1} est le Laplacien pondéré discret correspondant à l'algorithme ARTUR.

Dans le cas de LEGEND, on peut écrire à b^{n+1} fixé:

$$J^*(f, b_x^{n+1}, b_y^{n+1}) = \|p - Rf\|^2 + \lambda^2 \|b_x^{n+1} - D_x f\|^2 + \lambda^2 \|b_y^{n+1} - D_y f\|^2 + Cte(b^{n+1}) \quad (3-2)$$

$$Cte(b^{n+1}) = \lambda^2 \sum_k \left\{ \zeta[(b_x^{n+1})_k] + \zeta[(b_y^{n+1})_k] \right\}$$

avec dans ce cas:

On posera $\Delta = -D_x^t D_x - D_y^t D_y$. La matrice Δ est une approximation discrète du Laplacien.

La variable auxiliaire b^{n+1} étant fixée, l'énergie duale est quadratique en f dans les deux cas. Nous calculons f^{n+1} en résolvant les équations normales données par:

$$\left. \frac{\partial J^*(f, b_x^{n+1}, b_y^{n+1})}{\partial f} \right|_{f^{n+1}} = 0 \quad (3-3)$$

On obtient par un calcul élémentaire les expressions du tableau 3-1.

Algorithme	Équations Normales à résoudre
ARTUR	$(R^t R - \lambda^2 \Delta_A^{n+1}) f^{n+1} = R^t p$
LEGEND	$(R^t R - \lambda^2 \Delta) f^{n+1} = R^t p + \lambda^2 D_x^t b_x^{n+1} + \lambda^2 D_y^t b_y^{n+1}$

Tableau 3-2: Calcul de f^{n+1} (b^{n+1} fixé)

Remarque: dans le cas de LEGEND, il est possible de faire intervenir un Laplacien pondéré dans l'expression des équations normales en posant $\Delta_L^{n+1} = -D_x^t C_x^{n+1} D_x - D_y^t C_y^{n+1} D_y$. On a alors:

$$(R^t R - \lambda^2 \Delta) f^{n+1} = R^t p - \lambda^2 \Delta_L^{n+1} f^n \quad (3-4)$$

Les pondérations du Laplacien Δ_L^{n+1} sont alors données par:

$$C_x^{n+1} = \text{diag} \left\{ 1 - \frac{\phi' \left[(D_x f^n)_k \right]}{2 (D_x f^n)_k} \right\} \quad \text{et} \quad C_y^{n+1} = \text{diag} \left\{ 1 - \frac{\phi' \left[(D_y f^n)_k \right]}{2 (D_y f^n)_k} \right\} \quad (3-5)$$

2.2.3. Forme générale des algorithmes

En appliquant les propriétés de la régularisation semi-quadratique, nous avons donc obtenu:

- f^n étant fixé b_x^{n+1} et b_y^{n+1} peuvent être calculés directement grâce aux expressions du tableau 3-1,
- b_x^{n+1} et b_y^{n+1} étant fixés, f^{n+1} est solution des équations normales linéaires données dans le tableau 3-2.

Dans ces conditions, la forme générale des algorithmes ARTUR et LEGEND devient très simple (cf. figure 3-2).

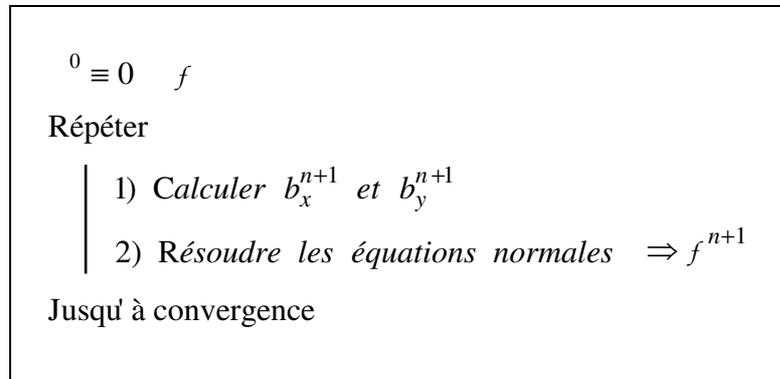


Figure 3-2: Forme générale des algorithmes ARTUR et LEGEND

Dans le paragraphe 4, nous donnons plus de détails pratiques sur l'implantation de ces algorithmes.

2.3. Remarques sur les algorithmes

2.3.1. De l'aspect auto-adaptatif des algorithmes

Les algorithmes ARTUR et LEGEND ont une forme très proche. On peut les interpréter de la façon suivante: plutôt que d'essayer de résoudre le système *non linéaire* d'équations normales associées à l'énergie du MAP $J(f)$, on préfère résoudre une *suite de systèmes linéaires* de la forme $Af = c$. Cette suite de systèmes linéaires est définie grâce à l'introduction d'une variable auxiliaire b . Elle évolue au fur et à mesure des mises à jour de b .

Un point commun aux deux algorithmes est le suivant: à chaque étape, l'expression de la variable auxiliaire (cf. tableau 3-1) dépend directement de l'estimée à l'étape précédente. L'évolution de la suite de systèmes linéaires dépend donc directement de l'évolution de l'estimée. En ce sens, on peut dire qu'ARTUR et LEGEND sont des algorithmes *auto-adaptatifs*.

La différence essentielle entre les deux algorithmes réside dans la façon dont évolue la suite de systèmes linéaires à travers les étapes: dans l'algorithme ARTUR, on fait varier la matrice A, alors que dans le cas de LEGEND, c'est le second membre qui évolue (cf. figure 3-3). Dans le cas où l'on utilise un potentiel convexe, ces deux algorithmes convergent vers le minimum global du critère et fournissent donc la même solution (cf. §3). On est alors en droit de s'interroger sur la rapidité relative des deux algorithmes.

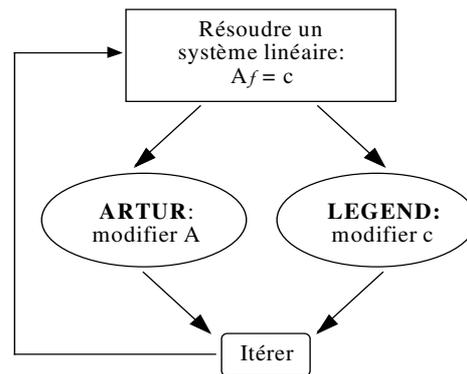


Figure 3-3: Interprétation des algorithmes ARTUR et LEGEND

Dans le cas de l'algorithme LEGEND, la matrice A ne varie pas au fil des étapes et peut donc être factorisée de façon préalable (cf. §4.1.3). Chaque résolution de système linéaire devient alors très rapide. On pourrait donc s'attendre à ce que LEGEND soit plus rapide qu'ARTUR. Toutefois, comme on modifie le second membre de l'équation, alors que dans ARTUR on modifie les coefficients du système linéaire, on peut penser intuitivement que LEGEND va converger plus lentement. C'est d'ailleurs ce que nous observons en pratique: il faut plus d'itérations à LEGEND pour fournir le même résultat qu'ARTUR. Il n'est donc pas évident de désigner *a priori* l'algorithme le plus rapide des deux. Une expérience comparative est proposée dans le Chapitre IV.

2.3.2. De l'introduction progressive des discontinuités

Un autre point commun important entre les deux algorithmes est leur comportement à la première étape. Prenons comme estimée initiale une image nulle, comme proposé au §2.2. Étant données les hypothèses faites sur la fonction de potentiel, les valeurs de b_x^1 et de b_y^1 en tout point (cf. tableau 3-1) sont égales:

- à 1 dans le cas d'ARTUR,
- à 0 dans le cas de LEGEND.

L'expression du critère $J^*(f, b_x^1, b_y^1)$ est alors, pour les deux algorithmes, de la forme suivante:

$$J^*(f, b_x^1, b_y^1) = \|p - Rf\|^2 + \lambda^2 \|D_x f\|^2 + \lambda^2 \|D_y f\|^2 + Cte \quad (3-6)$$

On reconnaît un critère d'énergie correspondant à une régularisation de Tikhonov. Autrement dit, à la première étape, les deux algorithmes calculent une solution lisse². Par la suite (cf. Chapitre IV, partie expérimentale), ils introduisent progressivement des discontinuités au niveau de la solution. En ce sens, on peut dire qu'ARTUR et LEGEND sont des algorithmes à *introduction progressive de discontinuités*.

3. Convergence des algorithmes ARTUR et LEGEND

Tout d'abord, nous rappelons les conditions de validité des algorithmes. La fonction de potentiel φ doit vérifier les conditions 2-32 définies au Chapitre II:

$$\begin{aligned} \text{i) } & \lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u} = 1 \\ \text{ii) } & \lim_{u \rightarrow +\infty} \frac{\varphi'(u)}{2u} = 0 \\ \text{iii) } & \frac{\varphi'(u)}{2u} \text{ est continue et strictement décroissante sur } [0, +\infty[. \end{aligned}$$

Ces conditions sont des conditions *suffisantes*. Ce que nous écrivons par la suite peut être vrai pour d'autres fonctions, mais cela requiert une étude particulière.

3.1. Justification de la stratégie utilisée

Le problème de minimisation du critère du MAP consiste à trouver une estimée \hat{f} telle que:

$$\hat{f} = \arg \min_f J(f) \quad (3-7)$$

Au Chapitre II, nous avons vu comment transformer le problème du MAP en un problème semi-quadratique en posant:

$$\min_f J(f) = \min_f \min_{b_x, b_y} J^*(f, b_x, b_y) \quad (3-8)$$

On sait que l'on peut changer l'ordre des minimisations en f et $b = (b_x, b_y)$. On a donc:

$$\min_f J(f) = \min_f \min_{b_x, b_y} J^*(f, b_x, b_y) = \min_{b_x, b_y} \min_f J^*(f, b_x, b_y) \quad (3-9)$$

² Cette propriété peut être utilisée lors de la mise au point du programme. Il suffit d'arrêter le processus à la fin de la première étape. On peut alors comparer le résultat obtenu au résultat d'une reconstruction lissée au sens de Tikhonov, obtenue par un autre programme.

Définissons maintenant:

$$T(b_x, b_y) = \min_f J^*(f, b_x, b_y) \quad (3-10)$$

Comme $J^*(f, b_x, b_y)$ est quadratique en f à b fixé, l'équation (3-10) admet une solution unique, que nous noterons f_b car elle dépend de b . D'après (3-9) et (3-10), on a:

$$\min_f J(f) = \min_{b_x, b_y} T(b_x, b_y) \quad (3-11)$$

Le minimum de $T(b_x, b_y)$ est solution des équations normales associées, soit pour tout k :

$$\frac{\partial T(b_x, b_y)}{\partial (b_x)_k} = 0 \quad \text{et} \quad \frac{\partial T(b_x, b_y)}{\partial (b_y)_k} = 0 \quad (3-12)$$

Or, on peut montrer (cf. Annexe D) les résultats suivants, valables pour les deux algorithmes.

Théorème 4:

1) L'ensemble des solutions des équations normales associées à $T(b_x, b_y)$:

$$\frac{\partial T(b_x, b_y)}{\partial (b_x)_k} = 0 \quad \text{et} \quad \frac{\partial T(b_x, b_y)}{\partial (b_y)_k} = 0$$

est l'ensemble des \hat{b} vérifiant les équations du point fixe qui s'écrivent dans le cas d'ARTUR :

$$(\hat{b}_x)_k = \frac{\varphi'[(D_x f)_k]}{2(D_x f)_k} \quad \text{et} \quad (\hat{b}_y)_k = \frac{\varphi'[(D_y f)_k]}{2(D_y f)_k}$$

et, dans le cas de LEGEND:

$$(\hat{b}_y)_k = \left\{ 1 - \frac{\varphi'[(D_x f)_k]}{2(D_x f)_k} \right\} (D_x f)_k \quad \text{et} \quad (\hat{b}_y)_k = \left\{ 1 - \frac{\varphi'[(D_y f)_k]}{2(D_y f)_k} \right\} (D_y f)_k$$

2) Si \hat{b} est un minimum de $T(b_x, b_y)$ alors l'estimée correspondante $f_{\hat{b}}$ est une solution de (3-7).

Le premier résultat nous assure que tous les minima de $T(b_x, b_y)$ (en particulier le minimum global) font partie de l'ensemble des solutions des équations du point fixe.

Le second résultat est très important. En effet, malgré l'égalité (3-11), rien n'assure que le minimum de $T(b_x, b_y)$ soit tel que l'image associée soit un minimum du MAP. Le second résultat nous permet d'affirmer que:

$$\text{si } \hat{b} = \arg \min_{b_x, b_y} T(b_x, b_y) \quad \text{alors } f_{\hat{b}} = \arg \min_f J(f) \quad (3-13)$$

Ces deux résultats nous disent que si les algorithmes ARTUR et LEGEND convergent, le minimum du MAP fait partie de l'ensemble des solutions possibles. Dans le cas où le critère du MAP est convexe, le minimum du MAP est unique et on montre (cf. paragraphe suivant) qu'ARTUR et LEGEND convergent.

3.2. Résultats de convergence

Pour les algorithmes ARTUR et LEGEND, on montre les résultats suivants.

Théorème 5:

Si la fonction de potentiel vérifie:

$$\text{i) } \lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u} = 1 \quad \text{ii) } \lim_{u \rightarrow +\infty} \frac{\varphi'(u)}{2u} = 0$$

$$\text{iii) } \frac{\varphi'(u)}{2u} \text{ est continue et strictement décroissante sur } [0, +\infty[.$$

$$\text{iv) } \lim_{u \rightarrow 0} \varphi'''(u) = 0$$

Alors:

- La suite des énergies duales $J_n = J^*(f^n, b^{n+1})$ est convergente.

- On a $(b_x^{n+1} - b_x^n) \rightarrow 0$ et $(b_y^{n+1} - b_y^n) \rightarrow 0 \quad \forall k$

$$(D_x f^n - D_x f^{n-1}) \rightarrow 0 \quad \text{et} \quad (D_y f^n - D_y f^{n-1}) \rightarrow 0 \quad \forall k$$

- Si φ est convexe sur \mathfrak{R}^+ , les suites Rf^n , $D_x f^n$, $D_y f^n$ et b^n sont convergentes. Si de plus, R est inversible, alors la suite f^n converge également.

Ces résultats sont démontrés en Annexe E, dans le cas d'ARTUR. L'hypothèse iv) est vérifiée par toutes les fonctions que nous utilisons.

Lorsque φ est convexe, le critère du MAP est convexe, et son minimum est unique. Le minimum en b de $T(b)$ l'est donc aussi. Il y a donc un seul point où la dérivée de $T(b)$ s'annule. Dans le cas où R est de rang plein, la suite des estimées f^n converge. La solution trouvée est donc la solution unique de (3-7).

3.3. Perspectives: cas des potentiels non-convexe

Dans le cas non convexe, les deux premiers résultats du théorème 5 sont toujours valables. La convergence des algorithmes n'est pas encore prouvée en théorie, mais nous l'observons expérimentalement. Dans ce cas, le point où la dérivée de $T(b)$ s'annule n'est plus unique et ne correspond pas forcément à un minimum local. En plus de la convergence, il nous faut donc prouver que la solution estimée correspond bien à un minimum local du MAP.

4. Implantation des algorithmes ARTUR et LEGEND

D'une manière générale, les algorithmes ARTUR et LEGEND sont basés sur deux procédures. La première calcule les nouvelles valeurs de la variable auxiliaire à l'aide de l'estimée calculée à l'étape précédente. La seconde calcule la nouvelle estimée à partir de l'estimée précédente et des nouvelles valeurs de la variable auxiliaire. Dans ce paragraphe, nous donnons quelques détails pratiques sur le calcul de la nouvelle estimée, la paramétrisation du modèle et les tests d'arrêts utilisés.

4.1. Résolution des équations normales

4.1.1. Introduction

Dans les deux algorithmes, la résolution des équations normales implique la résolution de systèmes linéaires de la forme $Af = c$.

Dans le cas d'ARTUR, nous avons choisi de résoudre ces équations par la méthode des sur-relaxations successives ("Successive Over Relaxation" ou SOR, en anglais). Cet algorithme bien connu est détaillé dans beaucoup d'ouvrages (voir par exemple [CIAR], [STRA], [FLET], [NUMC]).

L'expression du pixel k de l'estimée à l'étape $m+1$, f_k^{m+1} , par l'algorithme SOR est:

$$f_k^{m+1} = (1 - \omega)f_k^m + \omega \frac{c_k - T_k}{A_{k,k}} \quad (3-14)$$

Dans cette expression, on voit apparaître le terme:

$$T_k = \sum_{l=0}^{k-1} A_{k,l} f_l^{m+1} + \sum_{l=k+1}^{N^2-1} A_{k,l} f_l^m \quad (3-15)$$

qui représente le produit scalaire de la ligne k de A par l'estimée, moins le terme $A_{k,k} f_k^{m+1}$.

4.1.2. Cas de l'algorithme ARTUR

Dans le cas d'ARTUR, nous n'utilisons pas directement (3-14). En effet, on a:

$$A = R^t R - \lambda^2 \Delta_A^{n+1} \text{ et } c = R^t p \quad (3-16)$$

Nous ne formons jamais le Laplacien pondéré car c'est une matrice très creuse, et dont on connaît explicitement l'expression. Multiplier l'image par Δ_A^{n+1} revient à appliquer en chaque pixel k un filtre du type suivant (les lettres N, S, E, W signifient respectivement Nord, Sud, Est et Ouest):

$$\begin{pmatrix} 0 & \lambda_k^N & 0 \\ \lambda_k^W & -\Sigma_k & \lambda_k^E \\ 0 & \lambda_k^S & 0 \end{pmatrix} \quad (3-17)$$

dont les coefficients sont donnés par:

$\lambda_k^E = (b_x^{n+1})_{k+1}$	$\lambda_k^W = (b_x^{n+1})_k$
$\lambda_k^S = (b_y^{n+1})_{k+N}$	$\lambda_k^N = (b_y^{n+1})_k$

(3-18)

avec $\Sigma_k = \lambda_k^S + \lambda_k^N + \lambda_k^E + \lambda_k^W$. Ces pondérations sont stockées dans quatre tableaux NxN, λ^W , λ^E , λ^S et λ^N . Cela double la place mémoire nécessaire au stockage des variables intermédiaires, mais cela simplifie le programme au niveau de la gestion des indices et des problèmes de bords (cf. §4.5). On définit la *somme pondérée des voisins* par:

$$L_k = \lambda_k^S f_{k+N} + \lambda_k^N f_{k-N} + \lambda_k^E f_{k+1} + \lambda_k^W f_{k-1} \quad (3-19)$$

Le produit scalaire de la ligne k de Δ_A^{n+1} par l'estimée s'écrit alors

$$\begin{aligned}
 (\Delta_A^{n+1})_k f &= -\Sigma_k f_k + L_k \\
 &= (f_{k+N} - f_k) \lambda_k^S + (f_{k-N} - f_k) \lambda_k^N \\
 &+ (f_{k+1} - f_k) \lambda_k^E + (f_{k-1} - f_k) \lambda_k^W
 \end{aligned}
 \tag{3-20}$$

Nous utilisons cette expression pour modifier (3-14). Nous obtenons ainsi l'algorithme de calcul de f^{n+1} donné figure 3-4 (on note $f^{n+1,m+1}$ l'estimée à l'itération $m+1$ et à l'étape $n+1$. La valeur de l'indice m à convergence de l'algorithme est notée \hat{m}).

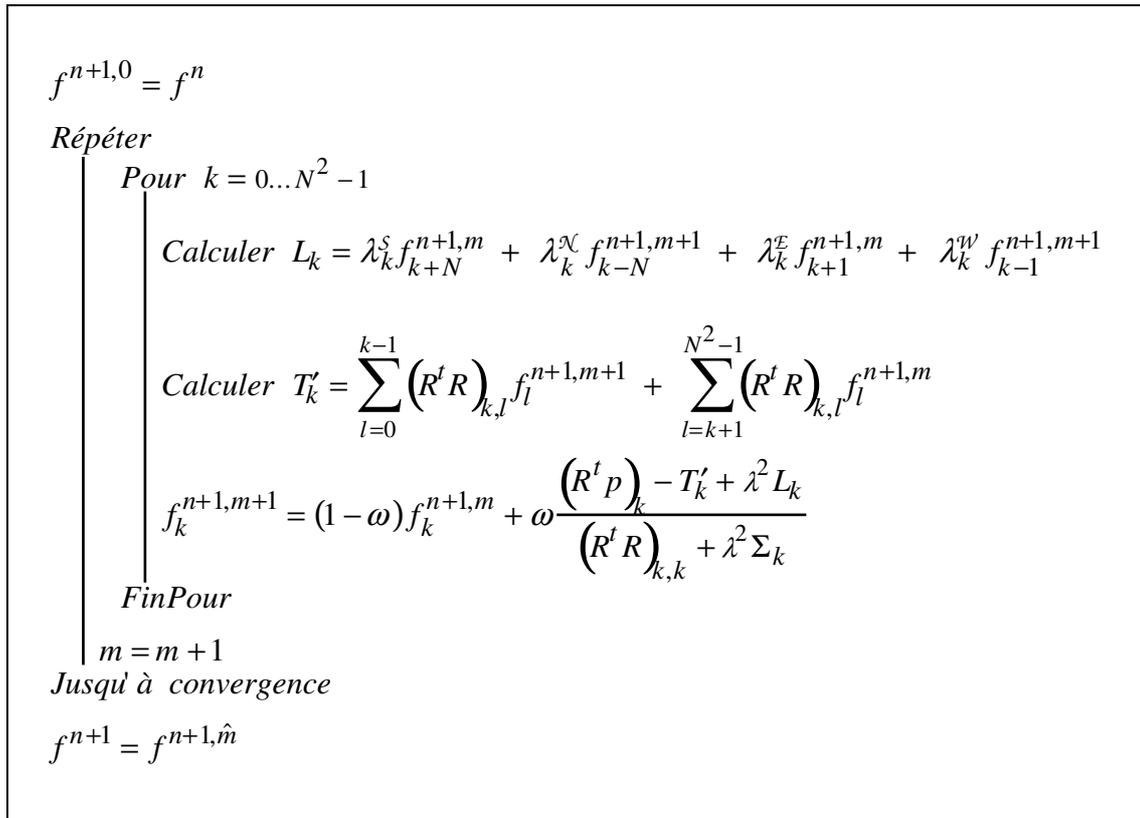


Figure 3-4: Algorithme SOR modifié pour le calcul de f^{n+1}

Nous avons utilisé une méthode de type SOR (ou Gauss-Seidel lorsque $\omega = 1$), mais on pourrait également utiliser une méthode de type Jacobi, ou un algorithme de type gradient. Dans ce dernier cas, on s'intéresse (à chaque étape) à la résolution d'un système dynamique de la forme:

$$\left. \frac{\partial J^*(f, b^{n+1})}{\partial f} \right|_{f^{n+1,m}} = \left. \frac{\partial f}{\partial t} \right|_{f^{n+1}}$$

En posant $\partial f^{n+1} = f^{n+1,m+1} - f^{n+1,m}$, cela conduit naturellement à un algorithme du gradient

$$f^{n+1,m+1} = f^{n+1,m} - \partial t \left[(R^t R - \lambda^2 \Delta_A^{n+1}) f^{n+1,m} - R^t p \right]$$

Cet algorithme est présenté sur la figure 3-5.

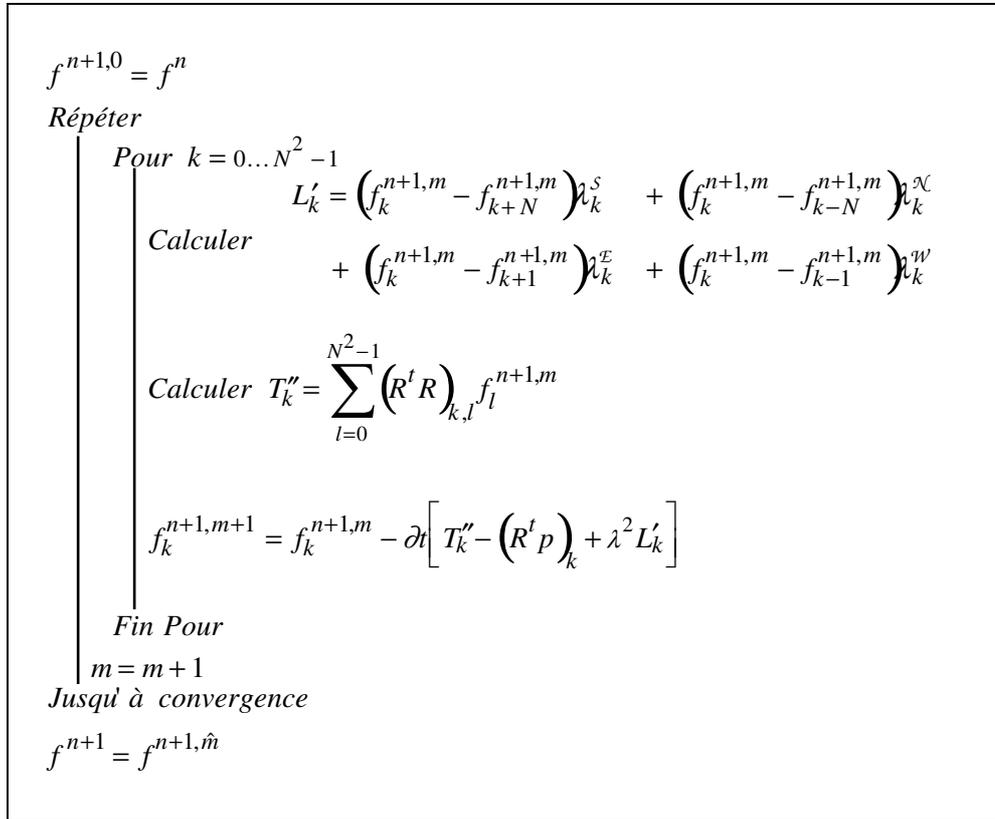


Figure 3-5: Algorithme de gradient pour le calcul de f^{n+1}

On pourrait également envisager d'utiliser un algorithme de type gradient conjugué.

4.1.3. Cas de l'algorithme LEGEND

Dans le cas de LEGEND, les équations normales à résoudre à chaque étape sont du même type que dans le cas d'une régularisation par Tikhonov. Toutes les pondérations du Laplacien sont égales à 1. On a :

$$A = R^t R - \lambda^2 \Delta \quad \text{et} \quad c = R^t p + \lambda^2 D_x^t b_x^{n+1} + \lambda^2 D_y^t b_y^{n+1} \quad (3-21)$$

On peut utiliser le SOR modifié, avec tous les poids à 1. Toutefois, ce n'est pas la meilleure façon de procéder.

En effet, on remarque que dans le cas de LEGEND, la matrice A peut être factorisée au préalable, car elle n'évolue pas entre les étapes. Comme la matrice A est symétrique et définie positive, nous utilisons une factorisation de Cholesky [CIAR] [CEA89]:

$$A = BB^t \quad (3-22)$$

(où B est une matrice triangulaire inférieure). Le système $Af = c$ s'écrit alors:

$$BB^t f = c \quad (3-23)$$

Dans ces conditions, calculer f^{n+1} revient à résoudre successivement le système triangulaire (inférieur):

$$Be = c \quad (3-24)$$

puis le système triangulaire (supérieur):

$$B^t f^{n+1} = e \quad (3-25)$$

Autrement dit, en supposant que l'on ait factorisé au préalable un nombre suffisant de matrices A (chacune correspondant à une valeur donnée de λ^2), le programme LEGEND se base sur des résolutions de systèmes triangulaires. L'opération de factorisation est très coûteuse, mais elle est effectuée *une fois pour toutes*. Chaque résolution des deux systèmes triangulaires est beaucoup plus rapide qu'une résolution itérative. Cette version est donc beaucoup plus rapide que celle basée sur l'algorithme SOR.

Une autre solution pour réduire le temps de calcul est présentée au Chapitre V. Elle consiste à résoudre l'équation $Af = c$ dans le domaine des ondelettes, où la matrice A est très creuse. Cette solution est plus pratique à mettre en œuvre dans le cas de LEGEND que dans le cas d'ARTUR, car la matrice A n'a pas à être recalculée entre chaque étape.

4.2. Paramétrisation du modèle

4.2.1. Harmonisation des fonctions de pondération

Comme nous l'avons fait remarquer au Chapitre II, paragraphe 1.5.3, les fonctions de pondération utilisées ont des vitesses de décroissance très différentes. On peut donc s'attendre à obtenir des contours moins francs avec certaines fonctions qu'avec d'autres. Afin d'éviter qu'à valeurs de paramètres fixées, les algorithmes fournissent des solutions très différentes selon la fonction utilisée, nous introduisons dans l'expression des fonctions de potentiel des constantes

d'harmonisation. Les expressions des fonctions ainsi obtenues sont données dans le tableau 3-3. Les fonctions sont tracées figure 3-6.

Potentiel	Expression de $\varphi'(u)/2u$	Valeur en 1
φ_{GM}	$\frac{1}{[1+(3u)^2]^2}$	1%
φ_{HL}	$\frac{1}{1+(7u)^2}$	2%
φ_{HS}	$\frac{1}{\sqrt{1+(30u)^2}}$	3.33%
φ_{GR}	$\begin{cases} 1 & u = 0 \\ \tanh(30u)/(30u) & u \neq 0 \end{cases}$	3.33%

Tableau 3-3: Tableau récapitulatif des fonctions de pondération utilisées en pratique

Les constantes ont été choisies de façon à ce que les fonctions de pondération soient proches de 0 pour $u=1$. Autrement dit, on considère qu'un gradient supérieur à 1 correspond à une discontinuité, et on s'arrange pour que le coefficient correspondant dans le Laplacien pondéré soit quasiment nul, ce qui interrompra le lissage.

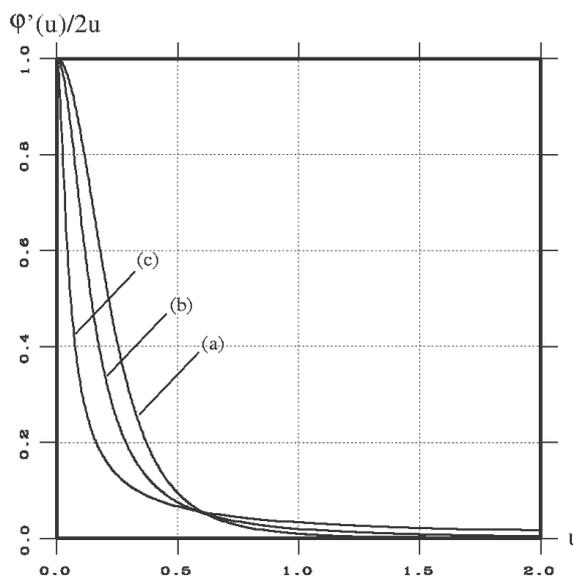


Figure 3-6: Fonctions de pondérations utilisées en pratique, correspondant à: (a) φ_{GM} , (b) φ_{HL} , (c) φ_{HS}

Les valeurs en $u=1$ sont également données dans le tableau 3-3. Ces valeurs ont pour l'instant été choisies de façon empirique. A notre connaissance, il existe très peu de publications sur l'harmonisation des fonctions utilisées. On trouve quelques propositions sur ce sujet dans [GREE90A] et [LALU92], mais ces articles s'intéressent à l'harmonisation des fonctions de potentiel et non à l'harmonisation des fonctions de pondération.

4.2.2. Paramètres des programmes

Les programmes ARTUR et LEGEND sont dédiés à la régularisation avec prise en compte des discontinuités. Le but est de lisser l'image tout en préservant les discontinuités. Il paraît donc normal que les paramètres des programmes soient:

- un coefficient de lissage qui contrôle le "degré" de lissage de la solution
- une hauteur minimum de discontinuité qui contrôle la valeur du gradient à partir de laquelle une discontinuité est introduite.

Si la notion de seuil d'introduction de discontinuité est naturelle lorsque l'on utilise un processus de ligne Booléen (cf. [BLAK87]), elle l'est moins dans le cas d'une variable auxiliaire réelle. En effet, si l'on regarde le tracé de la fonction de pondération (cf. figure 3-6), on voit qu'il n'y a pas de transition nette entre la zone correspondant au lissage et celle correspondant à la préservation des discontinuités. Toutefois, on peut considérer qu'à partir d'une certaine valeur du gradient, par exemple 1, la valeur de pondération associée est tellement faible que le lissage se trouve effectivement interrompu.

Les images que nous traitons ont des caractéristiques très différentes. En particulier, la hauteur de discontinuité minimum va dépendre de l'image étudiée. Soit δ la valeur du gradient à partir de laquelle on souhaite préserver les discontinuités. Nous considérons le critère du MAP:

$$J(f) = \|p - Rf\|^2 + \lambda^2 \sum_k \varphi[(D_x f)_k / \delta] + \lambda^2 \sum_k \varphi[(D_y f)_k / \delta] \quad (3-26)$$

Dans ce cas, un gradient de δ est ramené à 1 et le lissage se trouve interrompu à cet endroit de l'image. Le paramètre δ joue le rôle de coefficient de normalisation ou d'échelle. C'est aussi lui qui permet d'indiquer la valeur minimale du gradient à partir de laquelle une discontinuité est introduite.

Nous appliquons les résultats du Chapitre II en tenant compte du coefficient de normalisation δ et nous obtenons les expressions suivantes pour la variable auxiliaire:

$$(b_x^{n+1})_k = \frac{\varphi' \left[\frac{(D_x f^n / \delta)_k}{2} \right]}{2 (D_x f^n / \delta)_k} \quad (3-27)$$

dans le cas d'ARTUR et:

$$(b_x^{n+1})_k = \left\{ 1 - \frac{\varphi' \left[\frac{(D_x f^n / \delta)_k}{2} \right]}{2 (D_x f^n / \delta)_k} \right\} (D_x f^n / \delta)_k \quad (3-28)$$

dans le cas de LEGEND (les expressions en y sont similaires à celles en x). L'énergie duale à b fixé s'écrit à l'étape $n+1$:

$$J^*(f, b^{n+1}) = \|p - Rf\|^2 + \lambda'^2 \|D_x f\|_{B_x^{n+1}}^2 + \lambda'^2 \|D_y f\|_{B_y^{n+1}}^2 \quad (3-29)$$

dans le cas d'ARTUR et:

$$J^*(f, b^{n+1}) = \|p - Rf\|^2 + \lambda'^2 \|b_x^{n+1} - D_x f\|^2 + \lambda'^2 \|b_y^{n+1} - D_y f\|^2 \quad (3-30)$$

dans le cas de LEGEND avec $\lambda' = \lambda / \delta$.

Nous avons choisi comme paramètres des programmes:

- le coefficient de lissage, λ'
- la hauteur des discontinuités, δ

Il faut faire attention à ce que les valeurs de ces paramètres soient "raisonnables". En effet, λ' fait la balance entre le terme d'attache aux données et le terme de régularisation. S'il est choisi trop proche de 0, la régularisation peut être insuffisante et le système devenir instable. De même, une valeur trop faible de δ peut entraîner la détection de discontinuités en chaque pixel de l'image et on obtiendra une solution très bruitée. Enfin, on remarque que les deux paramètres ne sont pas totalement découplés. On peut interpréter $\lambda' = \lambda / \delta$ comme un coefficient de lissage normalisé.

Remarque: Nous prenons la même convention dans le cas de la régularisation de Tikhonov. L'énergie du MAP s'écrit alors:

$$J_{Tik}(f) = \|p - Rf\|^2 + \lambda^2 \|D_x f / \delta\|^2 + \lambda^2 \|D_y f / \delta\|^2 \quad (3-31)$$

Ce critère peut se réécrire sous la forme:

$$J_{Tik}(f) = \|p - Rf\|^2 + \lambda^2 \|D_x f\|^2 + \lambda^2 \|D_y f\|^2 \quad (3-32)$$

et le seul paramètre du programme est le coefficient de lissage normalisé λ' .

4.3. Utilisation des huit voisins

Pour ne pas surcharger les expressions développées jusqu'ici, nous n'avons utilisé que les quatre plus proches voisins de chaque pixel. En fait, dans la réalité, nous utilisons couramment les huit voisins du pixel. Nous sommes donc amenés à rajouter au critère du MAP deux termes, fonctions des gradients diagonaux.

Par définition, les gradients diagonaux sont égaux à $u = \delta f / \sqrt{2}$ où δf représente la différence entre deux pixels voisins sur la même diagonale (distants de $\sqrt{2}$, comme le montre la figure 3-7).

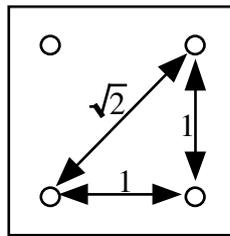


Figure 3-7: Distance entre pixels voisins

Tout ce qui a été développé ci-dessus s'applique de la même façon avec $u = \delta f / \sqrt{2}$. Les équations normales s'écrivent sous la même forme que dans le cas des quatre voisins (cf. tableau 3-2), moyennant une modification des Laplaciens qui y interviennent.

Les Laplaciens pondérés Δ_A^{n+1} et Δ_L^{n+1} s'écrivent sous la forme générale:

$$\begin{pmatrix} \lambda_k^{NW} & \lambda_k^N & \lambda_k^{NE} \\ \lambda_k^W & -\Sigma_k & \lambda_k^E \\ \lambda_k^{SW} & \lambda_k^S & \lambda_k^{SE} \end{pmatrix} \quad (3-33)$$

où Σ_k est la somme des huit pondérations. L'expression des pondérations horizontales et verticales ne change pas par rapport au cas des quatre voisins. Les pondérations diagonales sont données dans le cas d'ARTUR par:

$$\frac{1}{2} \frac{\varphi'(\delta f / \sqrt{2})}{2(\delta f / \sqrt{2})} \quad (3-34)$$

et dans le cas de LEGEND par:

$$\frac{1}{2} \left(1 - \frac{\varphi'(\delta f / \sqrt{2})}{2(\delta f / \sqrt{2})} \right) \quad (3-35)$$

Le Laplacien Δ intervenant dans les équations normales de LEGEND s'écrit:

$$\begin{pmatrix} 1/2 & 1 & 1/2 \\ 1 & -6 & 1 \\ 1/2 & 1 & 1/2 \end{pmatrix} \quad (3-36)$$

L'utilisation des huit voisins revient à rajouter deux termes au critère du MAP. De façon à avoir des résultats comparables à ceux obtenus avec quatre voisins (c'est à dire ayant le même "degré" de lissage), il faudra veiller à diviser par deux le coefficient de lissage dans le cas où on utilise huit voisins.

4.4. Tests d'arrêt

En cours d'exécution du programme, il est très coûteux de calculer la valeur du critère du MAP. Outre la charge supplémentaire de calcul que cela impose, on est obligé dans ce cas d'avoir en mémoire la matrice R en plus de la matrice $R^t R$. En conséquence, mettre un test d'arrêt sur le critère n'est pas réalisable dans le cadre d'une utilisation intensive du programme, sur une station de travail standard. C'est pourquoi nous avons utilisé comme critère d'arrêt l'évolution relative entre estimées successives. Ainsi, nous arrêtons les étapes lorsque:

$$\frac{\|f^{n+1} - f^n\|^2}{\|f^n\|^2} < F_e \quad (3-37)$$

et à l'étape $n+1$, on itère jusqu'à:

$$\frac{\|f^{n+1,m+1} - f^{n+1,m}\|^2}{\|f^{n+1,m}\|^2} < F_i \quad (3-38)$$

où F_e et F_i sont des seuils de précision fixés par l'utilisateur.

4.5. Conditions aux bords

Notre stratégie pour le traitement des bords est la suivante: nous ne fixons pas les conditions aux bords au niveau du critère du MAP, mais directement au niveau du calcul des variables intermédiaires et du calcul du Laplacien pondéré intervenant dans les équations normales.

4.5.1. Cas d'ARTUR

• Examinons d'abord le cas du calcul du Laplacien pondéré au pixel k , donné par l'équation (3-20). Pour faciliter la compréhension, nous écrivons de nouveau cette équation en utilisant l'indexation naturelle à deux dimensions: $f_k = f_{i,j}$ avec $k = i * N + j$

$$\begin{aligned} \left(\Delta_A^{n+1}\right)_k f &= (f_{i+1,j} - f_{i,j}) \lambda_{i,j}^S + (f_{i-1,j} - f_{i,j}) \lambda_{i,j}^N \\ &+ (f_{i,j+1} - f_{i,j}) \lambda_{i,j}^E + (f_{i,j-1} - f_{i,j}) \lambda_{i,j}^W \end{aligned} \quad (3-39)$$

Supposons que l'on se trouve sur un bord, par exemple sur le bord gauche de l'image, au pixel $(i,0)$. Il est clair que le calcul du terme $(f_{i,-1} - f_{i,0}) \lambda_{i,0}^W$ pose problème.

Les conditions de Neumann consistent à supposer que le gradient est nul perpendiculairement aux bords de l'image. Dans notre exemple, cela revient à poser $f_{i,-1} = f_{i,0}$. En fait, nous ne calculons jamais $\left(\Delta_A^{n+1}\right)_k f$ à l'aide de l'expression (3-39). L'itération du SOR modifié fait apparaître la somme pondérée des voisins au point considéré, L_k , et la somme des pondérations, Σ_k (cf. §4.1.2). On a: $\left(\Delta_A^{n+1}\right)_k f = L_k - \Sigma_k f_k$.

Dans notre exemple, d'après (3-39) et la définition de L_k et Σ_k , les conditions de Neumann s'expriment par:

$$\begin{cases} L_{i,0} = (f_{i+1,0}) \lambda_{i,0}^S + (f_{i-1,0}) \lambda_{i,0}^N + (f_{i,1}) \lambda_{i,0}^E + (f_{i,0}) \lambda_{i,0}^W \\ \Sigma_{i,0} = \lambda_{i,0}^S + \lambda_{i,0}^N + \lambda_{i,0}^E + \lambda_{i,0}^W \end{cases} \quad (3-40)$$

Nous agissons de la même façon sur les autres bords de l'image.

• Il nous reste à résoudre les problèmes de bords au niveau du calcul des pondérations. Dans notre exemple, en appliquant les conditions de Neumann, soit $f_{i,-1} = f_{i,0}$, on a $\lambda_{i,0}^W = 1$. Nous agissons de la même façon sur tous les bords de l'image. Nous mettons à 1:

- la première colonne du tableau λ^W ,
- la dernière colonne de λ^E ,
- la première ligne de λ^N ,
- la dernière ligne de λ^S ,

au moment du calcul de ces pondérations, c'est à dire au début de chaque étape.

• Notons que dans ce cas, (3-40) devient

$$\begin{cases} L_{i,0} = (f_{i+1,0})\lambda_{i,0}^S + (f_{i-1,0})\lambda_{i,0}^N + (f_{i,1})\lambda_{i,0}^E + (f_{i,0}) \\ \Sigma_{i,0} = \lambda_{i,0}^S + \lambda_{i,0}^N + \lambda_{i,0}^E + 1 \end{cases} \quad (3-41)$$

Remarque: Dans le cas de la reconstruction tomographique, l'objet n'occupe qu'une partie de l'image, entourée de valeurs nulles. On peut donc considérer que l'on a $f_{i,0} = 0$. (3-39) devient alors:

$$\left(\Delta_A^{n+1}\right)_k f = (f_{i+1,0})\lambda_{i,0}^S + (f_{i-1,0})\lambda_{i,0}^N + (f_{i,1})\lambda_{i,0}^E \quad (3-42)$$

Comme on ne se sert pas de $\lambda_{i,0}^W$, on peut lui donner une valeur quelconque, par exemple 0. C'est en fait ce que nous faisons en pratique dans le cas de la tomographie.

4.5.2. Cas de LEGEND

Nous appliquons les mêmes principes dans le cas de l'algorithme LEGEND. Il n'y a pas de changement en ce qui concerne les termes L_k et Σ_k intervenant dans l'expression de $\left(\Delta_L^{n+1}\right)_k f$ (cf. remarque du § 2.2.2). Dans ce cas toutefois, les pondérations du Laplacien sont définies de façon différente de celles d'ARTUR, cf. expression (3-5). D'après cette expression, on voit que les conditions de Neumann appliquées au calcul des pondérations obligent à mettre à 0 les bords des tableaux des pondérations au lieu de 1 dans le cas d'ARTUR.

5. Comparaisons avec d'autres algorithmes

Dans ce paragraphe, nous comparons de façon théorique ARTUR (et LEGEND, qui fonctionne de la même façon) avec d'autres algorithmes existants. Cette comparaison n'a pas pour but d'évaluer expérimentalement les algorithmes, mais simplement de mettre en évidence quelques points communs et certaines différences. Le but est d'une part de situer ARTUR par rapport aux algorithmes déterministes les plus connus, et d'autre part de dégager d'éventuelles perspectives de travail.

5.1. ARTUR et le GNC

5.1.1. Description rapide de l'algorithme GNC

L'algorithme à non convexité graduelle (plus connu par son abréviation anglaise GNC) a été proposé par Blake et Zisserman [BLAK87]. Le modèle considéré utilise un processus de ligne Booléen. Toutefois, ce processus de ligne n'est pas utilisé explicitement dans l'algorithme. Au

contraire, il est “éliminé” et n’apparaît plus qu’implicitement à travers la fonction de potentiel $\varphi_{BZ} = \min(u^2, 1)$.

Le principe de l’algorithme est de minimiser une suite de critères $J^{(p)}$, tels que $J^{(1)}$ soit convexe et que $J^{(p)}$ tende vers l’énergie du MAP lorsque $p \rightarrow 0$. Ainsi, on introduit graduellement la non convexité au niveau du critère minimisé.

Pour chaque valeur de p , on minimise $J^{(p)}$ par une méthode de descente simple, en partant de l’estimée précédente. En faisant varier p “assez lentement”, on peut espérer trouver le minimum global du critère du MAP (mais cela n’est pas prouvé dans le cas général). En pratique, on fait varier p de façon dyadique: $p = 1, \frac{1}{2}, \frac{1}{4}, \text{etc...}$ On remarque que comme le premier critère est convexe, la solution calculée par l’algorithme ne dépend pas de l’estimée initiale [BLAK87].

La suite de critères $J^{(p)}$ est construite en approximant φ_{BZ} par:

$$\varphi_{BZ}^{(p)}(u) = \begin{cases} u^2 & u < q \\ 1 - \frac{c}{2}(t-r)^2 & q \leq u < r \\ 1 & u \geq r \end{cases} \quad (3-43)$$

avec $r = \frac{2}{c} + \frac{1}{2}$, $q = \frac{1}{r}$ et $c = \frac{c^*}{p}$ (voir figure 3-8).

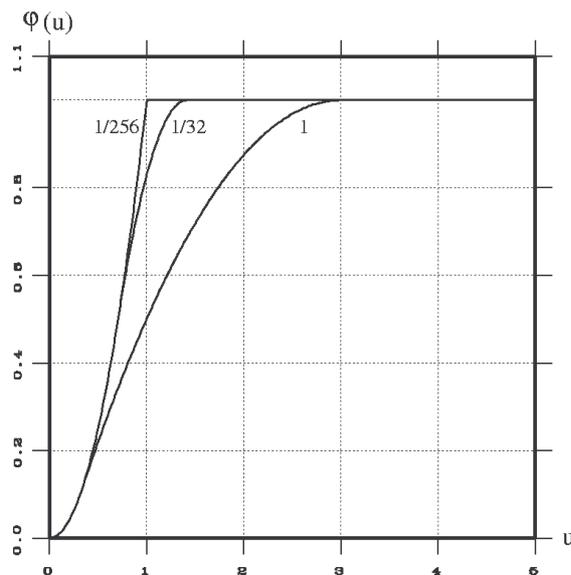


Figure 3-8: Fonction $\varphi_{BZ}^{(p)}$ pour plusieurs valeurs de p ($c^* = 1/4$).

On définit alors:

$$J^{(p)}(f) = \|p - Rf\|^2 + \lambda^2 \sum_k \varphi_{BZ}^{(p)} \left[(D_x f)_k \right] + \lambda^2 \sum_k \varphi_{BZ}^{(p)} \left[(D_y f)_k \right] \quad (3-44)$$

L'usage du GNC dans le cas où la matrice R est mal conditionnée peut poser problème. En effet, la constante c^* doit être choisie de façon à ce que $J^{(1)}$ soit convexe. Précisément, on doit avoir $c^* = \alpha_{\min}/4$ où α_{\min} est la plus petite valeur propre de l'opérateur auto-adjoint $R^t R$. Si la matrice R est mal conditionnée, α_{\min} est nulle ou quasiment nulle et donc la valeur de c^* est très faible, voire nulle. Il n'est donc pas toujours possible d'avoir $J^{(1)}$ convexe. Une solution (cf. [NIKO94]) consiste à ajouter au critère un terme quadratique pondéré par un coefficient a . Dans ce cas, on effectue d'abord plusieurs étapes en gardant P à 1 et en faisant varier le paramètre a , d'une valeur a_0 assurant la convexité du critère, vers 0. Ensuite, on commence à faire varier P .

5.1.2. ARTUR et le GNC

Les algorithmes que nous avons proposés ici et le GNC ont un point commun: ce sont des algorithmes implicites. Chaque étape consiste à minimiser un certain critère. Dans les deux cas, on définit donc une suite de critères. La construction de ces deux suites de critères est cependant très différente.

Tout d'abord, l'algorithme GNC, sous sa forme originale, s'applique au cas de la fonction de potentiel quadratique tronquée. ARTUR et LEGEND sont plus généraux dans la mesure où ils permettent l'emploi d'un grand nombre de fonctions de potentiel. Cependant, la quadratique tronquée ne satisfait pas les conditions de validité rappelées au paragraphe 3 et son emploi dans nos algorithmes requiert une mise en œuvre particulière.

D'autre part, la suite définie dans le GNC est basée sur une variation de la fonction de potentiel à l'aide du paramètre P . Dans notre cas, la fonction de potentiel reste la même au fil des étapes. La variation du critère repose sur la variation des variables auxiliaires en fonction de l'estimée précédente.

Enfin, et ce n'est pas la moindre des différences, les critères définis dans le GNC ne sont pas convexes (à part le premier), alors que dans le cas d'ARTUR et LEGEND, ils le sont. Dans le cas du GNC, on applique à chaque étape une méthode de descente simple sur un critère non quadratique et non convexe. Dans le cas d'ARTUR et de LEGEND, on minimise à chaque étape un critère quadratique, ce qui est plus simple et plus sûr.

5.2. ARTUR et le MFA

5.2.1. Description rapide de l'algorithme MFA

L'algorithme de recuit par champ moyen ("Mean Field Annealing" ou MFA en anglais) a été défini dans le cas où le critère à minimiser met en jeu un processus de ligne Booléen (3-45). Nous donnons ici une description simplifiée de l'algorithme, dans nos notations. En particulier, notre processus de ligne est défini de façon complémentaire à celui utilisé dans l'article original, et nous ne paramétrons pas le modèle. Le lecteur pourra se référer à [GEIG90], [GEIG91] pour une présentation plus complète dans le formalisme original.

Dans le cas du processus de ligne Booléen, on doit minimiser l'énergie suivante:

$$J_B^*(f, b) = \|p - Rf\|^2 + \sum_k \varphi_B^*[(D_x f)_k, (b_x)_k] + \sum_k \varphi_B^*[(D_y f)_k, (b_y)_k] \quad (3-45)$$

avec

$$\varphi_B^*(u, b) = bu^2 + (1 - b) \quad (3-46)$$

De façon équivalente, on peut dire qu'il faut maximiser la probabilité *a posteriori* associée. En fait, on utilise la notion de recuit en introduisant un paramètre $\beta = 1/T$, relié à la notion de température. Pour chaque valeur de ce paramètre, on maximise:

$$P_\beta(f, b|p) = \frac{1}{Z} e^{-\beta J_B^*(f, b)} \quad (3-47)$$

Le paramètre β évolue au cours des itérations de façon à faire baisser la température. A température nulle, les seules solutions de probabilité non nulle sont celles qui minimisent le critère du MAP. Dans l'expression (3-47), Z est la fonction de partition, définie comme la somme de la probabilité *a posteriori* sur toutes les configurations possibles de (f, b) :

$$Z = \sum_{\{f, b\}} e^{-\beta J_B^*(f, b)} \quad (3-48)$$

La fonction de partition joue un rôle clef dans la méthode. En effet, à partir de Z , il est possible d'obtenir l'expression littérale de la moyenne de chacune des variables. Par exemple, dans le cas de la variable b , on sait que:

$$\bar{b}_k = \frac{1}{Z} \sum_{\{f, b\}} b_k e^{-\beta J_B^*(f, b)} \quad (3-49)$$

Cette expression peut se mettre sous la forme:

$$(\bar{b}_x)_k = \frac{-1}{\beta} \frac{\partial \ln(Z)}{\partial (D_x f)_k^2} \quad (3-50)$$

On a la même expression dans la direction y. Étant donné le rôle central que joue Z dans la méthode, il est nécessaire d'en avoir une expression la plus simple possible.

Dans un premier temps, un raisonnement simple montre [GEIG91] que l'expression de Z peut se mettre sous la forme suivante:

$$Z = \sum_{\{f\}} e^{-\beta J_\beta(f)} \quad (3-51)$$

où

$$J_\beta(f) = \|p - Rf\|^2 + \sum_k \varphi_\beta[(D_x f)_k] + \sum_k \varphi_\beta[(D_y f)_k] \quad (3-52)$$

et

$$\varphi_\beta(u) = 1 - \ln\left(1 + e^{-\beta(u^2-1)}\right) \quad (3-53)$$

La fonction φ_β est tracée figure 3-9 pour plusieurs valeurs de β . On peut constater que lorsque β est très faible (soit pour une température élevée), la fonction φ_β se comporte comme une fonction convexe sur une large bande [CHAM93]. Par contre, lorsque la température diminue, la fonction se rapproche de la quadratique tronquée.

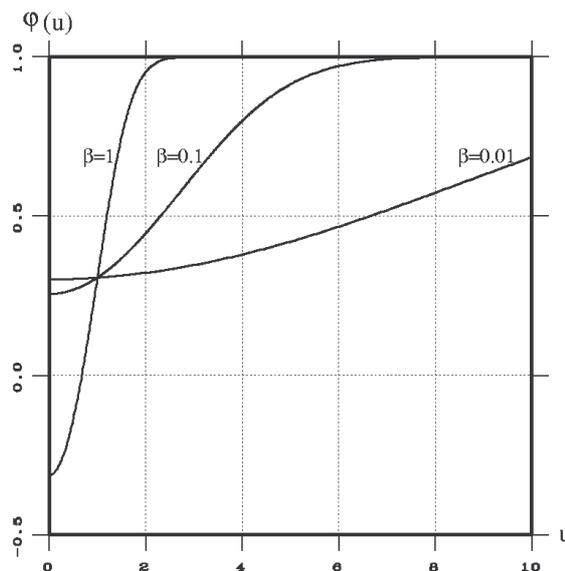


Figure 3-9: Fonction φ_β pour plusieurs valeurs de β

Le calcul de la fonction de partition étant trop compliqué, une autre idée-clé de la méthode est d'approximer Z par sa valeur maximale. Cette approximation, dite "du point-selle", revient à figer f à sa valeur moyenne \bar{f} . On a alors

$$Z \propto e^{-\beta J_\beta(\bar{f})} \quad (3-54)$$

et \bar{f} vérifie:

$$\frac{\partial J_\beta(\bar{f})}{\partial f_k} = 0 \quad \forall k \quad (3-55)$$

Ce qui équivaut à:

$$(R^t R - \bar{\Delta})\bar{f} = R^t p \quad (3-56)$$

Le terme $\bar{\Delta}$ est un Laplacien pondéré, comme dans les algorithmes décrits précédemment. Les pondérations sont les valeurs du processus de ligne moyen $\bar{b} = (\bar{b}_x, \bar{b}_y)$. Ces valeurs sont calculées à l'aide de (3-50), qui donne dans l'approximation (3-54):

$$(\bar{b}_x)_k = 1 - \frac{1}{1 + e^{\beta(1 - (D_x \bar{f})_k^2)}} \quad \text{et} \quad (\bar{b}_y)_k = 1 - \frac{1}{1 + e^{\beta(1 - (D_y \bar{f})_k^2)}} \quad (3-57)$$

Remarquons que la variable auxiliaire utilisée dans l'algorithme est à valeurs continues, sauf lorsque la température tend vers 0. Dans ce cas, la variable auxiliaire tend à devenir Booléenne.

Pour résoudre (3-56), on introduit un schéma dynamique du type:

$$\frac{\partial \bar{f}}{\partial t} = \frac{\partial J_\beta(\bar{f})}{\partial \bar{f}_k} \quad (3-58)$$

Ce qui nous conduit à un algorithme explicite du type gradient. A chaque itération, on commence par calculer les valeurs de $(\bar{b}_x^{n+1}, \bar{b}_y^{n+1})$ à l'aide des expressions (3-57) appliquées à \bar{f}^n . Cela nous permet de définir $\bar{\Delta}^{n+1}$. L'expression de l'estimée à l'itération n+1 est alors:

$$\bar{f}^{n+1} = \bar{f}^n - \partial t \left[(R^t R - \bar{\Delta}^{n+1})\bar{f}^n - R^t p \right] \quad (3-59)$$

Une version utilisant un algorithme de type gradient conjugué a été proposée dans [ZERU91]. Une extension au cas du modèle anisotrope a été effectuée dans [ZERU93].

Lorsque l'algorithme de gradient a convergé, on fait décroître la température, on calcule une nouvelle estimée du champ moyen, et ainsi de suite. A température nulle, le champ moyen calculé correspond à une estimée au sens du MAP. A notre connaissance, il n'y a pas de preuve de convergence vers un minimum global dans le cas du MFA.

5.2.2. ARTUR et le MFA

Les algorithmes ARTUR et MFA sont, à première vue très différents. En premier lieu, on peut noter qu'ils ne s'adressent pas aux mêmes fonctions de potentiel. En effet, même s'il utilise une variable auxiliaire continue, le MFA est défini dans le cas particulier où le critère à minimiser (3-45) met en jeu un processus de ligne Booléen. Cela correspond donc au cas de la fonction de potentiel "quadratique tronquée" φ_{BZ} . Cette fonction ne satisfait pas aux conditions de validité d'ARTUR et LEGEND (cf. paragraphe 3 du présent chapitre). On ne peut donc pas appliquer directement nos algorithmes dans ce cas. Par contre, ces conditions de validité sont remplies par une large gamme de fonctions de potentiels, correspondant à des processus de ligne continus.

De plus, la suite de critères minimisés dans le MFA est construite à partir de la variation de la fonction de potentiel. Cette variation s'opère ici via le paramètre β , lié à la température du système. L'algorithme ARTUR n'utilise pas la notion de recuit. La fonction de potentiel reste la même au fil du déroulement de l'algorithme. La suite des critères varie directement en fonction des estimées.

Une autre différence entre les deux algorithmes est qu'ARTUR suit un schéma implicite alors que pour chaque valeur de température, le MFA suit un schéma explicite. Autrement dit, à chaque étape, ARTUR doit résoudre une équation pour calculer la nouvelle estimée. Par contre, dans le cas du MFA à température fixée, la nouvelle estimée à chaque étape est donnée directement par l'expression (3-59).

Toutefois, hormis cette différence algorithmique, on note des similitudes frappantes. D'une part, l'expression (3-56) est semblable aux équations normales à b fixé données au paragraphe 2.2.2. Comme dans le cas d'ARTUR, la résolution de ces équations normales non linéaires est facilitée par l'utilisation de la variable auxiliaire. D'autre part, l'expression de la variable auxiliaire à \bar{f} fixée (3-57) est en fait la même que celle donnée au paragraphe 2.3. En effet, sous l'approximation du point-selle, on a:

$$\frac{-1}{\beta} \ln(Z) = J_{\beta}(\bar{f}) \quad (3-60)$$

Un calcul simple montre qu'alors, à \bar{f} fixé:

$$(\bar{b}_x)_k = \frac{\varphi'_{\beta} \left[(D_x \bar{f})_k \right]}{2(D_x \bar{f})_k} \text{ et } (\bar{b}_y)_k = \frac{\varphi'_{\beta} \left[(D_y \bar{f})_k \right]}{2(D_y \bar{f})_k} \quad (3-61)$$

Autrement dit, les équations (3-56) et (3-57), qui sont à la base du MFA, cadrent parfaitement avec ce que nous avons développé précédemment. Même si la façon dont elles sont obtenues est différente, on a les mêmes expressions, avec $\varphi = \varphi_{\beta}$, $b = \bar{b}$ et $f = \bar{f}$.

En conclusion, on peut dire qu'à température fixée, le MFA apparaît comme un analogue d'ARTUR (cf. §7), dans le cas particulier où $\varphi = \varphi_\beta$, et où l'on définit un schéma explicite.

5.3. Potentiels variables et utilisation de variable auxiliaire

On peut noter que le MFA fonctionne selon un principe proche de celui du GNC. Il s'agit ici aussi de minimiser une suite de critères, J_β , tendant vers le critère du MAP lorsque la température tend vers 0. En effet, pour β grand, la fonction φ_β tend vers la quadratique tronquée. Minimiser J_β revient alors à minimiser l'énergie définie dans (3-45), qui correspond à l'énergie $J^{(0)}$ du GNC [GEIG91]. La définition de la suite de critères diffère cependant selon l'algorithme.

L'algorithme GNC et l'algorithme MFA utilisent donc une idée intéressante: on fait varier la fonction de potentiel au cours de l'algorithme, en partant d'une fonction convexe et en tendant vers la fonction non-convexe correspondant au critère du MAP. Dans le GNC, cette variation s'opère à l'aide d'un paramètre P défini arbitrairement. Dans le cas du MFA, on utilise la notion de recuit et on fait varier la fonction de potentiel φ_β à l'aide de β .

Comme nous l'avons remarqué au paragraphe précédent, le MFA à température fixée peut être vu comme un algorithme explicite basé sur une régularisation semi-quadratique, dans le cas particulier du potentiel φ_β . Le MFA met donc en œuvre à la fois la notion de variable auxiliaire et la notion de variation du potentiel. On pourrait imaginer d'introduire la notion de variation de potentiel dans d'autres algorithmes basés sur la régularisation semi-quadratique. En utilisant un schéma explicite, trois algorithmes de ce type ont été proposés par J.M. Bruneau dans [BRUN93] sous le nom "d'algorithmes paramétriques". On pourrait également songer à appliquer cette technique dans le cas de l'algorithme ARTUR.

Algorithme	Avec variable auxiliaire	Sans variable auxiliaire
Avec φ variable	MFA, algorithmes "paramétriques" [BRUN93]	GNC
Avec φ fixée	MFA à température constante, ARTUR, LEGEND	Algorithmes de descente simples, ICM sans recuit

Tableau 3-4: Classification des principaux algorithmes déterministes dans le cas de la vraisemblance Gaussienne.

Nous avons rassemblé les principaux algorithmes déterministes dans le tableau 3-4, en les classant selon qu'ils utilisent ou non une variation du potentiel et/ou une variable auxiliaire.

La technique de variation du critère utilisée dans le MFA et le GNC est conçue pour répondre au problème de la non convexité du critère du MAP. La technique utilisant la variable auxiliaire est plutôt une réponse à l'aspect non quadratique du critère.

Dans le cas de critères convexes, nous avons vu (cf. §3) qu'ARTUR et LEGEND calculent le minimum global du critère. Toutefois, il n'est pas évident *a priori* de justifier leur utilisation dans le cas de critères non convexes. En effet, nos algorithmes ne s'attaquent pas directement au problème de la non convexité du critère, mais plutôt à son aspect non quadratique. Cependant, le principe reste le même:

“calculer d'abord une estimée en [minimisant] un critère [convexe]³, et modifier petit à petit cette estimée en changeant doucement les contraintes”.

Ce principe est très général. Pour preuve: la phrase que nous citons est empruntée à la description d'un algorithme déterministe, différent de ceux que nous avons cités jusqu'ici, le “Deterministic Pseudo Annealing”(DPA) [BERT92] [ZERU92] [KATO93]. La différence entre les algorithmes réside dans la façon dont on fait évoluer les contraintes. Dans notre cas, le critère reste convexe à chaque étape. Son évolution en fonction de b se fait de façon à introduire progressivement des discontinuités dans l'estimée.

Dans la pratique, on peut constater que même dans le cas de critères non convexes, ARTUR et LEGEND donnent de très bons résultats (cf. Chapitre IV). Dans le même registre d'idées, et même si cette affirmation est à nuancer dans le cas d'images très peu contrastées et très bruitées [ZERU91], Geiger et Girosi font remarquer [GEIG91] que les résultats obtenus par le MFA à température nulle sont souvent aussi bons que ceux obtenus en utilisant le recuit. En nous rappelant que le MFA à φ fixée est un analogue implicite d'ARTUR, cela nous conforte à penser que, même si ce n'est pas son but initial, l'utilisation de la variable auxiliaire apporte une réponse satisfaisante au problème de l'aspect non convexe du critère du MAP.

On peut donc se demander s'il serait vraiment judicieux d'associer variation du critère et utilisation de la variable auxiliaire dans un algorithme “ARTUR avec φ variable”, étant donné la charge de calcul supplémentaire que cela entraînerait.

³ Dans le cas du “Deterministic Pseudo Annealing” [BERT92], le premier critère est concave.

5.4. ARTUR et le MAP-EM OSL de Green

5.4.1. Algorithme ML-EM

L'algorithme itératif ayant longtemps fait référence en reconstruction tomographique est l'algorithme ML-EM (pour Maximum Likelihood Expectation Maximization algorithm) [DEMP77] [SHEPP82] [LANG84]. Cet algorithme n'utilise pas la régularisation, toutefois, nous donnons sa description détaillée car il est à la base des algorithmes MAP-EM déterministes.

En tomographie, les algorithmes EM s'utilisent dans le cas d'une modélisation Poissonienne de la formation des projections (cf. Chapitre I). Dans ce cas, le terme de vraisemblance s'écrit:

$$J_1(f) = -\log[P(p|f)] = \sum_{i=0}^{N^2-1} \{p_i \ln(\langle R_i, f \rangle) - \langle R_i, f \rangle\} + Cte \quad (3-62)$$

où R_i représente la i -ème ligne de R . La plupart du temps, on introduit à ce niveau une variable auxiliaire non observée $z_{t,s}$, les "données complètes", représentant le nombre de photons émis par le pixel s de l'image et captés par l'élément t du détecteur. On écrit alors:

$$p_t = \sum_s z_{t,s} \quad (3-63)$$

L'algorithme consiste en deux phases (qui lui donnent son nom): l'estimation des $z_{t,s}$ et la maximisation de $P(z|f)$. La première phase revient à calculer [GREE90A]:

$$z_{t,s} = E(z_{t,s} | p_t, \hat{f}^n) = \frac{p_t R_{t,s} \hat{f}_s^n}{\langle R_t, \hat{f}^n \rangle} \quad (3-64)$$

La seconde phase est équivalent à la minimisation de $J_1'(f) = -\log[P(z|f)]$; on résout les équations normales associées à l'énergie $J_1'(f)$ faisant intervenir les données complètes:

$$\hat{f}_s^{n+1} = \frac{\sum_t z_{t,s}}{\sum_t R_{t,s}} \quad (3-65)$$

En général, on combine les deux expressions en:

$$\hat{f}_s^{n+1} = \frac{\hat{f}_s^n}{\sum_t R_{t,s}} \sum_t \frac{p_t R_{t,s}}{\langle R_t, \hat{f}^n \rangle} \quad (3-66)$$

On peut réécrire (3-66) sous forme additive:

$$\hat{f}_s^{n+1} = \hat{f}_s^n + \frac{\hat{f}_s^n}{\sum_t R_{t,s}} \left[\sum_t \frac{p_t R_{t,s}}{\langle R_t, \hat{f}^n \rangle} - R_{t,s} \right] \quad (3-67)$$

Lorsque $\sum_t R_{t,s} = 1$, cet algorithme est aussi connu sous le nom d'algorithme de Richardson-Lucy. Nous notons que l'on aurait pu obtenir directement cette expression en construisant un schéma itératif explicite de la façon suivante:

$$\hat{f}^{n+1} = \hat{f}^n + \nabla J_1 \cdot \hat{f}^n \quad (3-68)$$

avec
$$\nabla J_1 = \text{diag} \left(\frac{1}{\sum_t R_{t,s}} \frac{\partial J_1(\hat{f})}{\partial f_s} \right)$$

Notons également que cet algorithme, de type gradient, peut être accéléré en introduisant un coefficient de sur-relaxation [LEWI86].

5.4.2. Algorithme MAP-EM-OSL

Dans le cas où on utilise la régularisation dans le cadre du MAP (ex. [HURW75] [HANS83] [LIAN88] [GREE90A]), on a à résoudre une équation du type:

$$J(f) = \sum_{i=0}^{N^2-1} \{ p_i \ln \langle R_i, f \rangle - \langle R_i, f \rangle \} + \lambda^2 J_2(f) \quad (3-69)$$

avec:
$$J_2(f) = \sum_k \varphi[(D_x f)_k] + \sum_k \varphi[(D_y f)_k]$$

On peut comme dans le cas du ML-EM, introduire une variable auxiliaire représentant les données complètes. On arrive à une expression ressemblant à (3-66):

$$\hat{f}_s^{n+1} = \frac{\hat{f}_s^n}{\sum_t R_{t,s} + \lambda^2 \frac{\partial J_2(f)}{\partial f_s}} \sum_t \frac{p_t R_{t,s}}{\langle R_t, \hat{f}^n \rangle} \quad (3-70)$$

Dans cette expression, le terme $\partial J_2(f) / \partial f_s$ est non linéaire et son calcul pose problème [LANG90] [LALU92]. B. Chalmond a proposé un algorithme stochastique pour résoudre le problème du MAP-EM (cf. [CHAL89]). Dans une approche déterministe, J. Green propose (cf.

[GREE90A] et [GREE90B]), de calculer le terme problématique à l'aide de l'estimée précédente, \hat{f}^n , (d'où le nom d'algorithme "One-Step-Late" ou OSL). L'expression de l'itération (3-70) devient:

$$\hat{f}_s^{n+1} = \frac{\hat{f}_s^n}{\sum_t R_{t,s} + \lambda^2 \left. \frac{\partial J_2(f)}{\partial f_s} \right|_{\hat{f}^n}} \sum_t \frac{P_t R_{t,s}}{\langle R_t, \hat{f}^n \rangle} \quad (3-71)$$

Dans ce cas également, on peut remettre cette expression sous forme additive [LANG90]. Une version accélérée a été proposée dans [NOUM93] dans le cas de la fonction de potentiel quadratique.

5.4.3. ARTUR et le MAP-EM-OSL

Les algorithmes MAP-EM-OSL et ARTUR diffèrent nettement de par le modèle de vraisemblance utilisé. Dans le cas d'ARTUR, on fait une hypothèse Gaussienne, alors que dans le cas du MAP-EM-OSL, on considère une statistique de Poisson. L'utilisation d'une vraisemblance Gaussienne en tomographie est sujette à caution. En effet, cette hypothèse n'est en principe valable qu'à fort taux de comptage. Nous avons donc essayé de comparer expérimentalement les deux algorithmes dans [KOUL94] en faisant varier le taux de comptage dans les projections. Il apparaît que pour les taux de comptage habituels en reconstruction clinique, ARTUR donne d'aussi bons résultats que le MAP-EM-OSL pour un temps de calcul largement inférieur.

Toutefois, il nous a également paru intéressant de comparer ces algorithmes d'un point de vue théorique. Il est possible de réécrire l'algorithme MAP-EM-OSL en faisant apparaître la variable auxiliaire b . Le MAP-EM-OSL prend alors la forme suivante:

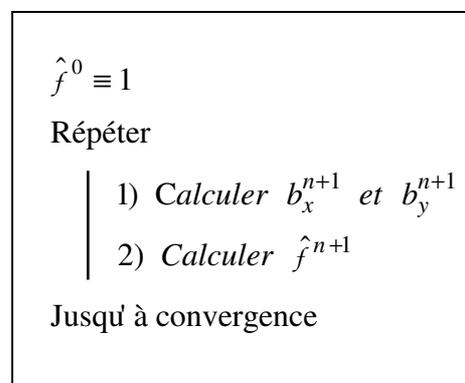


Figure 3-10: Algorithme MAP-EM-OSL

Les valeurs des variables auxiliaires sont calculées en utilisant leurs expressions explicites (cf. tableau 3-1, paragraphe 2.2.1). En reprenant les conventions du paragraphe 4.1.2, on peut écrire:

$$\left. \frac{\partial J_2(f)}{\partial f_s} \right|_{\hat{f}^n} = - \left(\Delta_A^{n+1} \right)_s \hat{f}^n = \sum_s \hat{f}_s^n - L'_s \quad (3-72)$$

Dans cette expression, L'_s est la somme des voisins du pixel s , pondérée par les valeurs de la variable auxiliaire: $L'_s = \lambda_s^S \hat{f}_{s+N}^n + \lambda_s^N \hat{f}_{s-N}^n + \lambda_s^E \hat{f}_{s+1}^n + \lambda_s^W \hat{f}_{s-1}^n$ et \sum_s est la somme des pondérations. Le calcul de \hat{f}^{n+1} se fait en appliquant à chaque pixel s l'expression:

$$\hat{f}_s^{n+1} = \frac{\hat{f}_s^n}{\sum_t R_{t,s} + \lambda^2 \left(\sum_s \hat{f}_s^n - L_s \right)} \sum_t \frac{p_t R_{t,s}}{\langle R_t, \hat{f}^n \rangle} \quad (3-73)$$

Nous remarquons qu'à la différence d'ARTUR, ce schéma est *explicite*, dans la mesure où à chaque étape on a directement l'expression de \hat{f}^{n+1} en chaque pixel.

Les deux différences entre ARTUR et le MAP-EM-OSL sont donc d'une part le modèle de vraisemblance utilisé, d'autre part le type de schéma mis en œuvre.

5.4.4. Perspective: ARTUR dans le cas de la vraisemblance Poissonienne

En perspective, nous proposons une version d'ARTUR, dans le cas de la vraisemblance Poissonienne. On peut également voir cet algorithme comme une version implicite du MAP-EM-OSL. Cet algorithme se déroule par étapes, comme ARTUR. A chaque étape, on emploie un algorithme itératif de type MAP-EM pour calculer \hat{f}^{n+1} .

Plus précisément, à l'étape $n+1$, et à l'itération $m+1$, on obtient en suivant le raisonnement habituel:

$$\hat{f}_s^{n+1,m+1} = \frac{\hat{f}_s^{n+1,m}}{\sum_t R_{t,s} + \lambda^2 \left. \frac{\partial J_2(f)}{\partial f_s} \right|_{\hat{f}^{n+1}}} \sum_t \frac{p_t R_{t,s}}{\langle R_t, \hat{f}^{n+1,m} \rangle} \quad (3-74)$$

A la différence de l'algorithme OSL, nous n'évaluons pas la dérivée de l'énergie a priori à l'aide de l'estimée à l'étape précédente, mais nous posons:

$$\left. \frac{\partial J_2(f)}{\partial f_s} \right|_{\hat{f}^{n+1}} = - \left(\Delta_A^{n+1} \right)_s \hat{f}^{n+1} = \sum_s \hat{f}_s^{n+1,m+1} - L_s \quad (3-75)$$

où la somme pondérée des pixels voisins est définie, comme dans ARTUR, par:

$$L_s = \lambda_s^S \hat{f}_{s+N}^{n+1,m} + \lambda_s^N \hat{f}_{s-N}^{n+1,m+1} + \lambda_s^E \hat{f}_{s+1}^{n+1,m} + \lambda_s^W \hat{f}_{s-1}^{n+1,m+1} \quad (3-76)$$

Pour simplifier, nous noterons:

$$S_s = \sum_t R_{t,s} \quad \text{et} \quad A_s = \hat{f}_s^{n+1,m} \sum_t \frac{p_t R_{t,s}}{\langle R_t, \hat{f}_s^{n+1,m} \rangle} \quad (3-77)$$

L'équation (3-74) se transforme alors en équation du second degré:

$$\lambda^2 \Sigma_s (\hat{f}_s^{n+1,m+1})^2 + (S_s - \lambda^2 L_s) \hat{f}_s^{n+1,m+1} - A_s = 0 \quad (3-78)$$

Il faut donc résoudre cette équation en chaque site s . A chaque fois, on retient la solution strictement positive, afin d'assurer la positivité de l'estimée. Le discriminant de (3-78) est donné par: $d_s = (\lambda^2 L_s - S_s)^2 + 4\lambda^2 \Sigma_s A_s$. Comme $\hat{f}_s^{n+1,m}$ est toujours strictement positive le discriminant l'est aussi. On obtient ainsi l'algorithme suivant pour le calcul de l'estimée à l'étape $n+1$.

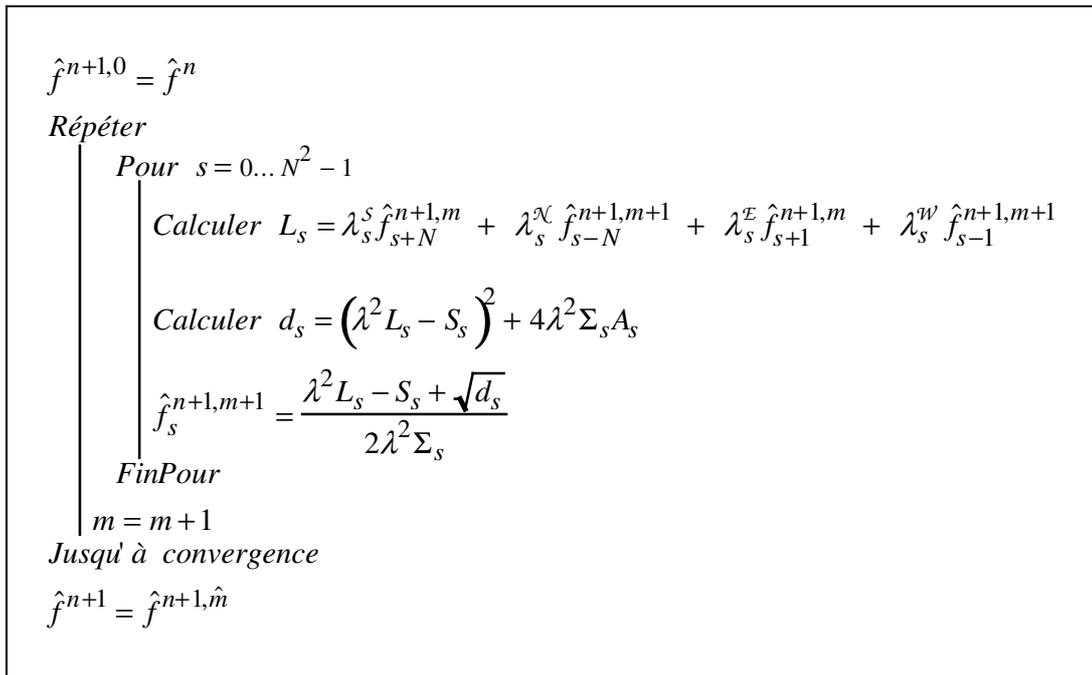


Figure 3-11: Algorithme MAP-EM utilisé à chaque étape pour le calcul de \hat{f}^{n+1}

A la première étape, toutes les pondérations sont égales à 1. Dans ce cas, on vérifie facilement que l'on retrouve l'algorithme proposé par Levitan et Hermann [LEVI87] dans le cas d'un *a priori* Gaussien (à la seule différence que dans [LEVI87], l'*a priori* porte sur la norme de l'image et non sur la norme du gradient.).

Cet algorithme, appelé MOISE (pour Modified One-step-late Implicit Scheme Em), a été mis en œuvre par P.M. Koulibaly, et est actuellement en cours de tests. Les premiers résultats semblent prometteurs (voir Chapitre IV). Il reste également à étudier ses propriétés théoriques de convergence.

6. Perspectives: autres stratégies possibles d'optimisation

6.1. Introduction

Nous avons vu dans le chapitre II comment passer du problème de la minimisation d'un critère du MAP non quadratique à un problème semi-quadratique, en introduisant une variable auxiliaire b . Cette introduction peut se faire de manière multiplicative (théorème 2) ou additive (théorème 3). Le problème à résoudre consiste à minimiser la fonctionnelle d'énergie duale $J^*(f, b)$. Nous avons décidé de procéder par étapes, en calculant successivement la nouvelle valeur de b , puis la nouvelle valeur de f . Depuis le début de ce chapitre, nous avons étudié une façon particulière de faire, ce qui nous a conduit aux algorithmes ARTUR et LEGEND. Cette méthode n'est pas unique et l'on peut imaginer d'autres algorithmes. Dans ce paragraphe, nous proposons de passer en revue les principaux types d'algorithmes qui nous paraissent possibles. Pour cela, nous proposons une petite classification des algorithmes.

Nous distinguons d'abord deux catégories d'algorithmes de reconstruction. En premier lieu, les algorithmes basés sur des schémas stationnaires résolvent les équations normales données par: $\partial J^* / \partial f = 0$. En second lieu, ceux utilisant des schémas dynamiques cherchent la solution d'une équation du type: $\partial J^* / \partial f = \partial f / \partial t$. Dans ce paragraphe, nous passons en revue les différents algorithmes qui peuvent être obtenus, selon la façon dont on évalue $\partial J^*(\bullet, b^{n+1})$.

Parmi les algorithmes obtenus, nous en distinguerons deux sortes: les algorithmes implicites et les algorithmes explicites. Un algorithme est dit implicite si à chaque étape, il est nécessaire de résoudre une équation pour trouver la nouvelle estimée. Au contraire, si la nouvelle estimée est donnée directement, on parlera d'algorithme explicite.

6.2. Schémas stationnaires

6.2.1. Schéma stationnaire implicite

Les équations normales d'ARTUR et LEGEND (cf. tableau 3-2 §2.2.2) ont été obtenues en appliquant l'expression:

$$\left. \frac{\partial J^*(f, b^{n+1})}{\partial f} \right|_{f^{n+1}} = 0 \quad (3-79)$$

A chaque étape, les équations normales sont résolues par un algorithme itératif. Nous classons donc les algorithmes ARTUR et LEGEND dans la catégorie des algorithmes à schéma stationnaire implicite.

6.2.2. Schéma stationnaire explicite

Nous plaçons dans la catégorie des schémas stationnaires explicites les algorithmes de type “One-Step-Late” (OSL) de Green. Ce type d’algorithme est surtout connu dans le cas de la vraisemblance Poissonienne (cf. §5.). Toutefois dans [GREE90B], Green propose un exemple dans le cas Gaussien. Cet exemple est donné dans le cas d’un *a priori* très simple du type Tikhonov avec l’opérateur identité. Il n’a, à notre connaissance, pas été appliqué pour de la reconstruction avec prise en compte des discontinuités. Nous n’avons pas testé cet algorithme, et ne le proposons donc qu’à titre de perspective.

Le principe des algorithmes OSL est d’évaluer la dérivée du terme de régularisation du critère à l’aide de l’estimée précédente, f^n . Cette dérivée s’écrit alors $\Delta_A^{n+1} f^n$ (dans le cas du découplage multiplicatif).

L’algorithme proposé dans [GREE90B] est un algorithme de type Jacobi modifié. Dans l’algorithme de Jacobi, on sépare la matrice à inverser en deux de façon à isoler la partie diagonale. Plus précisément, si on doit inverser le système $Sx = t$, on pose $S = \mathcal{D} - (\mathcal{E} + \mathcal{F})$ où \mathcal{D} est une matrice diagonale. L’expression de la solution à l’étape $n+1$ est alors donnée par:

$$\mathcal{D}x^{n+1} = (\mathcal{E} + \mathcal{F})x^n + t. \quad (3-80)$$

Nous appliquons ce principe avec $S = R^t R$ et $t = R^t p + \lambda^2 \Delta_A^{n+1} f^n$. L’algorithme proposé est alors de la forme:

$$\mathcal{D} f^{n+1} = (\mathcal{E} + \mathcal{F} + \lambda^2 \Delta_A^{n+1}) f^n - R^t p \quad (3-81)$$

Dans le cas du découplage additif, on obtient le même algorithme. La dérivée du critère, ∂J^* , est évaluée en partie à l’aide de f^n et en partie à l’aide de f^{n+1} .

L’algorithme (3-81), obtenu par un schéma stationnaire est de type explicite car à chaque étape, on obtient directement la nouvelle estimée en fonction de la précédente.

6.3. Schémas dynamiques

6.3.1. Schéma dynamique implicite

Considérons maintenant le cas où l'on évalue $\partial J^*(\bullet, b^{n+1})$ avec f^{n+1} :

$$\left. \frac{\partial J^*(f, b^{n+1})}{\partial f} \right|_{f^{n+1}} = \frac{\partial f}{\partial t} \quad (3-82)$$

Cette équation nous conduit dans le cas du découplage multiplicatif - cf. équation (3-1) - à :

$$\left[I + \partial t (R^t R - \lambda^2 \Delta_A^{n+1}) \right] f^{n+1} = f^n + \partial t R^t p \quad (3-83)$$

Cette expression définit un schéma implicite. Dans le cas du découplage additif - cf. équation (3-2) - on obtient un algorithme similaire :

$$\left[I + \partial t (R^t R - \lambda^2 \Delta) \right] f^{n+1} = (I - \partial t \lambda^2 \Delta_L^{n+1}) f^n + \partial t R^t p \quad (3-84)$$

De tels algorithmes sont actuellement étudiés par Luminita Lazaroaia dans le cadre de sa thèse [LAZA95], sous la direction de Gilles Aubert.

6.3.2. Schéma dynamique explicite

Pour définir un schéma dynamique explicite, on utilise

$$\left. \frac{\partial J^*(f, b^{n+1})}{\partial f} \right|_{f^n} = \frac{\partial f}{\partial t} \quad (3-85)$$

Dans le cas du découplage multiplicatif, on obtient :

$$f^{n+1} = f^n - \partial t \left[(R^t R - \lambda^2 \Delta_A^{n+1}) f^n - R^t p \right] \quad (3-86)$$

Dans le cas du découplage additif on retrouve la même expression, en remplaçant Δ_A^{n+1} par Δ_L^{n+1} .

Cet algorithme du type gradient peut être vu comme l'analogue explicite d'ARTUR. Il est déjà utilisé pour certaines fonctions de potentiel, par exemple dans le MFA [GEIG90] à chaque

température. Il pourrait être intéressant de faire une étude de convergence de ce type d'algorithme, dans le cas général où la fonction de potentiel vérifie les conditions de préservation des discontinuités, de la même façon que pour ARTUR et LEGEND.

6.4. Conclusion

Dans ce paragraphe, nous avons présenté quatre schémas possibles pour la minimisation de $J^*(f, b)$. Ces algorithmes diffèrent dans la façon dont on calcule f^{n+1} à partir de f^n et b^{n+1} (cf. tableau 3-5).

Type de schéma	Calcul de f^{n+1}
Stationnaire, implicite	$\left(R^t R - \lambda^2 \Delta_A^{n+1} \right) f^{n+1} = R^t p$ $\left(R^t R - \lambda^2 \Delta \right) f^{n+1} = R^t p + \lambda^2 D_x^t b_x^{n+1} + \lambda^2 D_y^t b_y^{n+1}$ <p style="text-align: center;">ARTUR, LEGEND</p>
Stationnaire, explicite	$\mathcal{D} f^{n+1} = \left(\mathcal{E} + \mathcal{F} + \lambda^2 \Delta_A^{n+1} \right) f^n - R^t p$ $R^t R = \mathcal{D} - (\mathcal{E} + \mathcal{F})$ <p style="text-align: center;">MAP-OSL Gaussien</p>
Dynamique, implicite	$\left[I + \partial_t \left(R^t R - \lambda^2 \Delta_A^{n+1} \right) \right] f^{n+1} = f^n + \partial_t R^t p$ $\left[I + \partial_t \left(R^t R - \lambda^2 \Delta \right) \right] f^{n+1} = \left(I - \partial_t \lambda^2 \Delta_L^{n+1} \right) f^n + \partial_t R^t p$ <p style="text-align: center;">[LAZA95]</p>
Dynamique, explicite	$f^{n+1} = f^n - \partial_t \left[\left(R^t R - \lambda^2 \Delta_A^{n+1} \right) f^n - R^t p \right]$ <p style="text-align: center;">MFA à température fixe</p>

Tableau 3-5: Exemples d'algorithmes déterministes pour la régularisation semi-quadratique (avec vraisemblance Gaussienne).

Nous avons obtenus deux types d'algorithmes (voir figure 3-12). Les algorithmes implicites fonctionnent de la façon suivante: à chaque étape, la nouvelle valeur de b définit un nouveau critère, que l'on minimise la plupart du temps par un algorithme itératif.

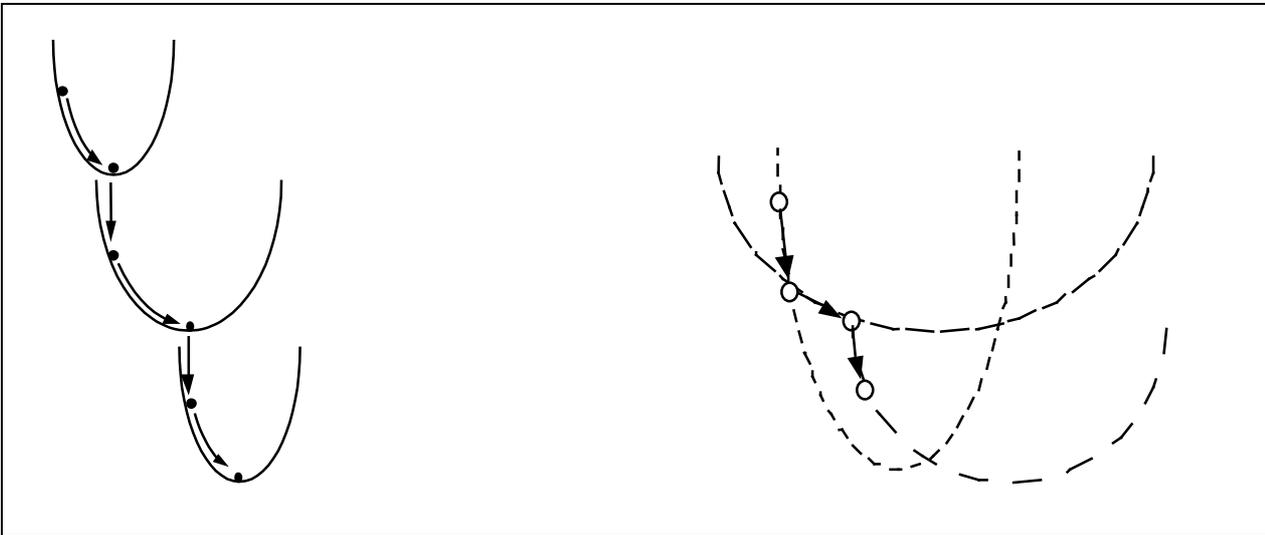


Figure 3-12: A gauche: trois étapes d'un schéma implicite: à chaque étape, à b fixé, on minimise un critère convexe. A droite: cas d'un schéma explicite: à chaque étape, à b fixé, on fait un pas de descente.

Les algorithmes implicites demandent plus de calculs à chaque étape, mais sont réputés plus stables [CIARL] que les algorithmes de la seconde catégorie, dits algorithmes explicites. Dans ce dernier cas, à chaque étape, la nouvelle valeur de b définit également un nouveau critère. Toutefois, on ne minimise pas ce critère, mais on fait simplement un pas de descente le long de celui-ci. Autrement dit, une étape consiste en une itération d'un algorithme de type gradient utilisant une détermination particulière du pas de descente. La direction de descente entre deux itérations est fonction de la nouvelle valeur de b , et varie donc de façon auto-adaptative.

7. Conclusion

Dans ce paragraphe, nous avons proposé deux algorithmes de reconstruction d'image utilisant la régularisation semi-quadratique:

- ARTUR dans le cas où la variable auxiliaire intervient de façon multiplicative dans l'énergie duale
- LEGEND dans le cas où la variable auxiliaire intervient de façon additive.

Ces deux algorithmes sont basés sur le même principe: le principe des minimisations alternées. De plus, il mettent à profit le fait qu'on a une expression littérale pour le calcul de la variable auxiliaire à chaque étape.

La mise en œuvre des algorithmes est relativement simple. En effet, ARTUR et LEGEND sont basés sur des résolutions de systèmes linéaires. On a donc remplacé le problème non-linéaire de départ, correspondant à la minimisation du critère du MAP, par une suite de problèmes linéaires.

Dans le cas où le critère est convexe, on montre qu'ARTUR et LEGEND fournissent la solution (unique) du problème de départ. Dans le cas non convexe, la solution estimée n'est pas encore caractérisée d'un point de vue théorique. Toutefois, nous verrons au Chapitre IV que les résultats expérimentaux obtenus sont de très bonne qualité.

De la comparaison avec le MFA et le GNC, il ressort que le principe de la plupart des algorithmes déterministes utilisés pour la minimisation de critères non convexe est le suivant: calculer d'abord une estimée en minimisant un critère convexe, et modifier petit à petit cette estimée en changeant doucement les contraintes. La différence entre les algorithmes réside dans la façon dont évoluent les contraintes. Dans notre cas, le critère minimisé à chaque étape est toujours convexe à b fixé, mais son évolution en fonction de b permet l'introduction progressive des discontinuités dans la solution.

Les algorithmes que nous avons présentés semblent très bien adaptés au problème de la reconstruction d'image avec prise en compte des discontinuités. Toutefois, il reste beaucoup d'autres possibilités. Les comparaisons et les essais de classification que nous avons proposés en fin de chapitre nous ouvrent des perspectives de travail intéressantes. En particulier, il serait intéressant d'effectuer une étude complète des différents algorithmes possibles, sur le plan théorique comme sur le plan expérimental. Une telle étude pourrait permettre de choisir le meilleur compromis selon divers critères, tels que les propriétés de convergence, la charge de calcul, l'adaptation au problème posé, la facilité d'implantation, etc...

Chapitre IV

Application à la reconstruction tomographique

1. Introduction

Au Chapitre III, nous avons présenté deux algorithmes déterministes de reconstruction d'image basés sur une régularisation semi-quadratique. Dans le présent chapitre, nous nous proposons d'appliquer ces algorithmes à la reconstruction tomographique. La tomographie est une des techniques d'imagerie nucléaire, qui ont pour but de fournir aux médecins des informations sur certains organes d'un patient, sans l'opérer. Nous nous intéressons plus particulièrement ici à la tomographie par émission de photon unique (en anglais "Single Photon Emission Computerized Tomography", ou SPECT) dans le cas bidimensionnel. Le problème de reconstruction consiste alors à calculer, à partir de données acquises autour du patient, une série d'images représentant des coupes de l'organe étudié.

Dans la première partie de ce chapitre, nous présentons rapidement la modélisation mathématique du problème et son aspect mal-posé (le lecteur pourra se référer à des ouvrages comme [NATT] ou [JAIN] pour plus de détails). Ensuite, nous présentons les résultats obtenus avec nos algorithmes, tant avec des données réelles qu'avec des données simulées.

2. La tomographie par émission de photon unique (SPECT)

2.1. Modélisation du problème

Lors d'un examen SPECT, un radiotracer est injecté au patient. Cet élément radioactif va être fixé par les cellules de l'organisme et émettre des photons Gamma dans toutes les directions. Cette

émission sera plus ou moins importante selon l'activité des cellules. Reconstruire une carte de la radioactivité permet donc de recueillir des renseignements *fonctionnels* sur l'organe étudié¹.



Figure 4-1: Tomographie par émission: acquisition des données

Pour cela, on utilise une caméra sensible aux photons Gamma (appelée Gamma-Caméra). Cette caméra est en rotation autour du patient, comme le montre la figure 4-1. Pour différents angles de vues, on mesure le rayonnement émis. C'est à partir de ce signal que l'on doit reconstruire des coupes de la distribution d'activité du traceur.

2.1.1. Modélisation analytique de la formation des projections

La Gamma-Caméra est équipée d'un dispositif de collimation qui permet de ne prendre en compte que les photons arrivant perpendiculairement au détecteur. Plaçons-nous dans une direction de projection θ donnée. Le signal mesuré à l'abscisse u du détecteur est proportionnel au nombre total de photons émis le long d'une ligne perpendiculaire au détecteur et passant par u .

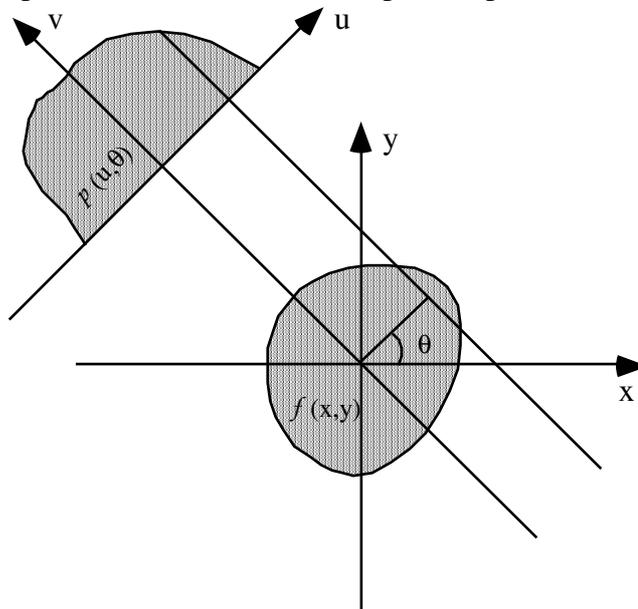


Figure 4-2: Géométrie de l'acquisition des projections (d'après [JAIN])

Notons $f(x, y)$ la distribution d'activité, et $p(u, \theta)$ le signal mesuré sur le détecteur à la position u pour

¹ à la différence d'examens tels que le scanner ou l'Imagerie par Résonance Magnétique, qui donnent des renseignements *anatomiques* sur l'organe considéré.

une angle de vue θ . D'après la remarque précédente, $p(u, \theta)$ est obtenu par intégration le long d'une ligne v perpendiculaire à u :

$$p(u, \theta) \equiv \mathcal{R}[f] = \int f(u, v) dv \quad (4-1)$$

Cette égalité est donnée dans le repère du détecteur, faisant un angle θ avec le repère de l'objet. Dans le repère de l'objet, on a:

$$p(u, \theta) \equiv \mathcal{R}[f] = \iint f(x, y) \delta(u, x \cos \theta + y \sin \theta) dx dy \quad (4-2)$$

Le signal mesuré sur le détecteur constitue l'ensemble des projections intégrales de l'objet sur le détecteur. L'opérateur de projection \mathcal{R} est appelée transformée de Radon. Elle permet de passer du domaine de l'objet au domaine des projections appelé domaine de Radon.

Un calcul simple montre que l'image d'un point du domaine réel est une sinusoïde dans le domaine de Radon. Un objet sera donc représenté dans l'espace de Radon par une superposition de sinusoïdes. C'est pourquoi on donne souvent le nom de sinogramme à l'image des projections.

Cette modélisation analytique suppose que l'on dispose d'un nombre infini de projections, ce qui n'est pas le cas en pratique.

2.1.2. Modélisation discrète de la formation des projections

Une autre approche consiste à considérer le problème discret. On divise le champ de la Gamma-caméra en une grille régulière $N \times N$. En général, on prend $N=64$ ou 128 . La taille du champ de la Caméra étant de l'ordre de 40 cm, on obtient donc une résolution spatiale de l'ordre de quelque millimètres (environ 6 mm pour $N=64$ et 3 mm pour $N=128$). La distribution du traceur dans la coupe, $f(x, y)$, est discrétisée en $f_{i,j}$ sur cette grille de pixel (cf. figure 4-3). Comme il n'y a (en principe) pas d'émission en dehors de l'objet étudié, f désignera par extension toute la grille $N \times N$, qui constituera l'image à reconstruire à partir des projections mesurées.

L'examen SPECT fournit un nombre M de projections. On discrétise chaque projection sur N pixels. Pour simplifier, on considère en général $M=N$ projections de N pixels chacune.

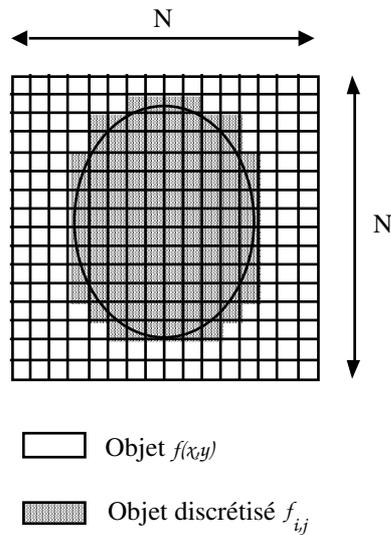


Figure 4-3: Discretisation de la distribution d'activité du traceur (N=16)

Dans l'approche discrète, les éléments du détecteur (ou "dexels") ne sont donc plus considérés ponctuels, comme dans le cas continu, mais ont une certaine largeur. De ce fait, l'intégration selon des lignes apparaissant dans les équations (4-1) et (4-2) est remplacée par une sommation sur des *raies*, comme le montre la figure 4-4.

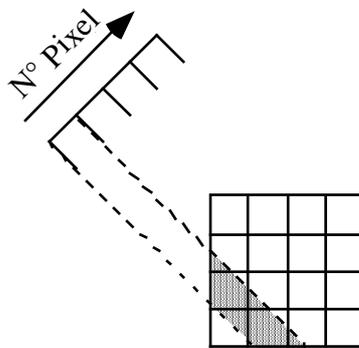


Figure 4-4: Projection d'une image 4x4 pour un angle de $\pi/4$. En grisé: raie de sommation du premier pixel

En considérant l'image de départ et le sinogramme arrangés lexicographiquement sous forme de vecteurs, l'opération de projection peut se modéliser par le système linéaire suivant:

$$p = Rf \tag{4-3}$$

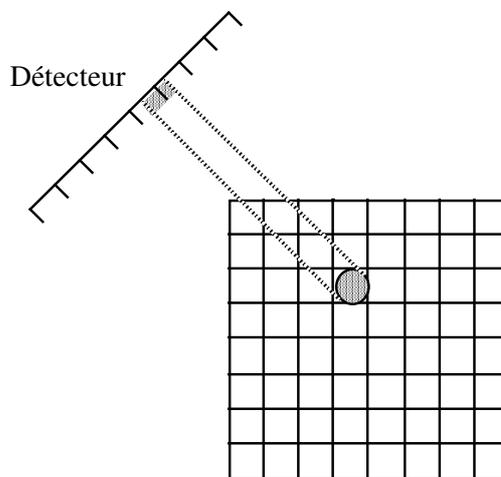
où R est une matrice de taille $N^2 \times N^2$. Nous appellerons R *matrice de Radon* car elle représente l'opération de projection du domaine réel dans l'espace de Radon.

Cette modélisation discrète permet de prendre en compte, à travers R , les perturbations physiques intervenant lors de l'acquisition des données. Ces phénomènes sont principalement l'atténuation, la perte de résolution en profondeur et la diffusion Compton [DARC92]. Toutefois, dans ce qui suit, nous considérons un modèle simple de matrice, correspondant à la projection idéale dans l'espace de Radon, sans perturbation.

2.1.3. Modèle de matrice de Radon utilisée

Le coefficient $R_{i,j}$ de la matrice de Radon correspond à la contribution du pixel f_j à la valeur p_i du vecteur des projections. C'est un réel compris entre 0 et 1 qui mesure la proportion du pixel qui participe au pixel de projection. Ce coefficient est déterminé géométriquement, ce qui nécessite de calculer la projection de chaque pixel f_j sur le détecteur [LAUR92].

Intuitivement, il est clair que le meilleur modèle de pixel est le modèle du pixel carré. Toutefois, la projection d'un pixel carré uniforme est un trapèze [KOUL91] et varie selon la position du pixel et l'angle de projection.



 Profil du pixel

Figure 4-5: Modèle du disque concave

Nous simplifions les calculs en approximant le pixel par un disque concave (cf. figure 4-5), dont la projection est un carré quelle que soit la position du pixel et l'angle de projection. On fait l'hypothèse que la projection du pixel a une amplitude de 1.

Chaque ligne i de la matrice de Radon correspond à un dixel dans une direction donnée (c.à.d. une valeur de θ). Pour calculer les coefficients de la ligne i de R , on parcourt l'image en calculant la projection de chaque pixel j sur le détecteur (voir figure 4-5).

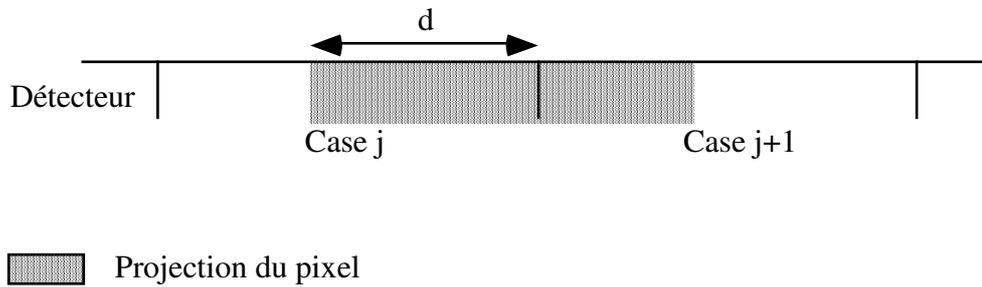


Figure 4-6: Projection d'un pixel sur le détecteur

Le coefficient $R_{i,j}$ correspond à la longueur d de l'intersection entre la projection du pixel i et la case j du détecteur (voir figure 4-6).

2.1.4. Quantification de la matrice de Radon

Le gros problème inhérent au modèle discret de l'équation (4-3) est la taille de la matrice de Radon. En effet, si p et f sont de dimension $N \times N$, la matrice R est de taille $N^2 \times N^2$. Dans le cadre d'une implantation informatique, R est très encombrante, que ce soit en ce qui concerne le stockage sur disque, ou le chargement en mémoire lors de l'exécution d'un programme. Le tableau 4-1 donne la taille de stockage de R , si chaque élément est stocké en réel sur 4 octets (en format flottant), en fonction de N .

N	16	32	64	128
Taille de R	256 Ko	4 Mo	64 Mo	1 Go

Tableau 4-1: Taille-mémoire de R en fonction de N (stockage en flottant sur 4 octets)

Le stockage et l'utilisation de matrices de grande taille devient rapidement difficile. C'est pourquoi nous proposons de quantifier ces matrices de façon à en limiter la taille-mémoire. Nous utilisons une quantification scalaire uniforme. Chaque pixel (à valeur sur $[0,1]$) est quantifié sur 16 bits (entre 0 et 65535) et stocké en entier non signé.

Un des avantages de la quantification scalaire uniforme est que la déquantification (nécessaire au cours des traitements utilisant R) est très rapide, puisqu'il s'agit d'une simple remise à l'échelle des données. D'autre part, la quantification sur 16 bits permet de diviser par 2 la taille-mémoire de R , tout en gardant une précision correcte. Pour visualiser cela, nous quantifions une matrice de Radon sur 16 bits et la déquantifions.

N	16	24	32	48
Erreur sur R	4,80e-11	5,12e-11	5,25e-11	5,43e-11
Erreur sur Rf	8,46e-12	8,55e-12	4,37e-12	3,70e-12
Erreur sur $R^t p$	2,58e-13	9,90e-14	6,30e-14	5,16e-14

Tableau 4-2: Etude de la quantification de R

Nous mesurons l'erreur relative (NMSE) entre la matrice ainsi obtenue et la matrice originale. D'autre part, comme R est souvent utilisée pour effectuer des projections (Rf) ou des rétroprojections ($R^t p$), nous mesurons également l'effet de la quantification de R sur le résultat de ces opérations. Pour ces expériences, nous utilisons le fantôme numérique "Nice" présenté au paragraphe 3.1. Les résultats sont donnés dans le tableau 4-2.

La matrice $R^t R$ intervient souvent dans les algorithmes de reconstruction. Nous proposons également de la quantifier. Cette fois-ci, la valeur du pixel original n'est plus comprise entre 0 et 1 mais entre 0 et max (la valeur de max dépend de N). Nous présentons dans le tableau 4-3 la valeur de max choisie pour la quantification, en fonction de N (pour plus de sécurité, la valeur de max est arrondie à l'entier supérieur).

N	16	24	32
Valeur de max	13	19	26
Erreur sur $R^t R$	1,58e-9	3,06e-9	5,46e-9
Erreur Tikhonov	9,34e-10	1,73e-9	2,81e-9
Paramètre de lissage utilisé	5	7,5	10

Tableau 4-3: Etude de la quantification de $R^t R$

Nous donnons également les mesures d'erreur relatives sur $R^t R$ et sur une reconstruction du type Tikhonov. Pour cette dernière expérience, on utilise le fantôme numérique "Nice". Les projections sont calculées à l'aide de l'équation 4-3 (sans bruit). Le paramètre de lissage utilisé dépend de la taille de l'image.

Comme on peut le constater, on obtient encore une précision satisfaisante. Dans ce qui suit, nous utiliserons donc les matrices quantifiées sur 16 bits. Une alternative à la quantification scalaire uniforme, permettant de diminuer encore plus la taille-mémoire de R , sera présentée en perspective au Chapitre V.

2.2. Reconstruction tomographique

2.2.1. Aspect mal-posé du problème de reconstruction

Si l'on réarrange une ligne de la matrice de Radon sous forme d'image $N \times N$, on visualise les pixels de l'image participant à la valeur détectée par un dixel dans une position donnée. En d'autres termes, on visualise la raie de projection associée à un dixel. Ces images sont appelées "noyau de détection des photons" par Smith et al dans [SMIT92]¹. Nous montrons sur la figure 4-7 les raies de projection associées au dixel 5 pour quatre angles différents, dans le cas où $N=16$. Les pixels de l'image participant complètement à la valeur détectée par le dixel apparaissent en noir, ceux y participant partiellement en gris et ceux n'y participant pas, en blanc.

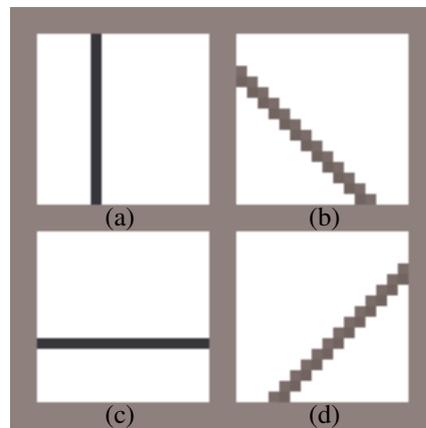


Figure 4-7: Noyaux de projection du dixel 5, pour les angles: (a) 0, (b) $\pi/4$, (c) $\pi/2$, (d) $3\pi/4$. Les pixels participant à la raie de projection n°5 sont en foncé

Il est aisé de constater qu'en superposant tous les noyaux de projection correspondant à l'angle de projection 0, on obtiendra une image uniforme de 1. Le même résultat est valable pour l'angle de projection $\pi/2$. En conséquence, une ligne de la matrice R correspondant à un angle nul peut être

¹Nous citons cet article car il est très pédagogique. Il permet de bien comprendre l'aspect mal-posé de la reconstruction tomographique grâce à une étude spectrale de la matrice de Radon et de son opérateur auto-adjoint. L'utilisation de l'inverse généralisé pour la reconstruction permet d'appréhender l'aspect spectral de la régularisation.

déduite par combinaison linéaire des autres lignes correspondant à un angle nul et des lignes correspondant à l'angle $\pi/2$. Si on note R_i^θ , $i = 0..N-1$ les lignes de la matrice R correspondant à la projection selon l'angle θ , on a:

$$R_i^0 = \sum_{j=0}^{N^2-1} R_j^{\pi/2} - \sum_{j=0, j \neq i}^{N^2-1} R_j^0 \quad (4-4)$$

Et le résultat réciproque est également vrai:

$$R_i^{\pi/2} = \sum_{j=0}^{N^2-1} R_j^0 - \sum_{j=0, j \neq i}^{N^2-1} R_j^{\pi/2} \quad (4-5)$$

Autrement dit, lorsque l'on recueille des projections pour les angles 0 et $\pi/2$, il existe un vecteur lié aux autres, ce qui fait perdre 1 rang à la matrice R [BATH93A].

En fait les équations (4-4) et (4-5) peuvent se généraliser à tous les couples d'angles séparés de $\pi/2$. Pour chaque couple, la matrice perd 1 rang. Lorsque le nombre N de projections entre 0 et π est pair, il existe $N/2$ couples d'angles séparés de $\pi/2$. La matrice de Radon est donc de rang $N^2 - N/2$. Par contre, si N est impair, on ne recueille pas de projections séparées de $\pi/2$ et la matrice de Radon est de rang plein.

La figure 4-8 (page suivante) montre le spectre des valeurs singulières de la matrice de Radon dans le cas où $N=16$. Les 8 dernières valeurs singulières sont quasiment nulles. Toutefois, ces valeurs ne sont pas *exactement* nulles. Cela est dû aux problèmes de précision machine: en pratique, R peut donc être considérée de rang plein dans tous les cas [BATH93A].

Le conditionnement d'une matrice, que nous noterons c , est défini dans le cas de la norme Euclidienne, comme le quotient de sa plus grande valeur singulière sur sa plus faible valeur singulière. Le conditionnement de R mesuré pour différentes valeurs de N est présenté figure 4-9. On voit nettement que dans le cas où N est pair, le conditionnement mesuré est beaucoup plus grand que lorsque N est impair.

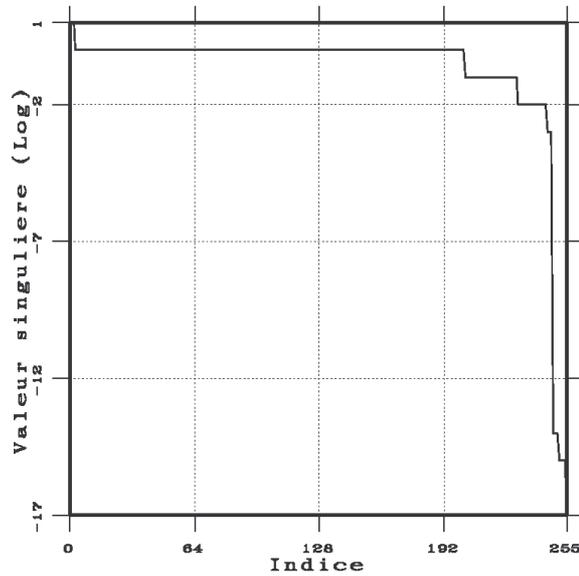


Figure 4-8: Spectre des valeurs singulières de R pour $N=16$ (échelle logarithmique)

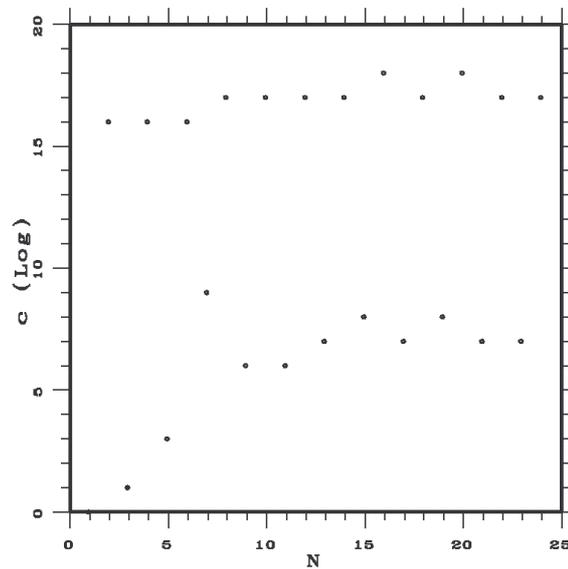


Figure 4-9: Conditionnement c pour N compris entre 1 et 24 (d'après [LAUR94])

Dans tous les cas cependant, la matrice est mal conditionnée. Or, on sait que l'erreur relative sur la solution et l'erreur relative sur la mesure des projections sont liées par la relation:

$$\frac{\|df\|}{\|f\|} \leq c \frac{\|dp\|}{\|p\|} \quad (4-6)$$

Dans le cas où $N=23$ on obtient pour une erreur relative de $1e-3$ sur les projections, une erreur de $1e4$ sur la solution. Dans le cas où $N=24$, la même erreur sur les mesures conduit à une erreur relative supérieure à $1e14$ sur la solution! Le problème est donc dans tous les cas très instable et l'inversion de R est problématique.

2.2.2. Effet de la quantification sur le conditionnement de la matrice

Les valeurs présentées sur la figure 4-9 concernent les matrices non quantifiées. On peut se demander quel est l'effet de la quantification de R sur son conditionnement.

N	16	24	32
Matrice non quantifiée	1,70e18	4,09e17	2,52e17
Matrice quantifiée/déquantifiée	1,45 e18	4,35 e17	5,64 e17

Tableau 4-4: Conditionnement en fonction de N: étude de l'effet de la quantification

C'est pourquoi nous donnons (tableau 4-4) le conditionnement de R après quantification sur 16 bits et déquantification, en fonction de N. On peut constater que le conditionnement reste du même ordre de grandeur que dans le cas où R est stockée en flottant.

3. Description des objets utilisés pour les tests

3.1. Le fantôme numérique Nice

Dans un premier temps, nous avons testé nos algorithmes sur un fantôme numérique défini par J. Darcourt [DARC92] d'après Shepp et Logan [SHEP74]. Cet objet est formé d'ellipses. Nous calculons ses projections en le multipliant par la matrice de Radon. Pour rendre ces projection plus réalistes, nous leur appliquons un bruit de Poisson. Le nombre total de coups¹ dans le vecteur des

¹ Par "nombre de coups", on entend: "nombre de photons détectés". On parle aussi de "taux de comptage".

projections est de 6 millions. Le rapport signal à bruit (SNR) en variance entre les projections bruitées et les projections non bruitées est de 26,70 dB.

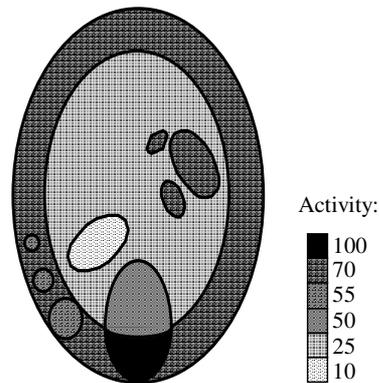


Figure 4-10: Objet Nice
(d'après [DARC92])

Utiliser un fantôme numérique est pratique pour les expérimentations. En effet, on connaît numériquement l'objet à reconstruire. On peut donc faire des mesures d'erreur entre images reconstruites et image originale. Le défaut de ce type de données est qu'elles sont fabriquées avec le même modèle de matrice de Radon que celui utilisé à l'inversion. Elles ne permettent donc pas de savoir si ce modèle de matrice correspond bien à la réalité physique. C'est pourquoi il est également souhaitable de tester nos algorithmes sur des données réelles.

3.2. Le fantôme réel de Jaszczak

Nous avons également testé nos algorithmes sur un fantôme de Jaszczak. Ce fantôme sert à l'étalonnage des appareils utilisés dans plusieurs hôpitaux français. Il est constitué d'une cuve en plexiglas que l'on remplit de liquide radioactif (Technétium ^{99m}Tc). Dans cette cuve sont plongés 6 tubes, vides de liquide radioactif. Ces tubes sont des zones "froides" au sens où ils n'émettent pas de radioactivité. Ce sont ces zones que nous chercherons à détecter.

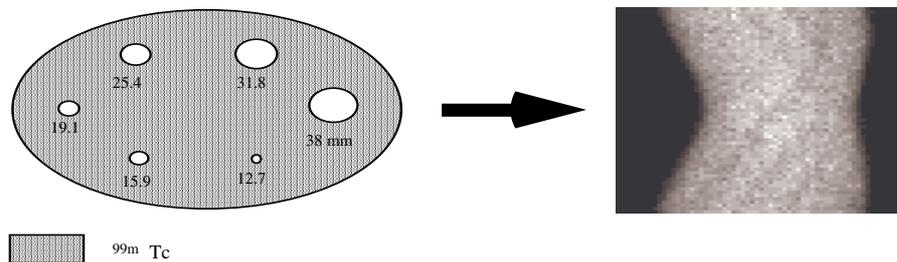


Figure 4-11: Fantôme de Jaszczak (à gauche) et ses projections (à droite) acquises au Centre Antoine Lacassagne de Nice

Les données sont acquises dans les conditions cliniques à l'aide d'une Gamma-caméra. Ce sont donc des données *réelles*. Nous ne connaissons pas numériquement la solution exacte. Il y a environ 250.000 coups dans le vecteur de projections utilisé.

3.3. Les données réelles

Enfin, nous avons testé les algorithmes sur des données réelles, acquises sur un patient anonyme, dans les conditions cliniques. Il y a un total d'environ 300.000 coups dans les projections de la coupe considérée.



Figure 4-12: Données réelles: projections d'un cerveau acquises au Centre Antoine Lacassagne de Nice

Ce patient est atteint d'une hypo-perfusion occipitale. Autrement dit, certaines cellules de l'occiput sont moins bien alimentées en sang. Cela forme une zone froide que l'on doit détecter.

4.Applications des algorithmes à la reconstruction tomographique

Remarque préliminaire: les temps de calcul donnés dans ce paragraphe ont été mesurés sur des stations de travail standard de type DEC 5000/240. Les programmes ont été écrits en C dans l'environnement de programmation Khoros.

4.1. Etude de l'algorithme ARTUR

4.1.1. Fonctionnement de l'algorithme

Cette expérience a pour but de montrer comment se déroule l'algorithme ARTUR. Nous utilisons pour cela les projections bruitées de l'objet Nice 64x64. Nous considérons la fonction de potentiel φ_{GM} . Nous rappelons que l'expression de la fonction de pondération est dans ce cas: $1/(1+u^2)^2$.

Le modèle Markovien mis en œuvre prend en compte les 8 voisins du pixel. Les seuils d'arrêt entre itérations et entre étapes sont choisis égaux à 1e-6. Le seuil d'introduction des discontinuités, δ , et le coefficient de lissage, λ^2 , sont choisis expérimentalement, selon une heuristique que nous décrivons au paragraphe 4.1.2. Le résultat que nous donnons correspond à $\delta = 7$ et $\lambda^2 = 525$. Le programme a effectué 18 étapes, pour un total de 104 itérations. Chaque itération dure environ 10 secondes. Le temps total de reconstruction est donc d'environ 17mn 20s.

- Evolution de la solution:

Nous montrons figure 4-13 (page suivante), l'évolution de la solution au fil des étapes. Pour une meilleure visualisation, la figure 4-13 est "en niveaux de gris", avec une table des niveaux inversée: le noir correspond aux valeurs importantes et le blanc, aux faibles valeurs.

On peut constater que la première solution est très lisse. Elle présente l'aspect flou des solutions obtenues par une régularisation de type Tikhonov avec un fort coefficient de lissage. On voit que par la suite, et de façon très rapide (en 3 ou 4 étapes), les bords extérieurs de l'objet deviennent nets. De manière générale, tous les contours présents dans l'image s'affinent de plus en plus au fil des étapes. Ceci illustre la remarque faite au Chapitre III (paragraphe 2.3): ARTUR est un algorithme à introduction progressive des discontinuités.

Le résultat final est formé de régions homogènes séparées par des bords francs, comme nous le souhaitons. Nous montrons également figure 4-13 l'image de référence, l'objet Nice lui-même. On peut constater visuellement que la solution obtenue est assez proche de la solution idéale. De fait, nous obtenons un SNR VAR de 24,70 dB (soit une erreur relative de 2e-3 par rapport à l'objet Nice).

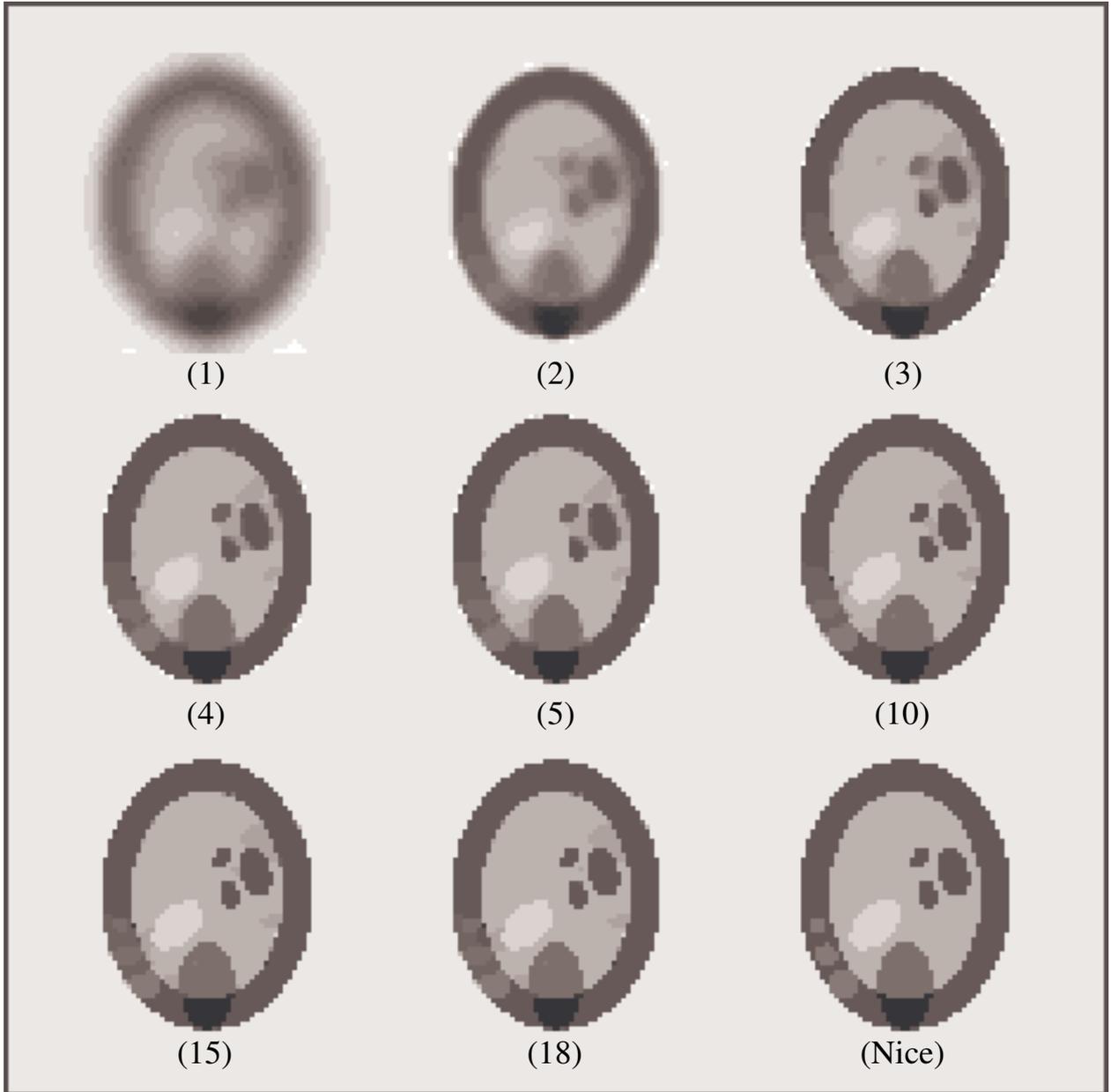


Figure 4-13: Evolution de la solution au fil des étapes (niveaux de gris, inversée)

- Evolution des variables auxiliaires:

Nous montrons pages suivantes l'évolution des variables auxiliaires au fil des étapes. La table des couleurs est en niveaux de gris (du noir pour 0, au blanc pour 1). Plus exactement, nous donnons les cartes des pondérations du Laplacien utilisées par l'algorithme (cf. Chapitre III). Ces pondérations sont données par:

$$\frac{\varphi'(u)}{2u} \text{ dans les directions horizontale et verticale,}$$

$$\frac{1}{2} \frac{\varphi'(u)}{2u} \text{ dans les directions diagonales.}$$

où u représente la différence entre valeurs de pixels voisins, dans la direction considérée, divisée par le paramètre δ .

Considérons tout d'abord les pondérations dans les directions horizontale et verticale. On constate qu'à la première étape, la variable auxiliaire est uniformément égale à 1 dans les deux cas. Ceci est dû au fait que les pondérations à la première étape sont calculées à partir de l'estimée initiale, qui est choisie nulle. Le gradient est donc nul en chaque pixel, et la pondération vaut 1 partout.

A partir de l'étape 2, on voit progressivement apparaître et s'affiner la carte des éléments de contours dans chacune des directions. Nous montrons également les valeurs des pondérations calculées à partir de l'image de référence, avec les mêmes paramètres. On peut voir qu'il reste encore quelques erreurs: quelques discontinuités "parasites" apparaissent, et que certains poids qui devraient être à 1 ne le sont pas tout à fait. On obtient une erreur relative de l'ordre de $1e-2$ par rapport aux pondérations calculées sur l'objet Nice.

Les variables auxiliaires diagonales ont le même comportement général. Toutefois, les pondérations diagonales sont comprises entre 0 et 0,5. C'est pourquoi elles apparaissent en gris sur la figures 4-16.

Comme on peut le constater sur les figures suivantes, une caractéristique importante des algorithmes utilisant la régularisation semi-quadratique est que l'on estime conjointement l'image reconstruite et une carte des contours de cette image.



Figure 4-14: Evolution du processus de ligne vertical au fil des étapes (niveaux de gris)

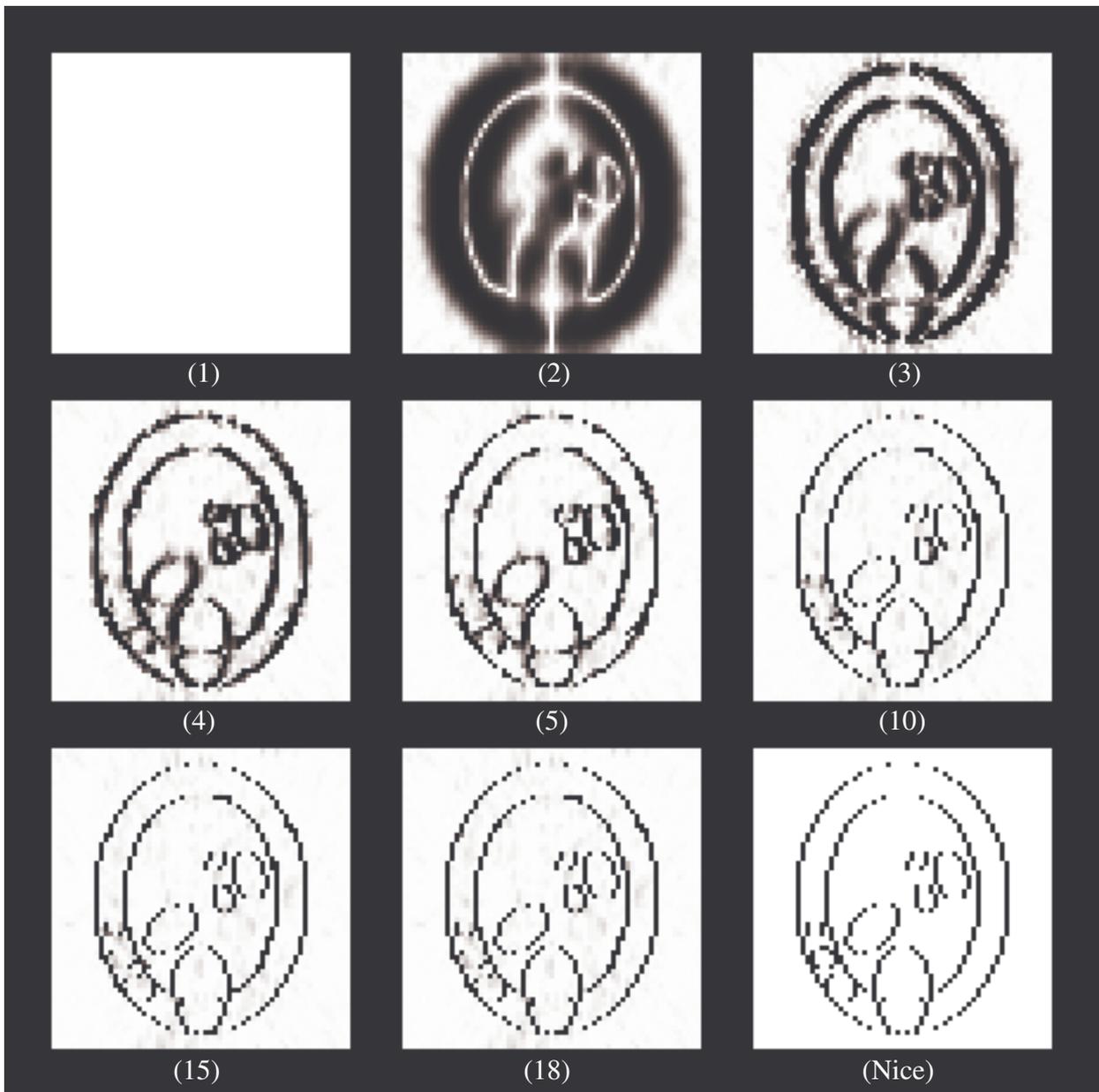


Figure 4-15: Evolution du processus de ligne horizontal au fil des étapes (niveaux de gris)

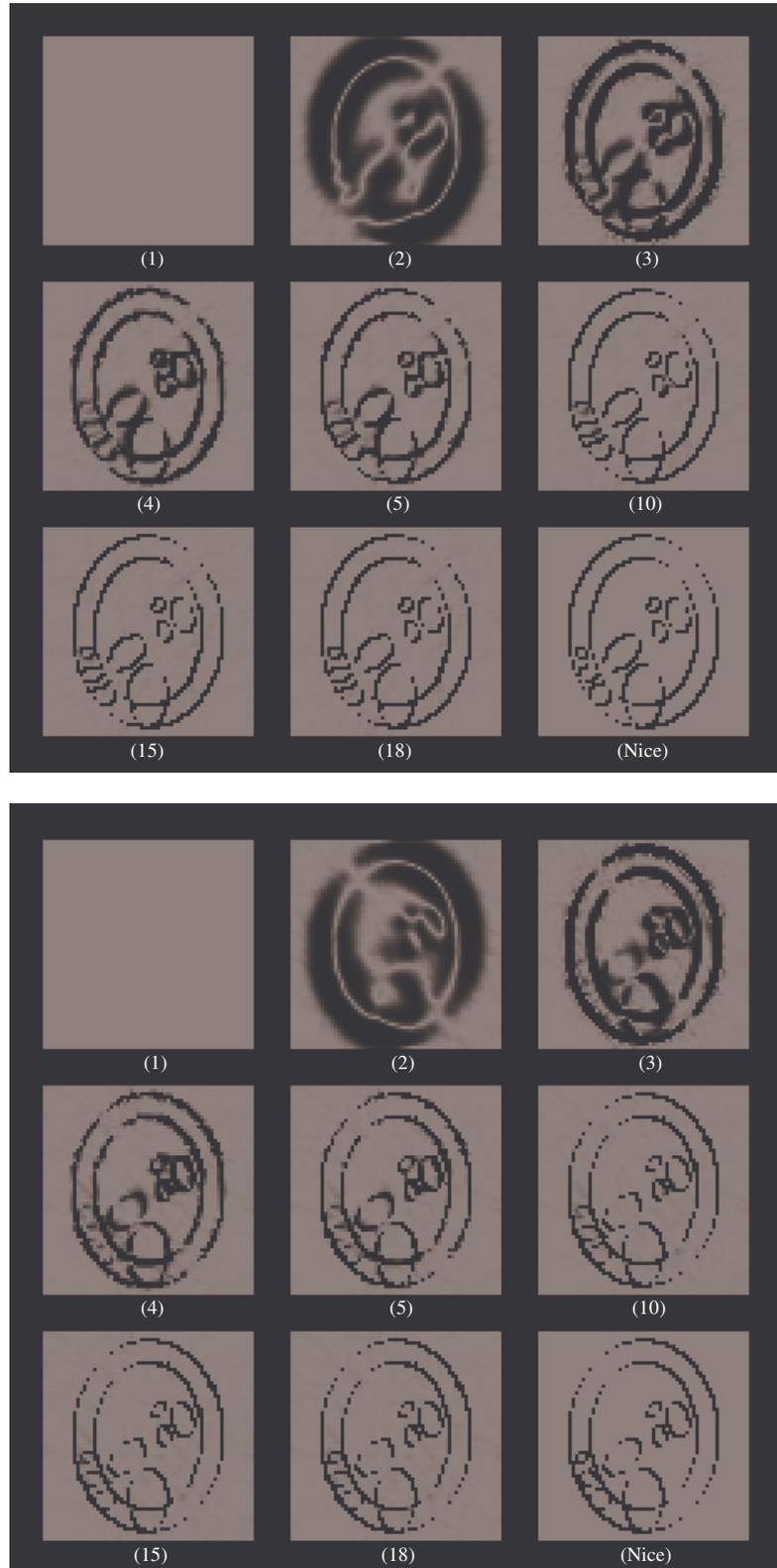


Figure 4-16: Evolution des variables auxiliaires diagonales au fil des étapes (niveaux de gris)

- Evolution de la différence entre estimées successives:

Les différences entre estimées successives sont une mesure du “changement” intervenant sur la solution, entre ces deux estimées. Nous nous intéressons ici à la différence relative entre estimées à deux *itérations* successives:

$$\frac{\|f^{n+1,m+1} - f^{n+1,m}\|^2}{\|f^{n+1,m}\|^2}$$

et à deux *étapes* successives:

$$\frac{\|f^{n+1} - f^n\|^2}{\|f^n\|^2}$$

Nous mesurons leur évolution au fil du déroulement de l’algorithme (figure 4-17).

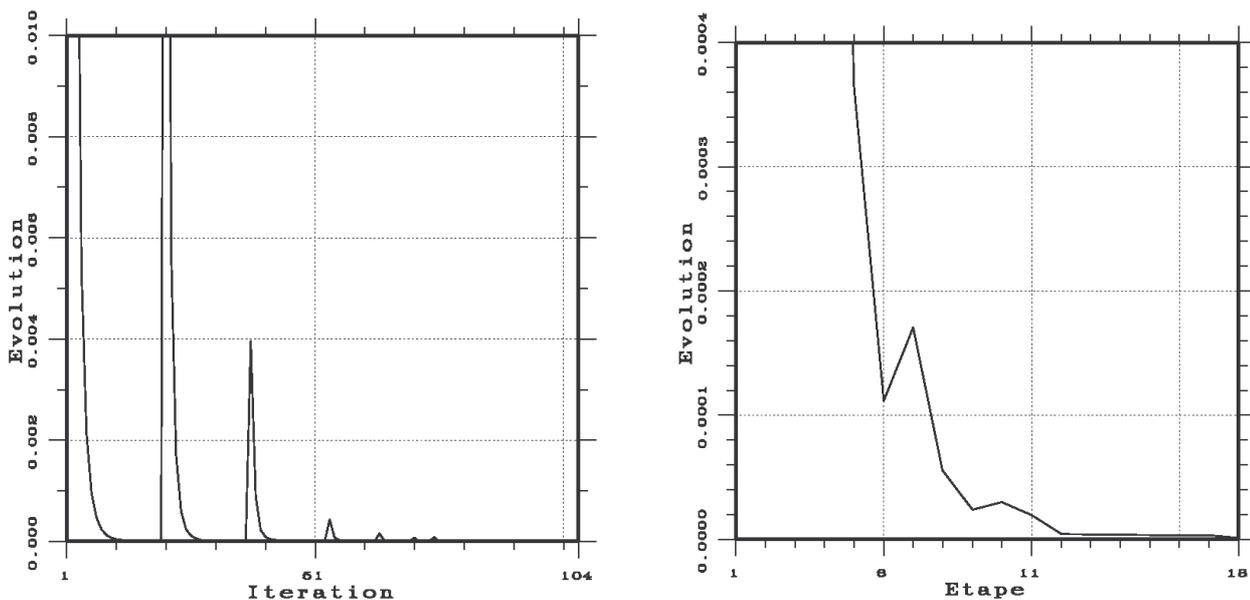


Figure 4-17: Evolution de la solution entre deux itérations successives (à gauche), entre deux étapes successives (à droite)

Examinons tout d’abord l’évolution de la solution entre itérations (à gauche sur la figure 4-17). La première étape de l’algorithme a duré 20 itérations. La seconde a duré de l’itération 21 à l’itération 37, la troisième de l’itération 38 à l’itération 53, etc...

Ces étapes sont caractérisées par des pics sur la courbe. En début d’étape, les pondérations sont nouvelles, et il y a beaucoup de changement entre les estimées successives. Au fur et à mesure du déroulement de l’étape, l’algorithme de minimisation en f converge, et il y a de moins en moins de

changement entre deux itérations. On voit aussi que les pics correspondant aux changements d'étapes sont de plus en plus faible amplitude. Autrement dit, il se passe de moins en moins de choses en début d'étape à partir de la 4ème. Ceci correspond au fait que les variables auxiliaires sont de moins en moins modifiées au fur et à mesure du déroulement de l'algorithme.

La courbe de l'évolution entre étapes (à droite sur la figure 4-17) confirme cette constatation: la différence entre estimées à deux étapes successives à l'étape 5 est déjà très faible (inférieure à $1e-4$). Les changements opérés sur la solution entre étapes successives sont donc concentrés dans les premières étapes. Toutefois, on ne peut pas dire en toute rigueur qu'il se passe "de moins en moins de choses" au fil des étapes. En effet, la courbe présente des "rebonds", par exemple entre l'étape 6 et l'étape 7. Il semble donc qu'il y ait eu un changement plus important à l'étape 7 qu'à l'étape 6. Cette présence de rebonds est caractéristique dans le cas de potentiels non convexes. Néanmoins, la courbe se stabilise et l'algorithme s'arrête lorsque la différence entre deux étapes successives est inférieure à $1e-6$.

- Evolution de la norme de la solution et des variables auxiliaires:

Nous nous intéressons maintenant à l'évolution de la norme de la solution et de la variable auxiliaire au cours des itérations. Pour cela, nous traçons figure 4-18 la RMS (de l'anglais "Root Mean Square") de ces images en fonction de l'itération (par définition, $RMS(x) = \frac{1}{N} \|x\|$)

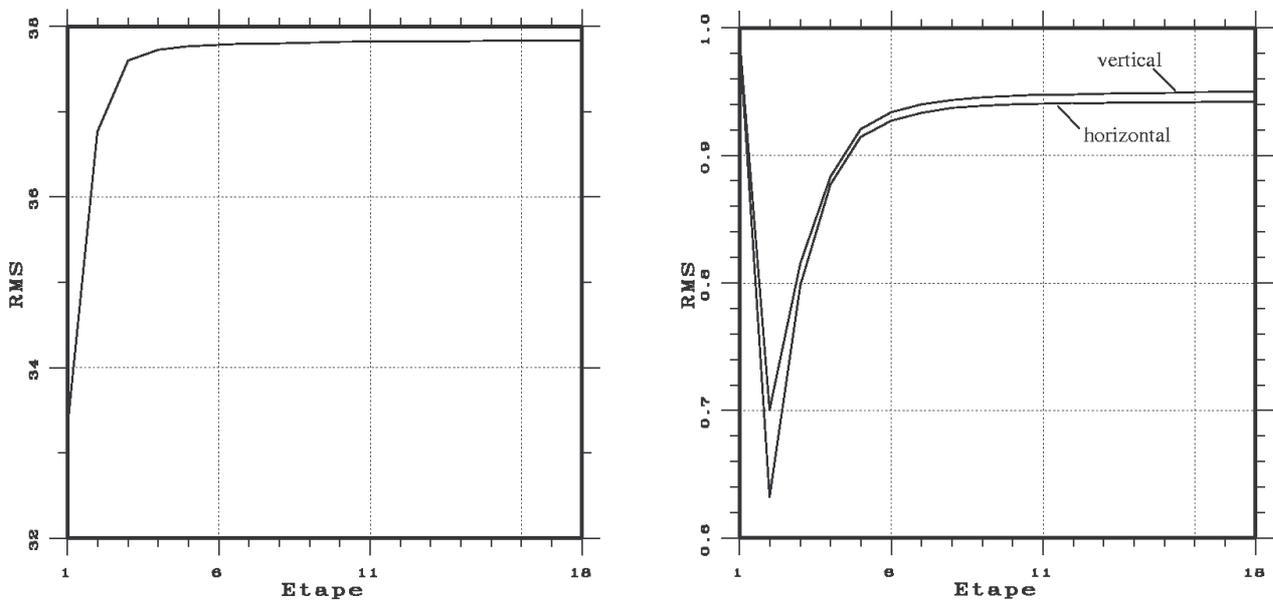


Figure 4-18: Evolution de la RMS de la solution (à gauche) et des processus de ligne (à droite) au fil des étapes

On peut constater que la RMS de la solution augmente toujours en fonction des étapes. On tend vers

une norme de 37,84. A titre indicatif, la RMS de l'image Nice est 37,88.

Pour ce qui est de la norme des processus de ligne, on voit que la RMS chute entre la première et la seconde étape et qu'à partir de la seconde étape, elle augmente toujours. A la première étape, nous avons vu précédemment que la variable auxiliaire vaut 1 partout (...ou presque, à cause du traitement des bords, voir Chapitre III). La RMS est donc proche de 1. Au début de la seconde étape, on introduit beaucoup de zéros dans la carte des discontinuités (cf. figures 4-15 et 4-14, pages précédentes). La norme des variables auxiliaires chute donc. Par la suite, la carte des discontinuités s'affine, autrement dit le nombre de points à zéro diminue. C'est pour cela qu'on observe une croissance de la RMS. On tend vers une limite de 0,95 pour le processus vertical et de 0,94 pour le processus horizontal. A titre indicatif, les RMS des variables auxiliaires calculées sur la solution exacte sont à 0,97 pour le processus vertical et de 0,95 pour le processus horizontal.

Remarque: la variation est la même pour les processus diagonaux.

- Evolution du critère du MAP et de l'erreur relative par rapport à la solution exacte:

Nous présentons ici les courbes des énergies en fonction des itérations. Nous proposons les courbes du terme d'attache aux données: $J_1(f^n)$, de l'énergie a priori: $J_2(f^n)$ et de l'énergie totale du MAP: $J(f^n) = J_1(f^n) + J_2(f^n)$.

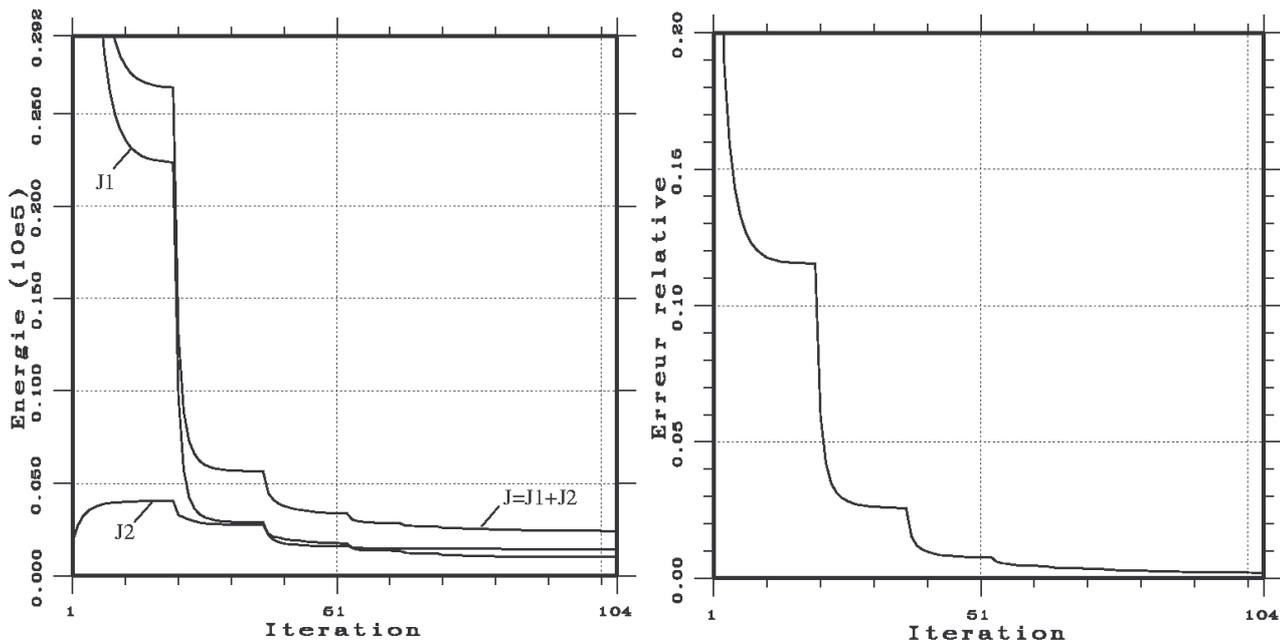


Figure 4-19: Evolution des énergies (à gauche) et de la NMSE par rapport à la solution exacte (à droite) au fil des itérations

Sur ces courbes, on distingue nettement les différentes étapes. Pour ce qui est des énergies, le terme de vraisemblance $J_1(f^n)$ diminue toujours. Quant au terme de régularisation, initialement nul (l'estimée initiale est prise nulle partout), il augmente à la première étape. Ensuite, $J_2(f^n)$ diminue toujours. L'énergie totale du MAP diminue toujours et converge vers une certaine valeur. Ceci est normal, étant donné ce que nous avons montré au Chapitre III. En effet, le théorème 3 nous dit que la suite des énergies duales $J_n = J^*(f^n, b^{n+1})$ est décroissante et convergente. Étant donnée la définition de J^* , cela revient à dire que la suite des valeurs du critère du MAP, mesurées à chaque étape, décroît vers une valeur limite. C'est bien ce que nous observons.

Nous présentons également figure 4-19 une courbe décrivant l'évolution de la qualité de l'estimée en fonction des itérations. Pour cela nous mesurons à chaque itération l'erreur relative (Normalized Mean Square Error ou NMSE) entre l'estimée à l'étape n et l'image de référence Nice. On constate que l'erreur relative par rapport à la solution exacte diminue toujours et converge vers la valeur $2e-3$.

- En conclusion:

Cet exemple nous a permis d'étudier le fonctionnement de l'algorithme. Partant d'une solution lisse "de Tikhonov", ARTUR se sert des tableaux des pondérations, calculés au début de chaque étape, pour introduire progressivement des discontinuités franches dans l'image estimée. La solution obtenue finalement est donc formée de zones homogènes séparées par des bords francs, comme nous le souhaitons. De plus, cette solution est très proche de l'image de référence.

Comme nous l'assurent les résultats du théorème 3, l'énergie du MAP diminue à chaque étape et tend vers une valeur limite. Nous observons de plus une convergence, en norme, de la solution et de la variable auxiliaire. L'erreur par rapport à l'image originale décroît également jusqu'à une valeur limite.

Même si le potentiel utilisé est *non convexe* dans cet exemple, l'algorithme converge vers une solution d'excellente qualité. Dans nos expérimentations, nous avons toujours observé ce comportement.

4.1.2. Réglage des paramètres

Cette expérience est faite dans les conditions décrites au paragraphe précédent. Nous nous intéressons ici au réglage des paramètres par l'utilisateur.

$\delta = 9,5$			
$\delta = 8$			
$\delta = 6,5$			
	$\lambda^2 = 275$	$\lambda^2 = 400$	$\lambda^2 = 525$

Figure 4-20: Solutions obtenues pour différentes valeurs des paramètres

Dans tous les cas, l'algorithme a convergé, en une centaine d'itérations environ. Les images obtenues (cf. figure 4-20) sont toutes de bonne qualité. En ce sens, le programme est assez robuste vis-à-vis des paramètres. On peut cependant voir que si on lisse peu et si on choisit une hauteur des discontinuités un peu trop faible, (ex: $\delta = 6,5$, $\lambda^2 = 275$) des pics de bruit apparaissent dans l'image reconstruite. A l'inverse, si on lisse beaucoup et si on fixe le seuil de discontinuités un peu trop haut (ex: $\delta = 9,5$, $\lambda^2 = 525$), on lisse certains détails de l'image.

Les paramètres du programme sont: le seuil de détection des discontinuités δ et le coefficient de régularisation normalisé λ^2 . Le coefficient λ^2 est relié au coefficient de lissage λ^2 , intervenant dans l'expression du critère du MAP, par la relation: $\lambda^2 \delta^2 = \lambda^2$ (cf. Chapitre III). Nous avons choisi δ et λ^2 comme paramètres du programme car ce sont eux qui interviennent dans l'expression de l'énergie duale et donc dans l'expression des équations normales associées. Comme le programme travaille directement au niveau des équations normales, ces paramètres sont les plus naturels.

Pour trouver les meilleures valeurs des paramètres, nous utilisons l'heuristique suivante: nous commençons par fixer la valeur de λ^2 et chercher la valeur de δ qui donne le meilleur résultat. Ensuite, nous faisons varier en même temps λ^2 et δ de façon à garder λ^2 à peu près constant. Bien sûr, ce n'est qu'une façon de procéder mais elle se révèle assez pratique à l'usage, dans notre cas.

4.1.3. Utilisation de différents potentiels

Dans ce paragraphe, nous présentons les résultats obtenus avec différents potentiels, convexes et non-convexes, à partir des projections bruitées de l'objet Nice (cf. figure 4-21). Nous utilisons un 8-voisinage. Les seuils d'arrêt sont fixés à 1e-6, entre itérations comme entre étapes. Les valeurs des paramètres sont données dans le tableau 4-5.

	Quadratique	φ_{GR}	φ_{HL}	φ_{GM}
δ	---	6,5	7	7
λ^2	10	525	525	525

Tableau 4-5: Valeurs des paramètres utilisés

Dans le cas de Tikhonov (potentiel quadratique), le coefficient de lissage a été choisi expérimentalement de façon à s'affranchir au mieux des effets du bruit sans trop abîmer les contours: si on lisse plus, on obtient une image floue, si on lisse moins on obtient une image très bruitée. On remarque aussi que la valeur de ce coefficient de lissage est très inférieure à celle utilisée dans le cas

des potentiels préservant les discontinuités. On peut donc lisser beaucoup plus fort avec ce type de potentiels, sans abîmer les contours.

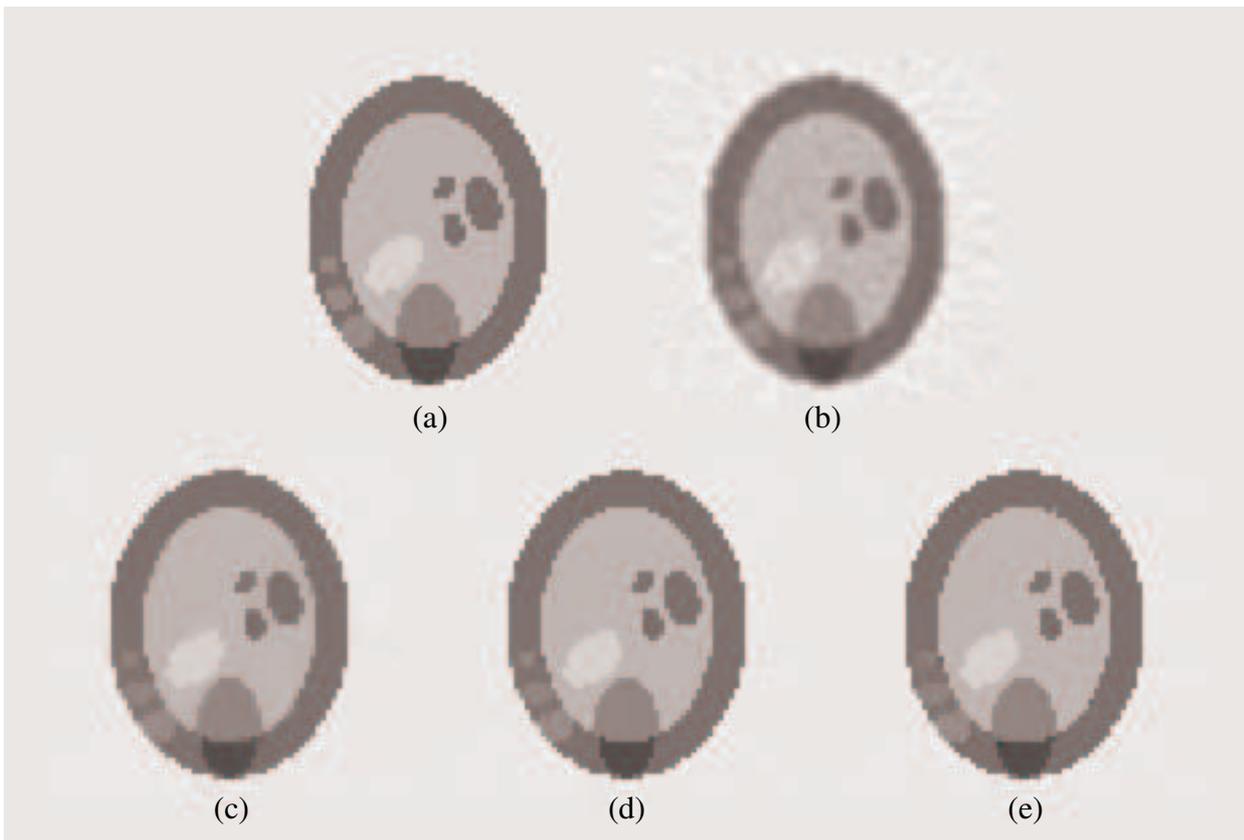


Figure 4-21: Objet Nice (a) et images reconstruites avec différentes fonctions de potentiel. On utilise en (b) le potentiel quadratique (Tikhonov); en (c) la fonction φ_{GR} ; en (d) la fonction φ_{HL} et en (e) φ_{GM} .

Les résultats obtenus montrent la supériorité de la régularisation avec préservation des discontinuités sur la régularisation de type Tikhonov classique. D'autre part, ils montrent qu'avec une fonction convexe (ici la fonction log cosh, mais on obtient un résultat identique avec φ_{HS}), il est possible de préserver les discontinuités. On peut tout de même constater que certains contours ne sont pas très nets. Pour confirmer ces impressions, nous proposons de visualiser des coupes des images reconstruites (cf. figure 4-22).

Pour cela, nous avons extrait la colonne 27 de l'image originale et des images reconstruites. Nous avons choisi cette partie de l'image car elle contient quatre échelons, de tailles différentes.

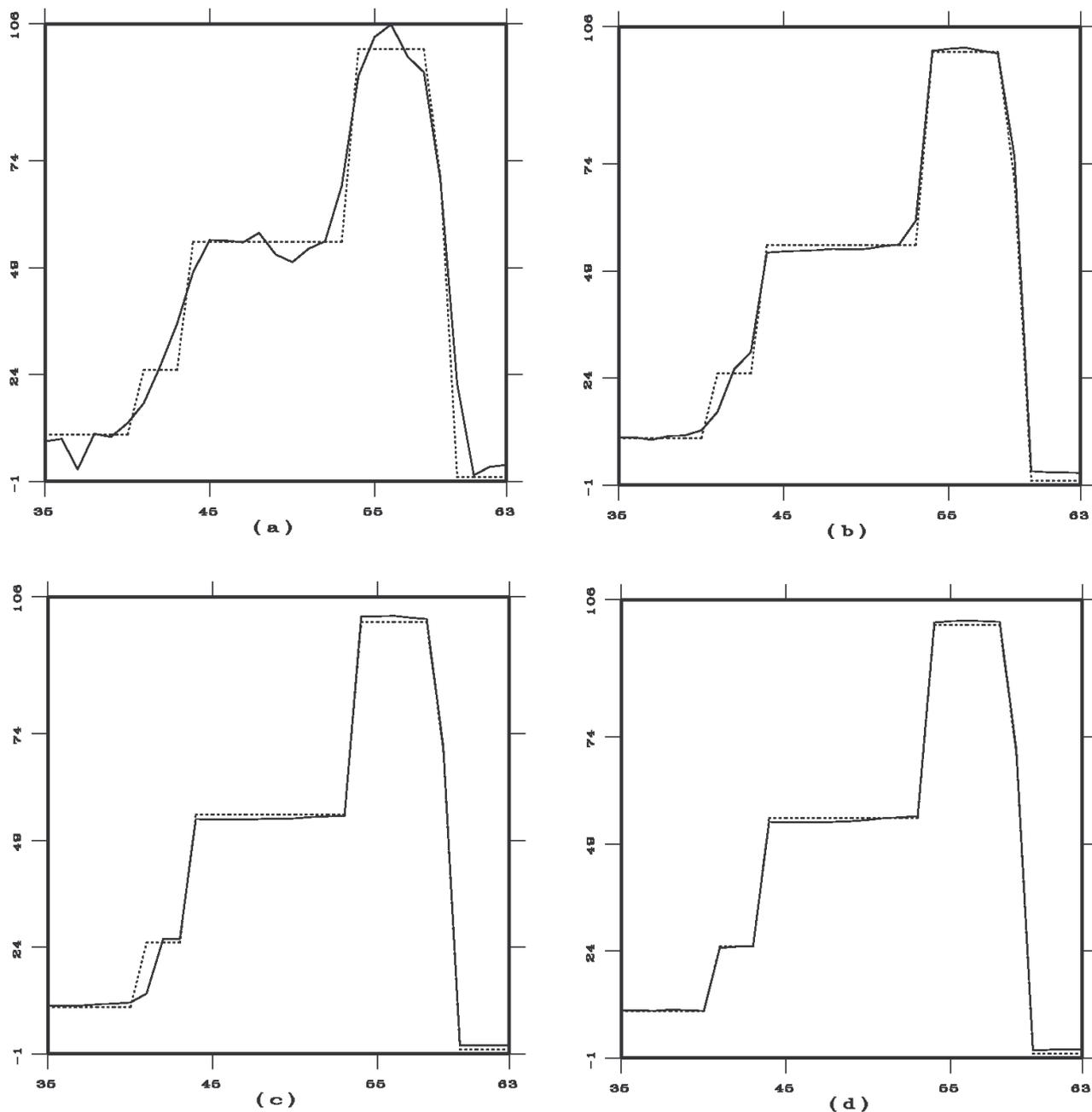


Figure 4-22: Coupes de l'objet Nice (en pointillés) et des images reconstruites (traits pleins). On utilise: (a) le potentiel quadratique (Tikhonov); (b), le potentiel convexe φ_{GR} ; (c) φ_{HL} et (d) φ_{GM} (potentiels non convexes)

Sur la coupe (a), on voit bien l'effet de lissage induit par le modèle quadratique utilisé. On ne distingue

plus les deux échelons de plus faible amplitude. On observe des pics résiduels dus au bruit. Si on lissait plus, ces pics disparaîtraient, mais cela se ferait au détriment des contours.

La coupe (b) représente le résultat obtenu avec la fonction $\log \cosh$ de Green. On peut voir que les échelons de plus forte amplitude sont bien préservés. Toutefois, on constate la présence d'un empatement au bas de certains échelons. Cela peut s'expliquer de façon "physique". En effet, nous avons vu au Chapitre II que la fonction de Green est "jumelle" de la fonction ϕ_{HS} , et se comporte donc comme elle. Nous avons fait remarquer que l'usage de cette dernière fonction revient à imposer une contrainte à la longueur totale des courbes décrites par les lignes et les colonnes de l'image.

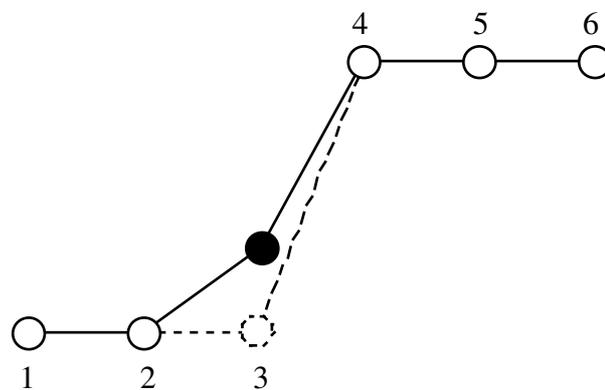


Figure 4-23: Comportement de la fonction $\log \cosh$ dans le cas d'un échelon

Examinons le cas d'un échelon (cf. figure 4-23). Supposons que le bruit ait tendance à faire "remonter" légèrement la valeur du pixel 3 à la valeur indiquée par un rond noir. Nous voudrions que le terme de régularisation "ramène" ce pixel à la position indiquée en pointillés. Ce n'est pas possible avec ce type de potentiel. En effet, le profil tracé en pointillés est plus long que celui tracé en trait plein. La fonction de régularisation va donc privilégier la seconde possibilité et la valeur du pixel ne sera pas modifiée.

On peut interpréter cette existence d'empattements d'une autre façon. En théorie, la discontinuité entre les pixels 3 et 4 étant détectée, la pondération correspondante du Laplacien devrait être nulle. Dans ce cas, il n'y aurait plus d'interaction entre les pixels 3 et 4. N'étant plus "retenu" par le pixel 4, le pixel 3 devrait donc prendre la valeur indiquée en pointillés, lorsqu'on applique le Laplacien en 2. En fait, la fonction de pondération associée à la fonction $\log \cosh$ ne décroît pas assez vite. La pondération du Laplacien n'est pas tout à fait nulle. On ne "casse" donc pas parfaitement l'interaction entre le pixel 3 et le pixel 4. C'est ce qui empêche le pixel 3 de prendre la valeur désirée. Ce phénomène est d'autant plus important que la discontinuité est de faible amplitude. Cela explique la forme du profil observé figure 4-22.

Les fonctions de potentiel non convexes donnent de meilleurs résultats. Les discontinuités sont

marquées plus franchement. Toutefois, on observe un défaut sur l'échelon de plus faible amplitude dans le profil (c). Contrairement au cas précédent, ce défaut n'est pas dû à la forme de la fonction de pondération mais au fait que la fonction de potentiel est *non convexe*. Dans ce cas, l'algorithme ne trouve pas forcément le minimum global du critère du MAP. C'est ce qui arrive ici (voir paragraphe suivant, figure 4-24).

- En conclusion, on peut donc dire que les potentiels convexes permettent la prise en compte des discontinuités. Toutefois, la qualité des contours est moins bonne que dans le cas des potentiels non convexes.

4.1.4. Influence de la condition initiale

Au chapitre III, nous avons proposé dans l'algorithme ARTUR de partir d'une estimée initiale uniformément nulle. On peut toutefois se poser la question de savoir ce qui se passerait si on choisissait une autre condition initiale. Nous proposons figure 4-24 les profils des solutions obtenues avec les mêmes paramètres qu'au paragraphe précédent, mais avec deux conditions initiales différentes, et avec des seuils de précision à $1e-8$ (de façon à bien laisser converger l'algorithme).

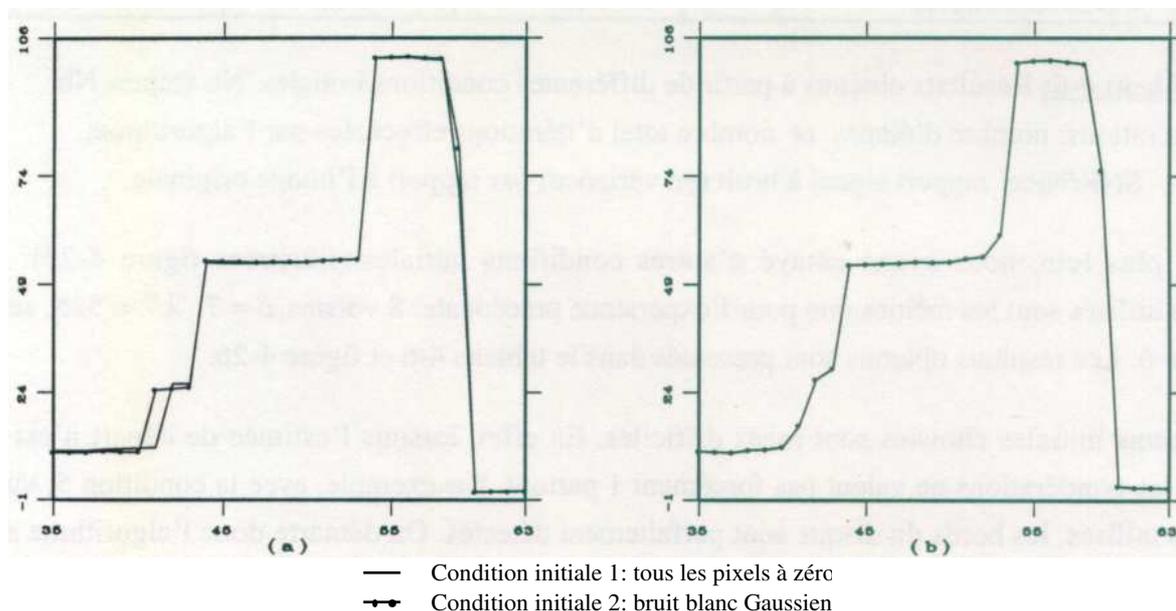


Figure 4-24: Profils des images reconstruites pour deux conditions initiales différentes avec: le potentiel non convexe $\varphi = \varphi_{HL}$ (a) et le potentiel convexe $\varphi = \varphi_{GR}$ (b)

La première condition initiale consiste à prendre l'image nulle. La seconde condition initiale consiste à partir d'un bruit blanc Gaussien de moyenne nulle et de variance 10. On peut voir que la solution obtenue est la même dans le cas d'un potentiel convexe. On vérifie bien ce que nous avons prouvé au Chapitre III: ARTUR converge vers le minimum global unique du critère du MAP.

Par contre, dans le cas du potentiel non convexe (fonction d'Hebert et Leahy φ_{HL}), on voit que la solution obtenue est différente dans le second cas. En particulier, le défaut observé au niveau de l'échelon de plus faible amplitude a disparu. L'algorithme converge vers une solution cohérente, mais que nous n'avons pas encore caractérisée mathématiquement.

Condition initiale choisie	Nb. étapes	Nb. iterations	SNR/Nice
1. Tous les pixels à 0	56	316	26,88 dB
2. Bruit Blanc Gaussien (Var. 10, moy. 0)	31	252	24,62 dB
3. Bruit Blanc Gaussien (Var. 100, moy. 10)	55	332	23,34 dB
4. Extrait de l'image "lena"	35	294	23,48 dB
5. Disque centré (fond à 0, disque à 255)	21	244	26,94 dB
6. Image "waves+shot noise"	25	478	25,20 dB

Tableau 4-6: Résultats obtenus à partir de différentes conditions initiales. Nb étapes, Nb itérations: nombre d'étapes et nombre total d'itérations effectuées par l'algorithme; SNR/Nice: rapport signal à bruit (en variance) par rapport à l'image originale

Pour aller plus loin, nous avons essayé d'autres conditions initiales (illustrées figure 4-25). Les paramètres utilisés sont les mêmes que pour l'expérience précédente: 8 voisins, $\delta = 7$, $\lambda^2 = 525$, seuils d'arrêt à $1e-6$. Les résultats obtenus sont présentés dans le tableau 4-6 et figure 4-26.

Les conditions initiales choisies sont assez difficiles. En effet, lorsque l'estimée de départ n'est pas constante, les pondérations ne valent pas forcément 1 partout. Par exemple, avec la condition 5, vu les paramètres utilisés, les bords du disque sont parfaitement détectés. On démarre donc l'algorithme avec une carte des discontinuités totalement inadaptée au problème. Autrement dit, on voit qu'on ne calcule pas forcément la solution de Tikhonov à la première étape. On ne se trouve pas dans les conditions normales d'utilisation de l'algorithme, décrites au Chapitre III.

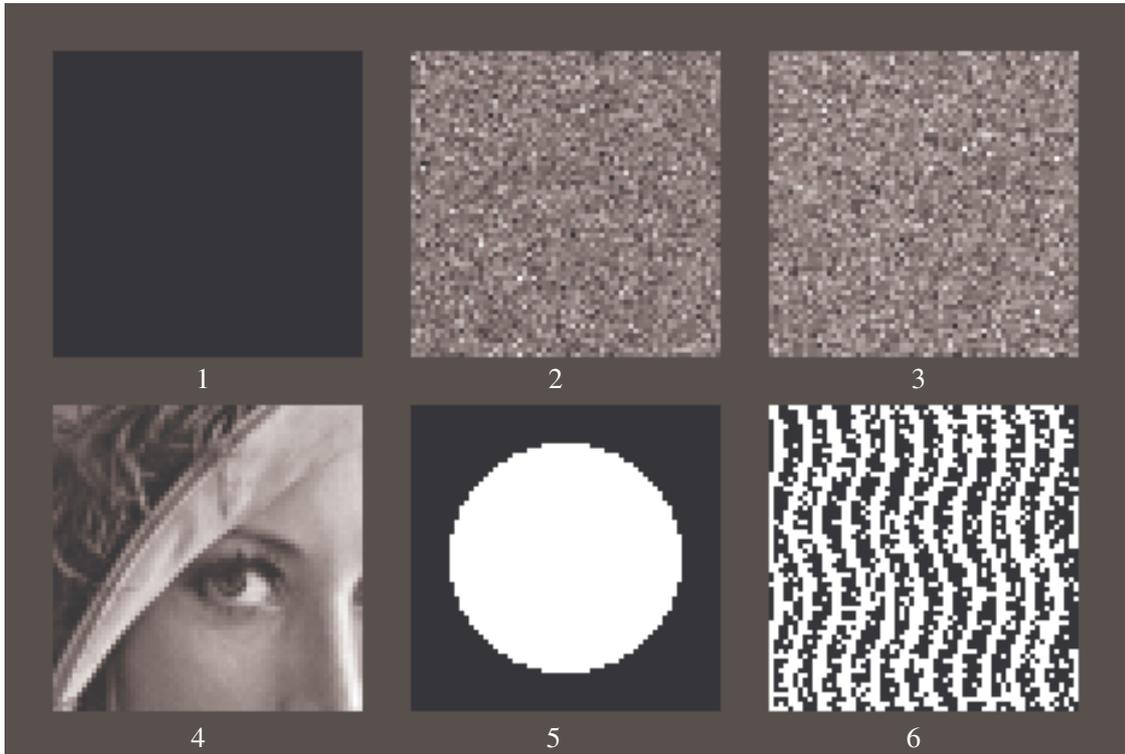


Figure 4-25: Conditions initiales utilisées

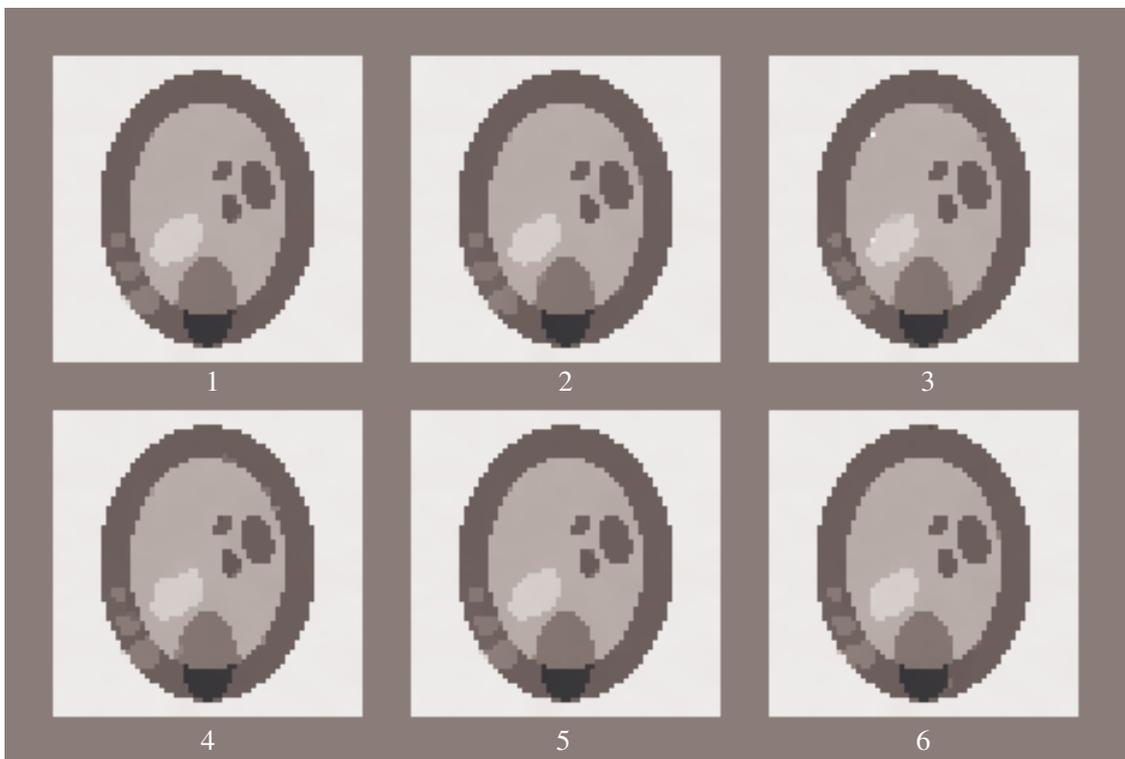


Figure 4-26: Résultats obtenus avec les différentes conditions initiales (table de couleurs: niveaux de gris, inversée)

Les résultats obtenus avec les différentes conditions initiales sont assez bons, que ce soit sur le plan visuel ou sur le plan du SNR. L'algorithme converge dans tous les cas vers une solution de bonne qualité. Cependant, on note des différences notables entre les différentes solutions obtenues.

- En conclusion: dans le cas où on utilise un potentiel convexe, nous avons vérifié expérimentalement que l'algorithme converge bien vers le minimum unique du critère du MAP, quel que soit le point de départ. Dans le cas non convexe, nous avons essayé diverses conditions initiales. Ces conditions initiales sont assez difficiles. L'algorithme n'est pas utilisé dans son mode de fonctionnement normal. Malgré cela, ARTUR a toujours convergé et fourni une solution cohérente. En ce sens, l'algorithme apparaît donc assez robuste. Par contre, il est clair que la solution trouvée dans le cas non convexe dépend de l'estimée initiale.

4.1.5. Influence du voisinage utilisé

Dans cette expérience, nous utilisons toujours les projections bruitées de l'objet Nice. Nous proposons de comparer les reconstructions obtenues en utilisant un voisinage à 4 ou 8 voisins. Nous fixons le paramètre δ à 7, les seuils d'arrêt à $1e-6$. Le paramètre λ^2 est fixé à 525 pour 8 voisins et à 1050 pour 4 voisins. Nous montrons figure 4-27 les résultats obtenus avec la fonction de Geman et McClure: φ_{GM} .

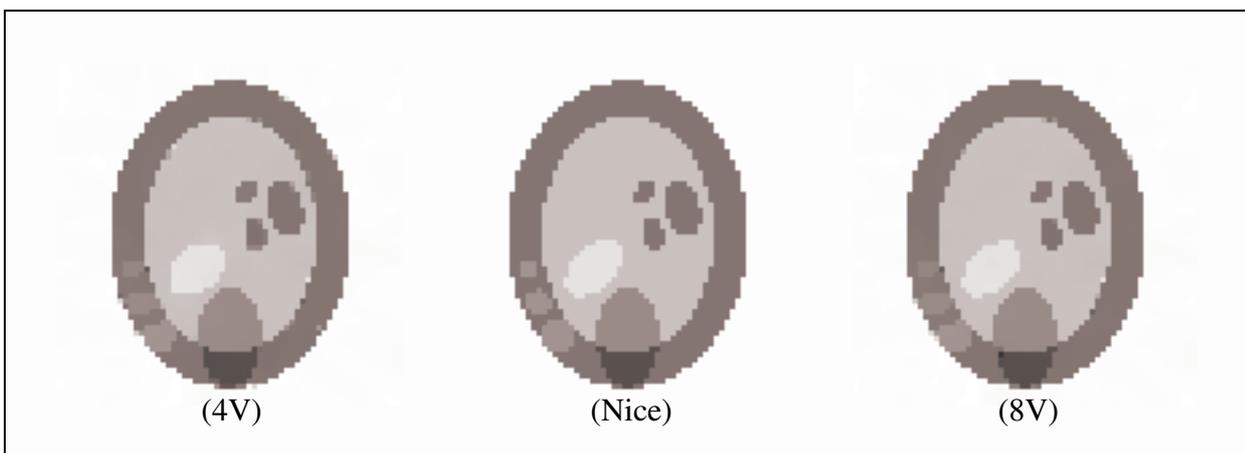


Figure 4-27: Résultat obtenu avec 4 voisins (4V) et 8 voisins (8V). Au centre: objet Nice (niveaux de gris, inversée). Fonction utilisée: φ_{GM}

On peut voir que les deux images obtenues présentent le même "degré de lissage". Par contre, au niveau des détails, l'image obtenue avec 8 voisins est meilleure. Cela se ressent au niveau du SNR. On obtient un rapport signal-à-bruit de 24,70 dB avec 8 voisins et de 22,52 dB avec 4 voisins. Les temps de calcul mesurés dans les deux cas sont très proches (9 s 46 par itération avec 4 voisins, 9 s 65 par

itération avec 8 voisins.

	φ_{HL}	φ_{GM}	φ_{GR}
4 Voisins	25,50 dB	22,52 dB	21,50 dB
8 Voisins	26,07 dB	24,70 dB	22,05 dB

Tableau 4-7: SNR par rapport à l'image originale avec différents potentiels, et avec les deux types de voisinage utilisés

Les mêmes observations sont valables pour les autres fonctions de potentiel, qu'elles soient convexes ou non (cf tableau 4-7). Etant donné que cela n'augmente quasiment pas la charge de calcul à chaque itération, on a donc intérêt à utiliser un 8-voisinage.

4.1.6. Application à des données réelles

Nous présentons page suivante les résultats obtenus avec les données réelles: le fantôme de Jaszczak et la coupe de cerveau. En ce qui concerne les résultats obtenus avec le fantôme de Jaszczak (figure 4-28), on peut faire les constatations suivantes:

- les résultats obtenus par ARTUR avec différentes fonctions de potentiels sont d'un aspect assez proche. On peut cependant noter que le résultat obtenu avec le potentiel convexe φ_{GR} est moins bon que les deux autres, au sens où les discontinuités apparaissent moins nettement.
- Sur ces images, on voit que la répartition d'activité n'est pas homogène dans l'objet. Ceci est dû au phénomène d'atténuation, qui n'est pas corrigé ici. De meilleurs résultats ont été obtenus par P.M. Koulibaly [KOUL95] en prenant en compte le phénomène d'atténuation lors de la construction de la matrice.
- Les images obtenues avec ARTUR sont proches de celle obtenue par l'algorithme MAP-EM-OSL. Ce dernier utilise le même modèle *a priori*. Par contre, la vraisemblance est considérée Poissonienne dans le cas du MAP-EM-OSL alors qu'on la suppose Gaussienne dans le cas d'ARTUR. On voit que dans ce cas, l'approximation Gaussienne n'a pas beaucoup d'influence sur la qualité de l'image obtenue.
- Enfin, on note la différence d'aspect entre les images obtenues à l'aide des méthodes algébriques régularisées et l'image calculée à l'aide de l'algorithme de filtrage-rétroprojection. Ce dernier est utilisé de façon courante dans les hôpitaux.

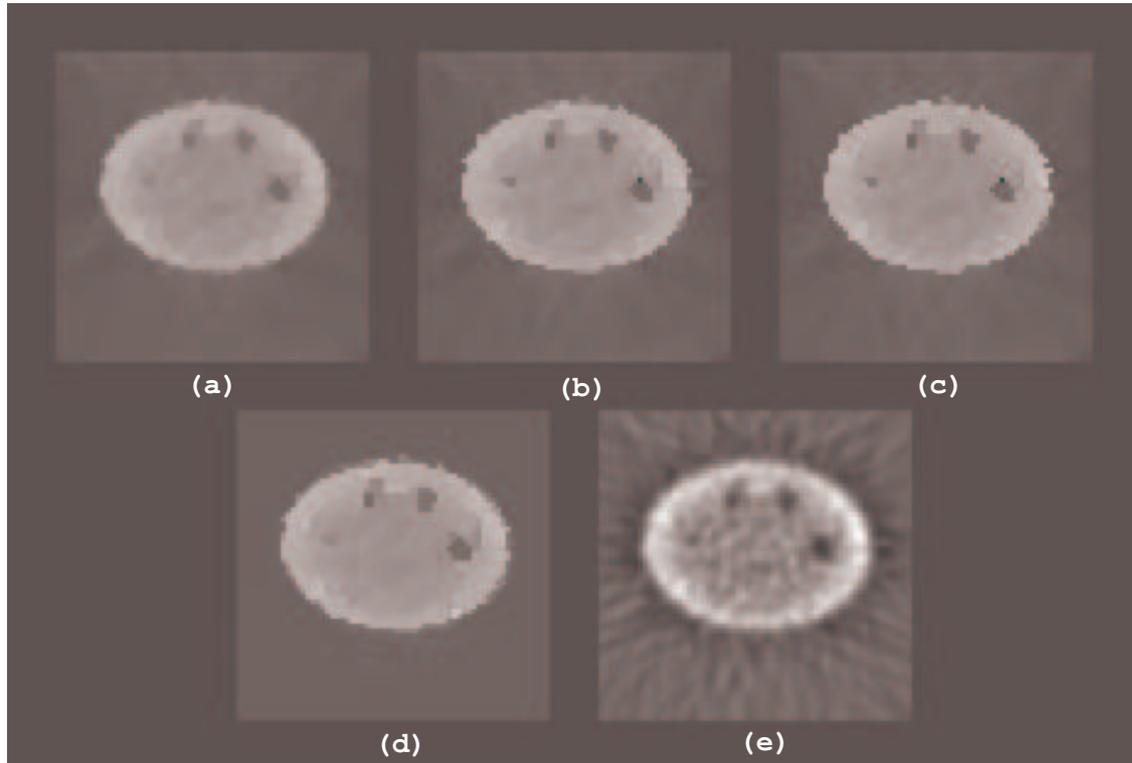


Figure 4-28: Fantôme de Jaszczak: reconstructions avec ARTUR (en haut). On utilise les potentiels φ_{GR} (a), φ_{HL} (b) et φ_{GM} (c). En bas: reconstruction avec le MAP-EM-OSL pour $\varphi = \varphi_{GM}$ (d) et l'algorithme de filtrage rétroprojection (e) utilisé couramment à l'hôpital (niveau de gris).

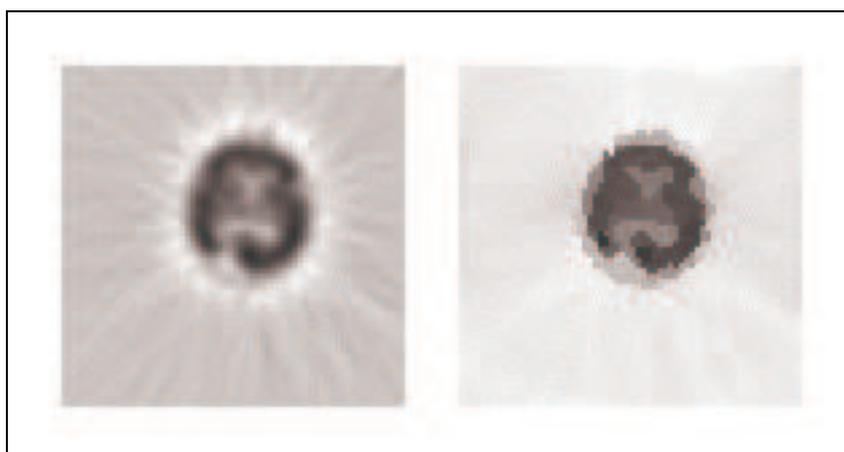


Figure 4-29: Données réelles: reconstruction d'une coupe de cerveau. A gauche: filtrage-rétroprojection, à droite ARTUR avec φ_{HL} (niveau de gris, inversée).

Sur l'image du cerveau (figure 4-29), on note la même différence d'aspect entre le résultat obtenu par ARTUR et le résultat du filtrage-rétroprojection. La lacune occipitale apparaît nettement sur les deux images. On peut noter que les contours du cerveau ne sont pas très bien définis sur ces images. En fait, cela pouvait déjà être noté sur le sinogramme, montré au paragraphe 3.3, dont les bords ne sont pas très nets. Cela peut être dû aux phénomènes physiques intervenant lors de l'acquisition des données. Cet exemple confirme que l'on a beaucoup à attendre de la correction des phénomènes physiques.

4.2. Etude de l'algorithme LEGEND

4.2.1. Marche de l'algorithme

Le but de cette expérience est de montrer comment fonctionne l'algorithme LEGEND. Tout comme ARTUR, l'algorithme LEGEND résout de manière itérative une série de systèmes linéaires de la forme: $Af^{n+1} = c$. A la première étape, en partant d'une image identiquement nulle, ARTUR et LEGEND résolvent le même système, correspondant à une régularisation de Tikhonov. Par la suite, à chaque étape, dans le cas de LEGEND, on modifie le second membre de ce système. Dans le cas d'ARTUR, on modifie les coefficients de la matrice A . On peut donc s'attendre à ce que LEGEND converge moins vite qu'ARTUR. D'autre part, nous avons montré que dans le cas convexe, ARTUR et LEGEND convergent vers la même solution, correspondant au minimum unique de l'énergie du MAP.

Pour vérifier cela expérimentalement, nous considérons les projections bruitées de l'objet Nice. La reconstruction est effectuée à l'aide de la fonction φ_{GR} en prenant en compte les 8 voisins de chaque pixel. Les paramètres sont fixés à 7 pour la hauteur de discontinuités et à 525 pour le coefficient de lissage. Le seuil d'arrêt entre itérations est choisi égal à $1e-6$ et à $1e-10$ entre étapes.

L'algorithme converge en 683 étapes soit un nombre total de 806 itérations. Comme pour ARTUR, chaque itération dure environ 10 secondes. Nous montrons figure 4-30 l'évolution de la solution au fil des étapes, la solution finale obtenue par LEGEND et celle obtenue par ARTUR. A titre de comparaison, avec les mêmes paramètres et dans les mêmes conditions, ARTUR converge en 78 étapes pour 136 itérations. On peut vérifier que le résultat obtenu par les deux algorithmes sont très proches. L'erreur relative entre les deux résultats est de l'ordre de $4e-6$. La différence est invisible à l'œil.

Nous présentons également (cf. figure 4-31) l'évolution de la variable auxiliaire dans la direction verticale, au fil des étapes. Plus exactement, pour comparer avec ARTUR, nous donnons la carte de $\varphi'(u)/2u$. On voit que la carte obtenue finalement est la même que celle estimée par ARTUR. D'autre part, on note que la carte obtenue est de moins bonne qualité que celle obtenue avec la fonction φ_{GM} (cf. § 4.1.1). Cela confirme le fait que la fonction φ_{GR} donne de moins bons résultats.

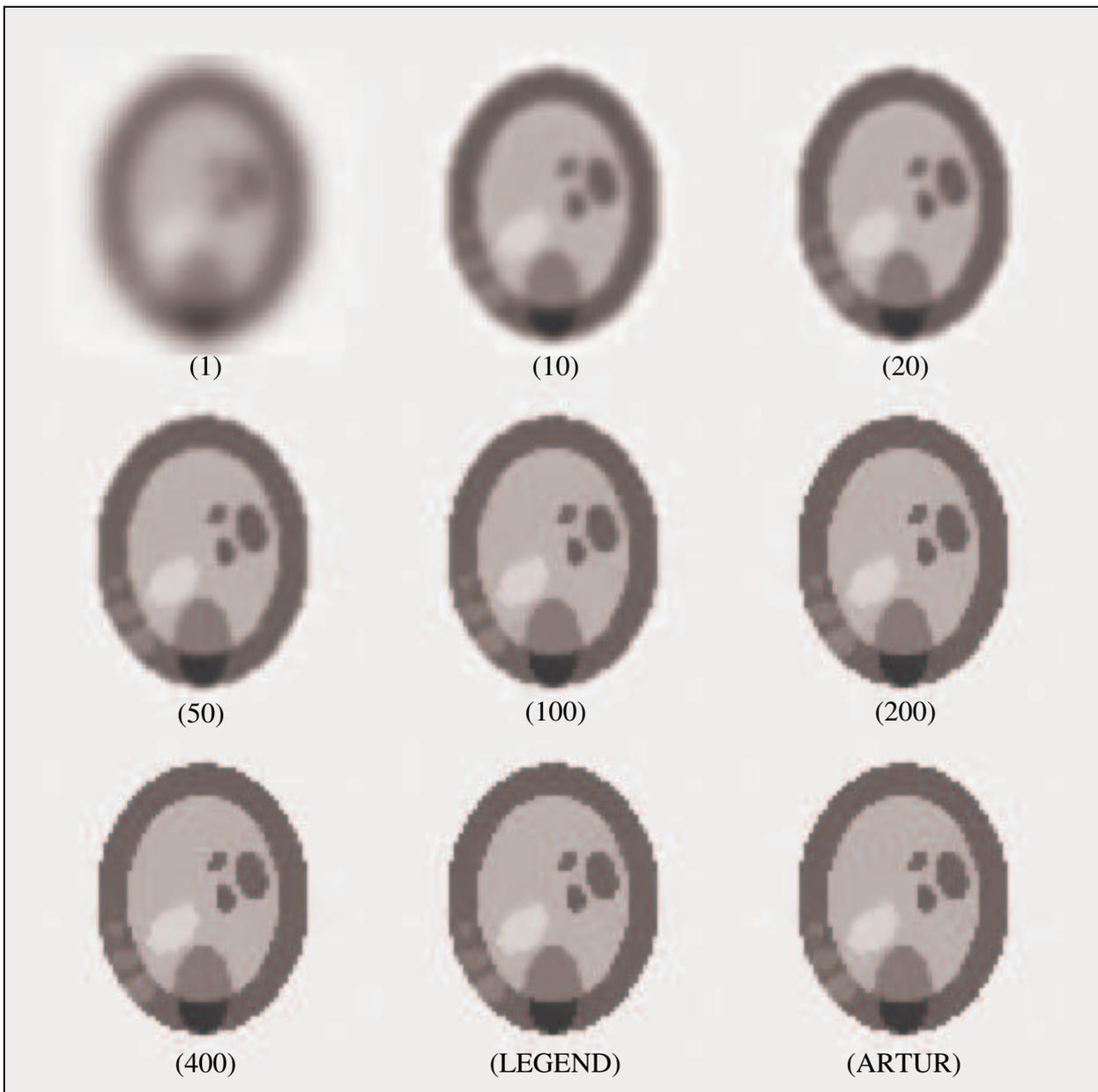


Figure 4-30: Reconstruction avec LEGEND: évolution de la solution au fil des étapes. En bas à droite: solutions finales obtenues avec LEGEND et ARTUR. On utilise le potentiel convexe φ_{GR}

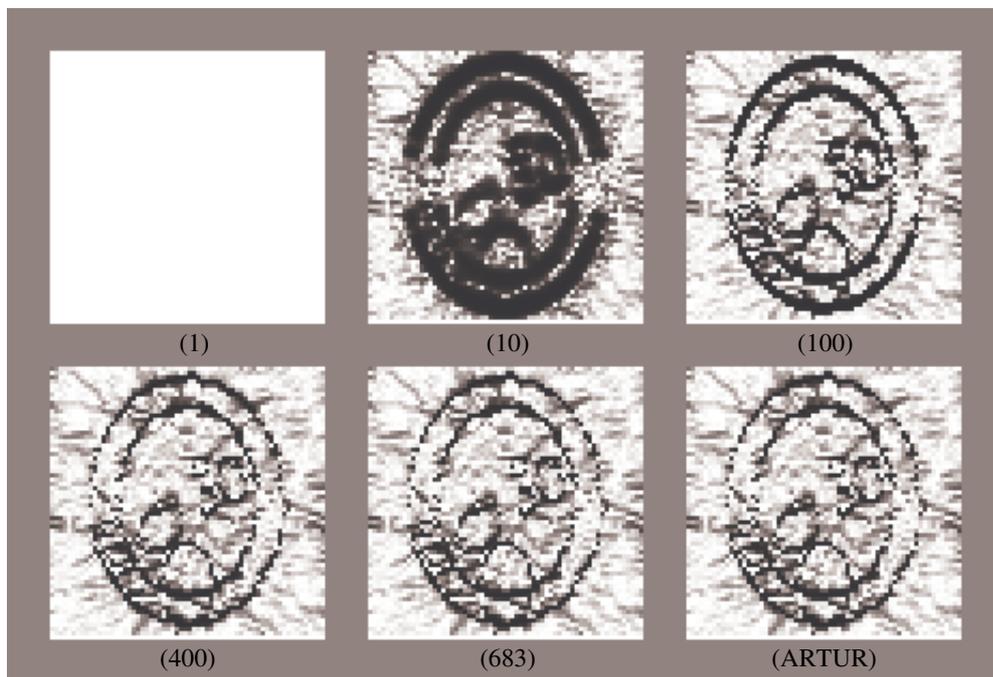


Figure 4-31: Reconstruction avec LEGEND: évolution de $\varphi'(u)/2u$ au fil des étapes (gradient pris dans la direction verticale). On utilise le potentiel convexe φ_{GR}

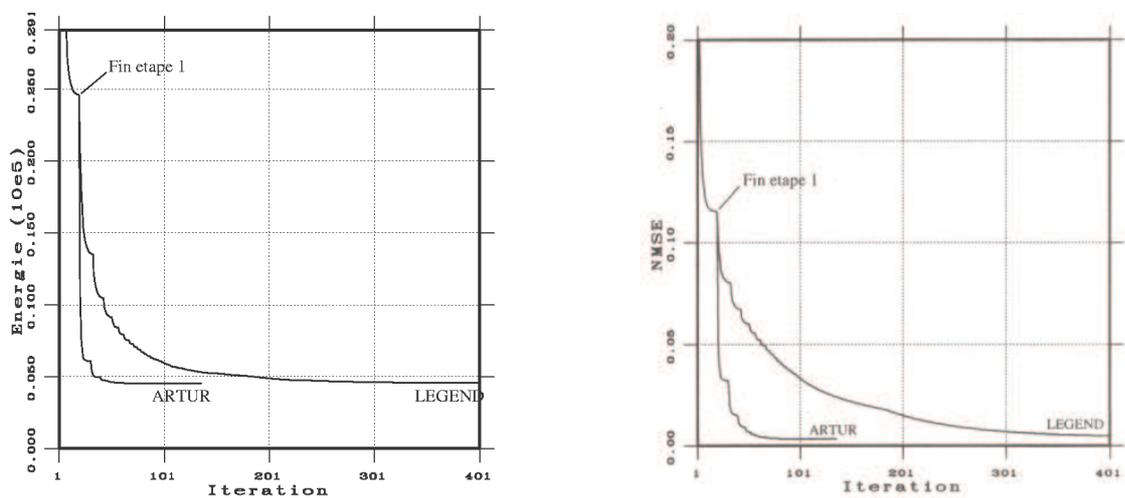


Figure 4-32: Evolution des énergies (à gauche) et de la NMSE par rapport à la solution exacte (à droite) au fil des itérations, pour ARTUR et LEGEND dans les mêmes conditions expérimentales.

Nous avons tracé figure 4-32 les courbes de variation de l'énergie du MAP et de la NMSE par rapport à la solution exacte. Ces courbes nous confirment qu'à la première étape (Tikhonov), ARTUR et LEGEND trouvent la même solution. Par la suite, l'évolution de Legend est plus lente, mais si on

laisse converger assez longtemps, ARTUR et LEGEND tendent vers le même point, conformément à ce que nous attendons.

4.2.2. Influence du seuil d'arrêt

Nous avons vu que LEGEND converge plus lentement qu'ARTUR. Dans cette expérience, nous étudions le nombre d'étapes nécessaires et la qualité de la reconstruction obtenue en fonction du seuil d'arrêt entre étapes. Le seuil d'arrêt entre itérations est fixé à $1e-6$. Les résultats obtenus sont montrés tableau 4-8.

Seuil d'arrêt entre étapes	$1e-6$	$1e-7$	$1e-8$	$1e-9$	$1e-10$
Nombre d'étapes ARTUR	13	23	38	55	78
Nombre d'étapes LEGEND	50	183	344	493	683
SNR ARTUR / ORIGINAL	22,05	22,39	22,53	22,58	22,59
SNR LEGEND / ORIGINAL	15,29	19,57	21,75	22,32	22,53
Erreur relative LEGEND / ARTUR	$9,02e-3$	$1,40e-3$	$1,51e-4$	$2,50e-5$	$4,33e-6$

Figure 4-8: Etude de l'influence du seuil d'arrêt entre étapes sur le nombre d'étapes effectuées et la qualité de reconstruction, pour les deux algorithmes ($\varphi = \varphi_{GR}$).

On peut faire les constatations suivantes:

- Le nombre d'étapes effectuées par LEGEND est toujours largement supérieur à celui effectué par ARTUR.
- ARTUR n'est pas très sensible au seuil d'arrêt entre étapes au niveau de la qualité de reconstruction. Entre $1e-6$ et $1e-10$, on gagne seulement 0,5 dB. Par contre, on multiplie par 6 le nombre d'étapes nécessaires.
- Pour ce qui est de la qualité de reconstruction, LEGEND est très sensible au seuil entre étapes. Il faut le laisser converger longtemps pour obtenir un résultat de qualité comparable à celle d'ARTUR. Il est encore plus sensible au niveau du nombre d'étapes effectuées, qui se trouve multiplié par un facteur 14 entre $1e-6$ et $1e-10$.
- Enfin, on constate que si on laisse converger assez longtemps les deux algorithmes, ceux-ci tendent vers la même solution, conformément à ce que nous avons prouvé au Chapitre III.

4.2.3. Utilisation d'un potentiel non convexe

Dans ce paragraphe, nous nous intéressons au cas où l'on utilise un potentiel non convexe pour la reconstruction. Ici, nous utiliserons la fonction de Geman et McClure, φ_{GM} . Les paramètres sont les mêmes que précédemment, et les seuils d'arrêt sont fixés à $1e-6$ entre itérations, et à $1e-10$ entre étapes. Nous présentons figure 4-33 les résultats obtenus avec ARTUR et LEGEND.



Figure 4-33: Reconstruction avec ARTUR (à gauche) et LEGEND (à droite) avec les mêmes paramètres, en utilisant le potentiel non-convexe φ_{GM}

La reconstruction par ARTUR a nécessité 70 étapes. Le SNR entre le résultat obtenu et l'objet Nice est de 24,89 dB. La reconstruction avec LEGEND a nécessité 681 étapes pour un SNR de 25,41 dB entre le résultat obtenu et l'objet Nice. Cette expérience nous montre que, pour un potentiel *non convexe*, on peut obtenir de très bons résultats avec LEGEND. Par contre, on n'est pas sûr de retrouver la même solution qu'avec ARTUR.

4.2.4. Factorisation de Cholesky

Tout comme ARTUR, l'algorithme LEGEND fournit de très bons résultats, même dans le cas non convexe. LEGEND est toutefois beaucoup moins rapide qu'ARTUR si on utilise à chaque étape un algorithme itératif pour résoudre le système $Af^{n+1} = c$.

Une version plus rapide de LEGEND peut être développée en tirant parti du fait que la matrice A ne change pas entre étapes, et en la factorisant une fois pour toutes. Comme A est symétrique et définie positive, on peut par exemple utiliser une factorisation de Cholesky. Chaque étape, se résume alors à la résolution de deux systèmes triangulaires, ce qui se fait directement. On peut ainsi espérer aller aussi

vite qu'avec ARTUR, voire plus vite, ce qui est finalement notre but: améliorer la vitesse de reconstruction.

Nous donnons ici un petit exemple, réalisé à partir des projections de l'objet Nice pour $N = 32$, avec un bruit de Poisson. On utilise la fonction de Green en prenant en compte les 8 voisins du pixel. Les paramètres sont $\delta = 9,5$ et $\lambda^2 = 100$. Les deux seuils d'arrêt sont fixés à $1e-8$. La factorisation de Cholesky prend 72 s. Nous présentons dans le tableau les nombres d'étapes et d'itérations, ainsi que le temps de calcul nécessaires à la reconstruction

	Nb. itérations	Nb. étapes	Temps CPU
ARTUR	156	31	85 s
LEGEND	784	280	437 s
LEGEND+CHOLESKY	-----	280	120 s

Tableau 4-9: Effet de la factorisation préalable de la matrice A sur le temps de calcul nécessaire à la reconstruction

On peut constater que le temps de calcul a quasiment été divisé par 4 en effectuant une factorisation de Cholesky. Cela rend l'algorithme LEGEND quasiment aussi rapide qu'ARTUR, mais pas plus. Cette méthode ne permet apparemment pas d'accélérer le temps de reconstruction. On ne gagne pas non plus au niveau de l'encombrement-mémoire. La matrice factorisée prend autant de place que la matrice de départ. Au Chapitre V, nous proposons d'utiliser la transformée en ondelettes, pour accélérer la reconstruction et diminuer notablement la place-mémoire nécessaire.

4.3. Autres algorithmes

4.3.1. Moise

Au Chapitre III, nous avons présenté l'algorithme MOISE, analogue d'ARTUR utilisant un modèle de vraisemblance Poissonien. Nous donnons ici un des premiers résultats expérimentaux obtenus avec cet algorithme, à partir des projections bruitées de l'objet Nice. Ce résultat a été obtenu avec le fonction de Geman et McClure, φ_{GM} . Il est présenté figure 4-34.

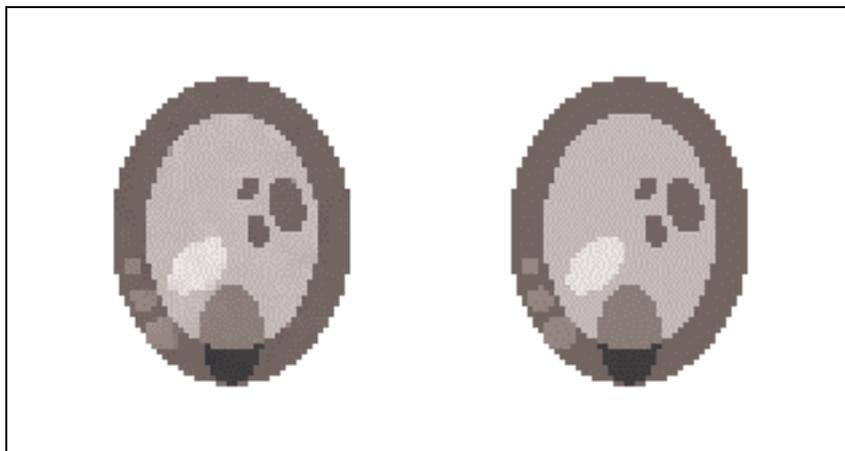


Figure 4-34: Résultat de reconstruction obtenu avec MOISE 4 voisins (à gauche) et objet Nice (à droite)

On peut constater que l'image obtenue est de très bonne qualité. Nous n'avons pourtant utilisé qu'un modèle à 4 voisins. Le SNR entre l'image calculée et l'image originale est de 26,25 dB. Ce résultat est légèrement meilleur que celui obtenu avec ARTUR en 8 voisins. Nous avons donc bon espoir que cet algorithme donne de très bons résultats. Le revers de la médaille est le temps de calcul nécessaire, puisqu'il a fallu effectuer 16 étapes, représentant au total 1481 itérations, pour obtenir l'image présentée. Chacune des itérations nécessite environ 20 secondes, soit le double du temps nécessaire à une itération d'ARTUR.

Cet algorithme pourrait toutefois s'avérer utile à très faible taux de comptage. Dans ce cas, l'approximation Gaussienne de la loi de Poisson, utilisée par ARTUR, n'est plus aussi bien justifiée qu'aux taux de comptage habituellement utilisés. Il serait intéressant de réaliser des tests afin de savoir si le surcoût de calcul imposé par MOISE se justifie ou non par une meilleure qualité des images reconstruites.

4.3.2. Artur explicite

Au Chapitre III, nous avons également évoqué la possibilité d'un analogue explicite d'ARTUR. Cet algorithme a déjà été utilisé dans des cas particuliers (ex: dans le MFA, à température fixée [GEIG90])

Nous présentons ici un premier résultat obtenu avec l'algorithme ARTUR explicite sur un objet simple. Nous prenons pour cette expérience une image 16x16, formée d'un carré centré, de valeur de pixel 128 et de taille 8x8, sur fond à 0. Nous générons les projections de cet objet en le multipliant par la matrice de Radon, puis nous les bruitons par du bruit de Poisson.

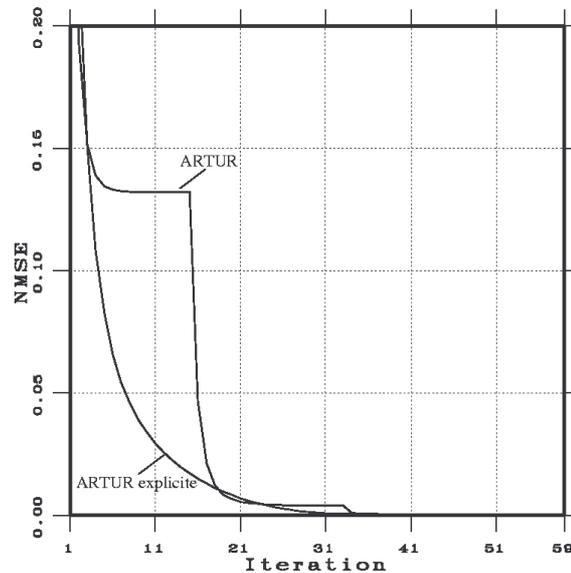


Figure 4-35: Un essai d’algorithme ARTUR explicite. Courbe de la NMSE par rapport à la solution exacte en fonction des itérations pour ARTUR et ARTUR explicite

Nous utilisons ces projections pour une reconstruction avec ARTUR et ARTUR explicite, avec le même jeu de paramètres, et en prenant comme fonction de potentiel la fonction φ_{HL} . La valeur du coefficient de relaxation ∂t intervenant dans l’algorithme ARTUR explicite (cf. Chapitre III) est fixée de façon expérimentale à $5e-3$: des valeurs inférieures donnent une convergence plus lente, et si on prend une valeur plus forte, l’algorithme ne converge pas.

L’algorithme ARTUR converge en 7 étapes, soit 59 itérations. L’algorithme ARTUR explicite converge en 53 itérations. Nous traçons figure 4-35 la courbe de la NMSE par rapport à la solution exacte en fonction de l’itération, pour chacun des deux algorithmes. Dans les deux cas, le SNR entre l’image reconstruite et l’image de départ est de 36,4 dB. L’erreur relative entre la solution obtenue avec ARTUR explicite et la solution obtenue par ARTUR est de $1,8e-7$.

Il semble donc que dans cet exemple, les deux algorithmes donnent le même résultat. L’étude théorique et expérimentale des algorithmes explicites nous paraît donc constituer une bonne perspective de travail. En particulier, il serait intéressant d’étudier les propriétés de convergence de ces algorithmes dans le cadre théorique de la régularisation semi-quadratique.

6. Conclusion

Dans ce chapitre, nous avons présenté les résultats obtenus avec ARTUR et LEGEND en reconstruction tomographique. Ces exemples nous ont permis de mieux comprendre le fonctionnement des algorithmes et de vérifier les résultats annoncés au Chapitre III.

Dans le cas où on utilise un potentiel convexe, la convergence vers la solution du MAP est assurée. On obtient un résultat formé de zones homogènes et séparées par des bords francs. Toutefois, la qualité des contours est moins bonne qu'en utilisant un potentiel non convexe. Dans ce dernier cas, la convergence n'est pas prouvée mathématiquement, mais on l'observe toujours en pratique, et les solutions obtenues sont toujours de bonne qualité.

L'algorithme LEGEND converge plus lentement qu'ARTUR. Il nécessite beaucoup plus d'étapes pour trouver une solution de même qualité. Si on utilise un algorithme itératif pour résoudre le système d'équations normales en f à chaque étape, le temps de reconstruction est beaucoup plus important avec LEGEND. Par contre, on peut utiliser le fait que seul le second membre de ces équations varie, et que la matrice mise en jeu n'est pas modifiée au fil des étapes. Une factorisation préalable de la matrice par l'algorithme de Cholesky permet un gain de temps appréciable à chaque étape. Par contre, cette technique ne réduit pas la taille mémoire nécessaire. Nous proposons au Chapitre V une autre solution, basée sur l'utilisation de la transformée en ondelettes, permettant de réduire à la fois le temps de calcul et la place mémoire nécessaires à la reconstruction.

Enfin, nous avons pu constater sur ces quelques exemples que nos algorithmes sont bien adaptés au problème de la reconstruction tomographique, même s'ils sont basés sur une approximation Gaussienne de la vraisemblance. Cela reste vrai même pour des taux de comptage plus faibles [KOUL94]. Une adaptation d'ARTUR au cas de la vraisemblance Poissonienne est de toutes façons possible. C'est l'algorithme MOISE, qui semble pouvoir donner des résultats encore meilleurs qu'ARTUR, mais pour une charge de calcul plus importante.

Nous avons présenté les résultats obtenus par nos algorithmes dans le cas de la tomographie SPECT 2D, sans prise en compte des phénomènes physiques intervenant lors de l'acquisition des projections. Notons qu'ARTUR est également utilisé, avec succès, par P.M. Koulibaly dans le cadre de l'étude des corrections physiques [KOUL95].

Une autre perspective de travail importante est l'adaptation des algorithmes au cas 3D. Une version 3D en géométrie parallèle a été réalisée par V. Amar [AMAR94]. L'extension au cas 3D en géométrie conique est actuellement étudiée par I. Laurette [LAUR94].

Des versions de l'algorithme ARTUR ont aussi été réalisées pour d'autres applications, telles que la restauration d'image et l'estimation de mouvements [BLAN92] [ROUC94]. Les résultats obtenus sont dans tous les cas très satisfaisants. Il semble donc que les algorithmes déterministes basés sur une régularisation semi-quadratique soient particulièrement bien adaptés aux problèmes inverses mal-posés que l'on rencontre en traitement d'image

Chapitre V

Résolution dans le domaine de la transformée en ondelettes 2D

1. Introduction

Les méthodes algébriques, qui consistent à considérer le problème de reconstruction sous son aspect discret, permettent la prise en compte des phénomènes physiques intervenant lors de l'acquisition des données et offrent un moyen simple de les corriger. L'inconvénient majeur des méthodes algébriques est qu'elles mettent en jeu des matrices de très grandes dimensions. Cela a une incidence sur l'encombrement-mémoire des algorithmes et sur le temps de reconstruction. Ceux-ci sont déjà importants dans le cas bidimensionnel, même pour des images de petite taille. Ils deviennent rapidement rédhibitoires si on envisage une extension à des dimensions d'images plus grandes. Or, l'amélioration de la résolution des images et la reconstruction de volumes impliquent une augmentation de la taille des données. Pour favoriser l'utilisation de tels algorithmes en routine médicale il est nécessaire de les rendre moins coûteux en ressources informatiques. Cela passe par une réduction importante du volume d'information mis en jeu.

Dans le cas particulier où l'opérateur impliqué est invariant par translation, des méthodes basées sur la transformée de Fourier peuvent être utilisées. Dans le cas général, une possibilité, proposée dans [BEYL91], consiste à effectuer une transformée en ondelettes [MEYE] de l'opérateur matriciel impliqué. Ceci permet de "creuser" la matrice en y introduisant un nombre important de zéros. On peut ainsi définir des algorithmes rapides, tout en gardant le contrôle sur la précision du résultat obtenu. Nous proposons ici, à titre de perspective, d'adapter nos méthodes de reconstruction tomographique à l'utilisation de la transformée en ondelettes biorthogonales [ANTO92].

Dans un premier temps, nous donnons quelques rappels sur la mise en œuvre de la transformée en ondelettes biorthogonale, et définissons comme dans [BEYL91] la transformée d'une matrice. Dans un second temps, nous présentons une autre forme de transformée, mieux adaptée au traitement des images. Nous donnons ensuite un exemple d'utilisation de ces transformées dans le cas de la régularisation de Tikhonov. Enfin, nous étudions l'adaptation d'ARTUR et de LEGEND à la résolution dans le domaine de la transformée en ondelettes.

2. Présentation des transformées en ondelettes BCR et 2D

2.2. Transformée en ondelettes multirésolution

La transformée en ondelettes multirésolution se met en œuvre à l'aide de bancs de filtres numériques [MALL89]. On associe à la fonction d'échelle un filtre passe-bas h , et à l'ondelette un filtre passe-haut g . L'approximation du signal à la résolution d'indice m est déduite de l'approximation à la résolution d'indice $m - 1$:

$$s_m(n) = \sum_k h(2n - k) s_{m-1}(k) \quad (5-1)$$

Les coefficients d'ondelettes à la résolution d'indice m sont donnés par:

$$c_m(n) = \sum_k g(2n - k) s_{m-1}(k) \quad (5-2)$$

Remarque: Il est souhaitable d'utiliser des filtres courts (pour des calculs rapides) et symétriques (pour ne pas dégrader la phase du signal) tout en conservant la propriété de reconstruction exacte. Il est impossible de réunir simultanément ces conditions avec des bases orthogonales [ANTO92]. C'est pourquoi on relâche la condition d'orthogonalité et on utilise des ondelettes bi-orthogonales [COHE90]. Cela revient à effectuer la synthèse à l'aide de filtres différents de ceux utilisés à l'analyse. On a alors:

$$s_{m-1}(k) = \sum_n \tilde{h}(2n - k) s_m(n) + \sum_n \tilde{g}(2n - k) c_m(n) \quad (5-3)$$

Les filtres sont reliés entre eux par les relations suivantes. La biorthogonalité s'exprime par:

$$\sum_n h(n) \tilde{h}(2k + n) = \sum_n g(n) \tilde{g}(2k + n) = \delta_{k,0} \quad (5-4)$$

D'autre part, on a les relations:

$$\begin{aligned} \tilde{g}(n) &= (-1)^n h(-n+1) \\ g(n) &= (-1)^n \tilde{h}(-n+1) \end{aligned} \quad (5-5)$$

D'après l'expression (5-1), on voit que l'approximation du signal à la résolution d'indice m est obtenue en convoluant s_{m-1} avec le filtre h et en ne gardant que les termes d'indice pair. Dans l'expression (5-2), le filtre g est considéré centré en 1. Les coefficients d'ondelette sont donc obtenus en convoluant le signal s_{m-1} par g et en ne gardant que les termes d'indice impair [ANTO92].

La synthèse du signal à la résolution d'indice $m - 1$ est obtenue à l'aide de sur-échantillonnages et de convolutions. Nous représentons sur la figure 5-1 le banc de filtre correspondant à l'analyse-synthèse d'un signal 1D sur un niveau de résolution.

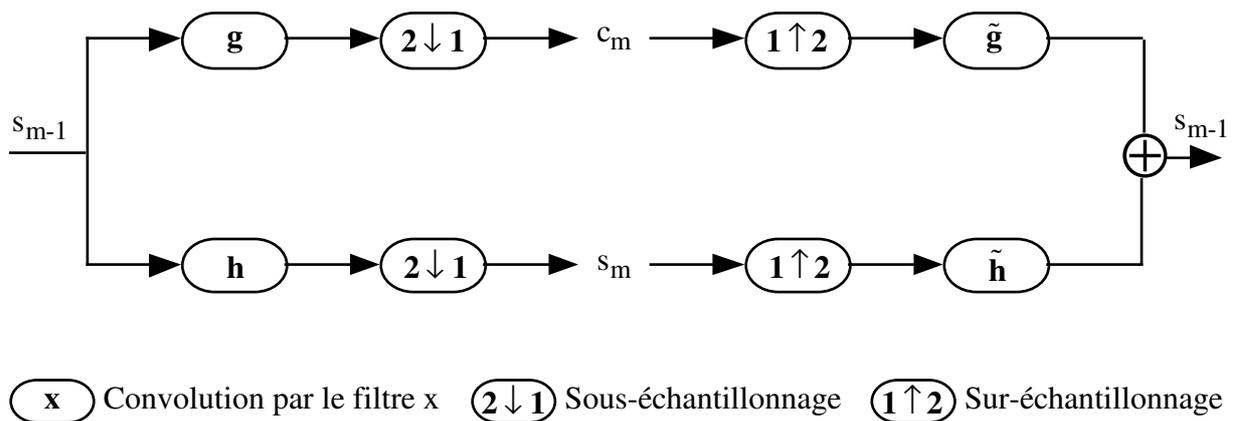


Figure 5-1: Analyse-synthèse d'un signal 1D sur un niveau

Du fait de la dualité des filtres, il est également possible d'effectuer l'analyse en utilisant (\tilde{g}, \tilde{h}) et la synthèse avec (g, h) .

2.3. Représentation matricielle: transformée BCR

2.3.1. Transformée d'un vecteur

- L'analyse en ondelettes se met en œuvre par des convolutions discrètes. Si l'on range dans le même vecteur s_1 et c_1 , on peut réécrire les équations (5-1) et (5-2) sous forme matricielle.

On a alors ¹:

$$\begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = TO \cdot s_0 \quad (5-6)$$

où TO représente l'opérateur de transformation en ondelettes 1D et s'écrit:

$$TO = \begin{bmatrix} G \\ H \end{bmatrix} \quad (5-7)$$

Les matrices G et H font $N^2/2$ lignes sur N^2 colonnes. Elles représentent les opérations de sous-échantillonnage et de convolution par les filtres g et h . Des précautions particulières doivent être prises en ce qui concerne le traitement des bords du vecteur, et on écrit:

$$TO = \begin{bmatrix} G \\ H \end{bmatrix} = \begin{bmatrix} G_0 \\ H_0 \end{bmatrix} + \begin{bmatrix} G_B \\ H_B \end{bmatrix} = TO_0 + TO_B \quad (5-8)$$

Nous ne rentrerons pas dans les détails au sujet du traitement des bords (le lecteur pourra se référer à [LOBE93] où ces problèmes sont étudiés de façon plus complète). Nous utilisons une symétrisation du signal aux bords.

- De la même façon que pour l'analyse, l'opération de synthèse peut s'écrire sous forme matricielle:

$$s_0 = \tilde{TO}^{-1} \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} \quad (5-9)$$

Du fait du traitement particuliers aux bords, on a:

$$\tilde{TO}^{-1} = \begin{bmatrix} \tilde{G}_0^t & \tilde{H}_0^t \end{bmatrix} + \begin{bmatrix} \tilde{G}_{B^t} & \tilde{H}_{B^t} \end{bmatrix} = \tilde{TO}_0^t + \tilde{TO}_{B^t} \quad (5-10)$$

avec $\tilde{TO}_{B^t} \neq \left(\tilde{TO}_B\right)^t$. Les matrices \tilde{G}_0 et \tilde{H}_0 représentent les opérations de sous-échantillonnage et de convolution par les filtres \tilde{g} et \tilde{h} (en dehors des bords).

- La transformée en ondelettes étant à reconstruction exacte, on a:

¹ Pour simplifier, on considère dans un premier temps un seul niveau de résolution.

$$\tilde{TO}^{-1} \cdot TO = TO^{-1} \cdot \tilde{TO} = I \quad (5-11)$$

Les filtres étant biorthogonaux, on a:

$$TO \cdot \tilde{TO}^{-1} = \tilde{TO} \cdot TO^{-1} = I \quad (5-12)$$

• Dans le cas où on considère n niveaux de résolution, le vecteur transformé s'écrit sous la forme suivante:

$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \\ s_n \end{bmatrix} \quad (5-13)$$

où c_i est un vecteur de dimension $(N^2/2^i) \times 1$ contenant les coefficients d'ondelette à la résolution d'indice i .

L'analyse et la synthèse d'un vecteur sur n niveaux de résolution peuvent également s'écrire sous forme matricielle.

2.3.2. Transformée d'une matrice

• Considérons maintenant un système linéaire de la forme:

$$Sf = t \quad (5-14)$$

D'après ce que nous venons de voir, on a:

$$TO \cdot (Sf) = TO \cdot t \quad (5-15)$$

En utilisant la propriété (5-11), on peut écrire:

$$\left(TO \cdot S \cdot \tilde{TO}^{-1} \right) \cdot TO \cdot f = TO \cdot t \quad (5-16)$$

On voit donc apparaître une transformée agissant sur une matrice 2D. Cette transformée fait correspondre à la matrice S la matrice S^{BCR} définie par:

$$S^{BCR} = TO \cdot S \cdot \tilde{TO}^{-1} \quad (5-17)$$

L'opérateur inverse s'écrit facilement:

$$S = \tilde{TO}^{-1} \cdot S^{BCR} \cdot TO \quad (5-18)$$

Remarque: Aux problèmes de bords près, on a $S^{BCR} \approx TO_0 \cdot S \cdot \tilde{TO}_0^t$, ce qui revient à appliquer l'opérateur 1D d'analyse à chaque vecteur-colonne de S en utilisant les filtres g et h , puis à l'appliquer à chaque vecteur-ligne de la matrice ainsi obtenue avec les filtres \tilde{g} et \tilde{h} .

D'après la remarque précédente, et l'expression (5-13), on visualise facilement la structure de la transformée en ondelettes d'une matrice sur n niveaux de résolution (voir exemple suivant).

- Ayant été introduite par Beylkin, Coifman et Rokhlin [BEYL91], cette transformée est appelée transformée BCR. Cette transformée a été appliquée dans [BEYL91] à des opérateurs de Calderon-Zygmund (qui décroissent en $1/d$, où d est la distance à la diagonale) et à des opérateurs pseudo-différentiels. Dans ce cas, on a des résultats théoriques sur le comportement de la transformée en fonction des propriétés de l'ondelette [BEYL91].

A titre d'exemple, nous donnons figure 5-2 la transformée BCR de la matrice B définie par:

$$\begin{cases} B_{i,j} = 1 & \text{si } i = j \\ B_{i,j} = \frac{1}{|i-j|} & \text{sinon} \end{cases} \quad (5-19)$$

Pour la visualisation, nous avons binarisé la matrice transformée en mettant à 0 les entrées de la matrice transformée dont la valeur était inférieure à 4‰ du maximum (ces entrées apparaissent en blanc) et à 1 les valeurs supérieures à ce seuil (en noir). On constate que la matrice est très creuse: elle contient 92,8% de zéros.

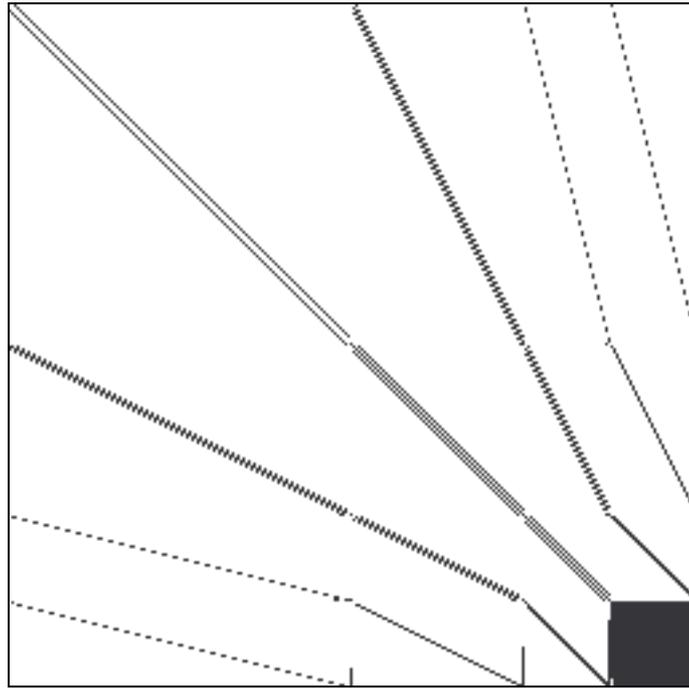


Figure 5-2: Transformée BCR de la matrice B sur 3 niveaux de résolution. En noir: valeurs supérieures à 4‰ du maximum, en blanc: valeurs inférieures à ce seuil (représentant 92,8% de la matrice)

2.4. Application aux images: transformée “2D”

2.4.1. Transformée d'un vecteur

L'inconvénient majeur de la transformée BCR est qu'elle ne tient pas compte de la cohérence spatiale qui existe entre les lignes de l'image. Il est vrai que la plupart du temps, l'image est manipulée sous forme de vecteur dans la résolution de systèmes linéaires (voir paragraphe suivant). Toutefois, rien n'empêche de la considérer comme un champ bidimensionnel F de taille $N \times N$ lorsqu'on opère sa transformée en ondelettes. Nous utilisons une transformée 2D dyadique (cf. [MALL89]). Nous rangeons les coefficients dans l'image transformée F^{2D} de façon à écrire:

$$F^{2D} = TO.F.TO^t \quad (5-20)$$

où la matrice TO (ici de taille $N \times N$) est définie comme au paragraphe précédent. La transformée inverse est définie par:

$$F = \tilde{TO}^{-1} \cdot F^{2D} \cdot \left(\tilde{TO}^{-1} \right)^t \quad (5-21)$$

La propriété (5-11) nous assure que la reconstruction est exacte.

En utilisant le produit de Kronecker, \otimes , il est possible de se ramener à une représentation vectorielle de l'image. Les équations (5-20) et (5-21) peuvent se mettre sous la forme suivante:

$$f^{2D} = [TO \otimes TO] \cdot f = TO \cdot f \quad (5-22)$$

$$f^{2D} = \left[\tilde{TO}^{-1} \otimes \tilde{TO}^{-1} \right] \cdot f = \tilde{TO}^{-1} \cdot f \quad (5-23)$$

et la reconstruction exacte s'exprime par:

$$\tilde{TO}^{-1} \cdot TO = I \quad (5-24)$$

Ainsi, bien qu'agissant sur l'image rangée sous forme vectorielle, les opérateurs TO et \tilde{TO}^{-1} mettent en œuvre une transformée dyadique 2D et conservent donc toute la cohérence spatiale de l'image.

2.4.2. Transformée d'une matrice

La définition de l'opérateur agissant sur les matrices se fait de la même façon que dans le cas de la transformée BCR. On obtient la définition suivante pour la transformée directe:

$$S^{2D} = TO \cdot S \cdot \tilde{TO}^{-1} \quad (5-25)$$

et pour la transformée inverse:

$$S = \tilde{TO}^{-1} \cdot S^{2D} \cdot TO \quad (5-26)$$

Comme dans le cas de la transformée BCR, on peut décomposer la transformée "2D" en une transformée agissant sur les vecteurs-colonnes de S, puis sur les vecteur-lignes du résultat. Cette transformée a été mise en œuvre par Pierre Lobel au cours de son stage de DEA (cf. [LOBE93]. En pratique, on ne forme jamais TO ni \tilde{TO}^{-1} , mais on utilise une transformée dyadique, en réarrangeant chaque vecteur-colonne (ou ligne) de la matrice sous forme d'image.

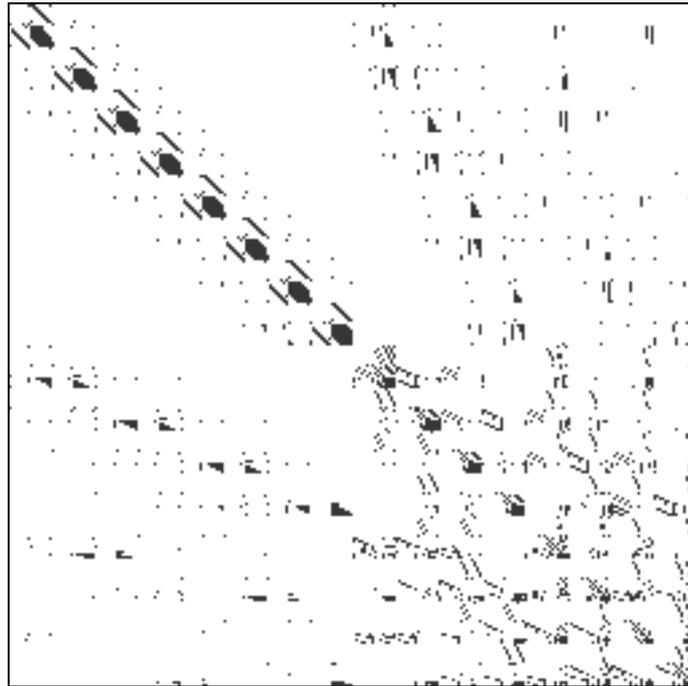


Figure 5-3: Transformée “2D” de la matrice B sur 3 niveaux de résolution. En noir: valeurs supérieures à 4‰ du maximum, en blanc: valeurs inférieures à ce seuil (représentant 90,7% de la matrice)

A titre d'exemple, nous donnons Figure 5-3 la transformée de la matrice B définie au paragraphe précédent. Pour la visualisation, nous avons binarisé la matrice transformée en utilisant le même seuil que dans le cas de la transformée BCR. On peut voir que la forme “en doigts” est remplacée par une structure plus compliquée. On peut encore constater que la matrice est très creuse.

3. Résolution de systèmes linéaires de grande taille dans le domaine de la TO

3.1. Introduction

Comme nous l'avons remarqué précédemment, la transformée en ondelettes permet de rendre très creux certains opérateurs matriciels. Dans [BEYL91], cette technique a été appliquée à des opérateurs de Calderon-Zygmund et à des opérateurs pseudo-différentiels. En effectuant un simple seuillage, il est possible d'introduire un nombre très important de zéros dans la matrice transformée.

On peut se servir de cette propriété pour faciliter la résolution des systèmes linéaires de très grande taille intervenant en reconstruction d'image (voir par exemple [BATH93A] [BATH93B])

[PENT91] [WAKU93] [BLAN92]). En effet, la plupart des opérateurs matriciels utilisés en traitement d'image ont une diagonale dominante, et se "creusent" de façon très importante dans le domaine de la transformée en ondelettes.

Résoudre le système linéaire en question dans le domaine de la transformée en ondelettes à l'aide d'une algorithmique dédiée aux matrices creuses permet un gain important en place mémoire et en temps de calcul.

En tomographie, on pourrait ainsi envisager des dimensions d'images plus grandes, afin d'obtenir une meilleure résolution, et donc de faciliter le diagnostic. On pourrait également envisager d'appliquer ces techniques à la reconstruction 3D, excessivement gourmande en ressources informatiques [LAUR94] [AMAR94]. Nous donnons ici les premiers résultats obtenus en reconstruction tomographique 2D.

3.2. Exemple: cas de la régularisation de Tikhonov

Dans un premier temps, nous nous sommes intéressés [BLAN92], [LOBE93], [BLAN94] au cas simple de la reconstruction avec régularisation de type Tikhonov. Dans ce cas, l'énergie du MAP s'écrit:

$$J(f) = \|p - Rf\|^2 + \lambda^2 \|D_x f\|^2 + \lambda^2 \|D_y f\|^2 \quad (5-27)$$

Cette fonctionnelle est convexe, et n'admet donc qu'un seul minimum. La solution unique du problème du MAP, \hat{f} , est calculée en résolvant les équations normales associées:

$$(R^t R - \lambda^2 \Delta) \hat{f} = R^t p \quad (5-28)$$

qui sont de la forme $Sf = t$. La matrice S a ses valeurs les plus importantes près de la diagonale, comme on peut le constater sur la figure 5-4, avec $N = 16$ et $\lambda^2 = 5$.

Nous présentons sur les figures 5-5 et 5-6 les transformée en ondelettes BCR et "2D" de la matrice S . Pour la décomposition, on a employé un jeu de filtres 5-7 (cf. [ANTO91]). Nous avons binarisé les images après seuillage à 4%. Les valeurs mises à zéro apparaissent en blanc, les autres en noir. On peut constater que la transformée "2D" de S est beaucoup plus creuse (91,2% de zéros introduits dans la matrice) que sa transformée BCR (62,3% de zéros).

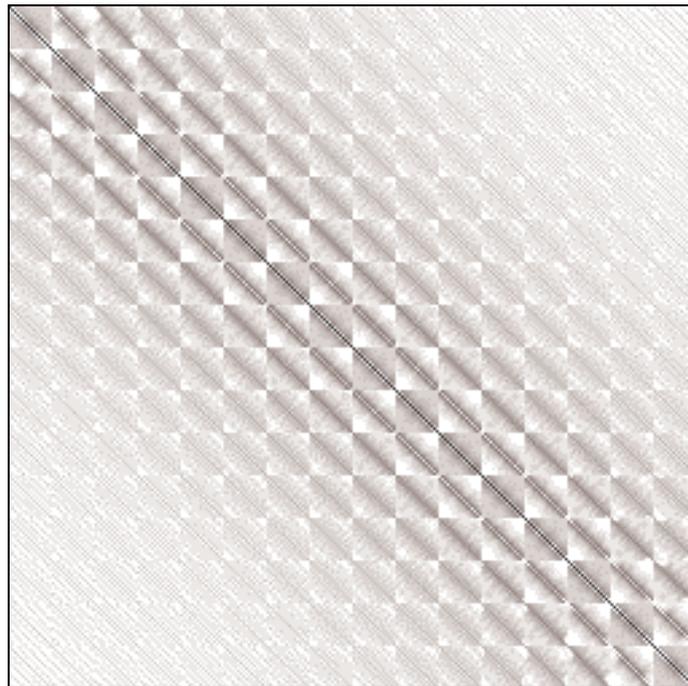


Figure 5-4: Matrice $R^t R - 5\Delta$ ($N = 16$). Valeurs faibles en blanc, valeurs importantes en noir

Nous présentons également, pour les deux transformées une étude de l'effet du seuillage. Pour cela, nous considérons le cas où $N = 32$. On prend $\lambda^2 = 10$. La transformée est effectuée avec un filtre 9-7. Nous nous intéressons aux effets du seuillage:

- sur le pourcentage de zéros introduits dans la matrice (cf. figure 5-7),
- sur la vitesse de reconstruction (cf. figure 5-8). Nous mesurons le temps mis par un algorithme de Gauss-Seidel "creux" utilisé pour la reconstruction. Le temps de calcul est normalisé par le temps mis par l'algorithme de Gauss-Seidel utilisant les matrices pleines.

Nous nous intéressons également à la qualité de la reconstruction en fonction du nombre de zéros introduits dans la matrice (cf. figure 5-9). La qualité est mesurée par le rapport signal-à-bruit (SNR) en variances.

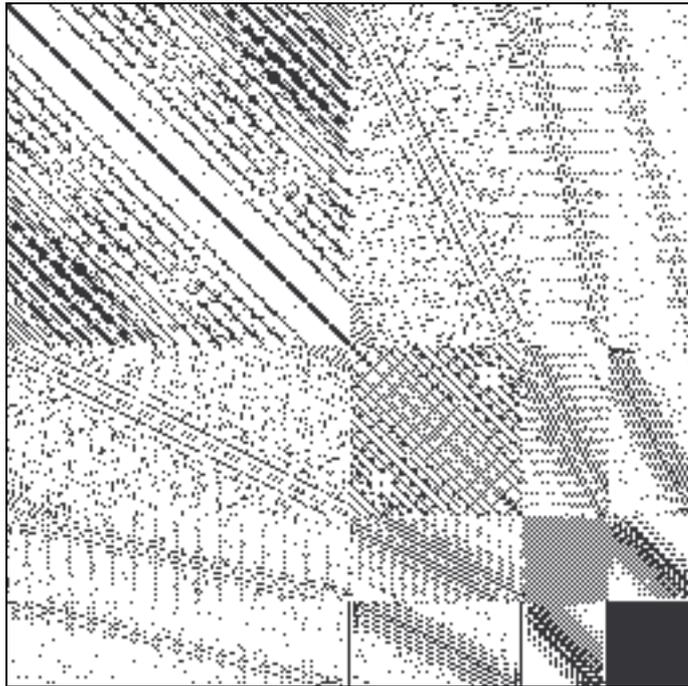


Figure 5-5: Transformée BCR de la matrice $R^t R - 5\Delta$, seuillée à 4‰ (62,3% de zéros introduits dans la matrice). L'image est binarisée pour l'affichage

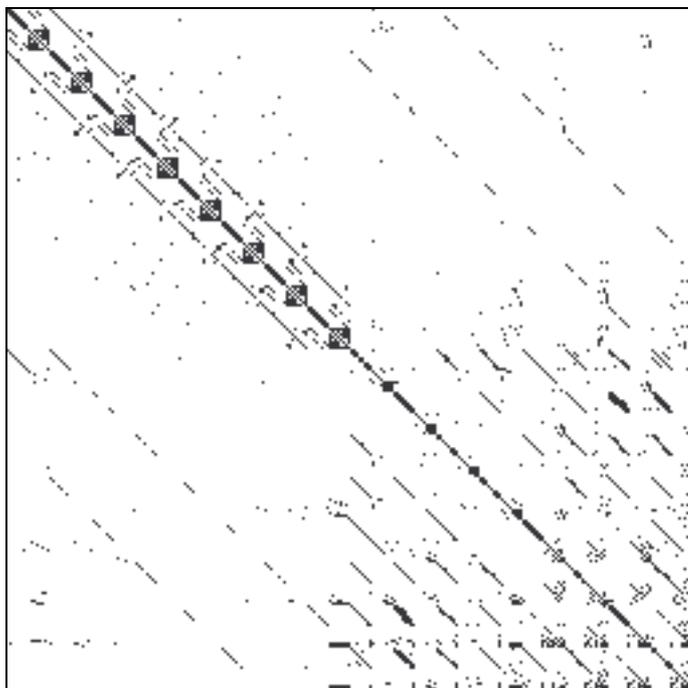


Figure 5-6: Transformée "2D" de la matrice $R^t R - 5\Delta$, seuillée à 4‰ (91,2% de zéros introduits dans la matrice). L'image est binarisée pour l'affichage

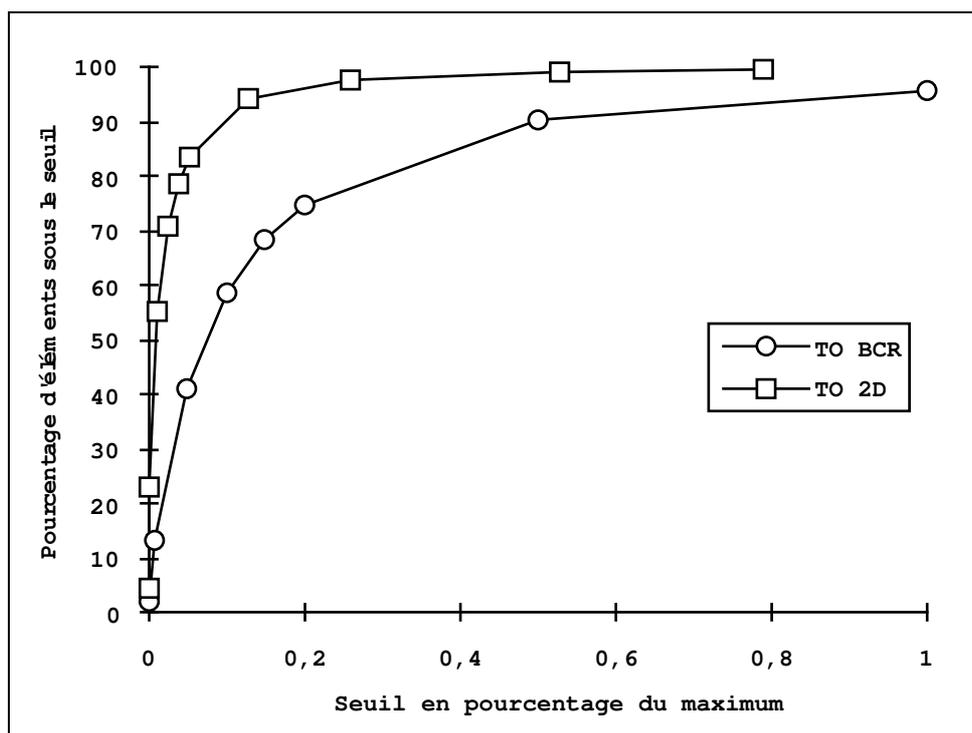


Figure 5-7: Nombre de zéros introduits dans la matrice en fonction du seuil

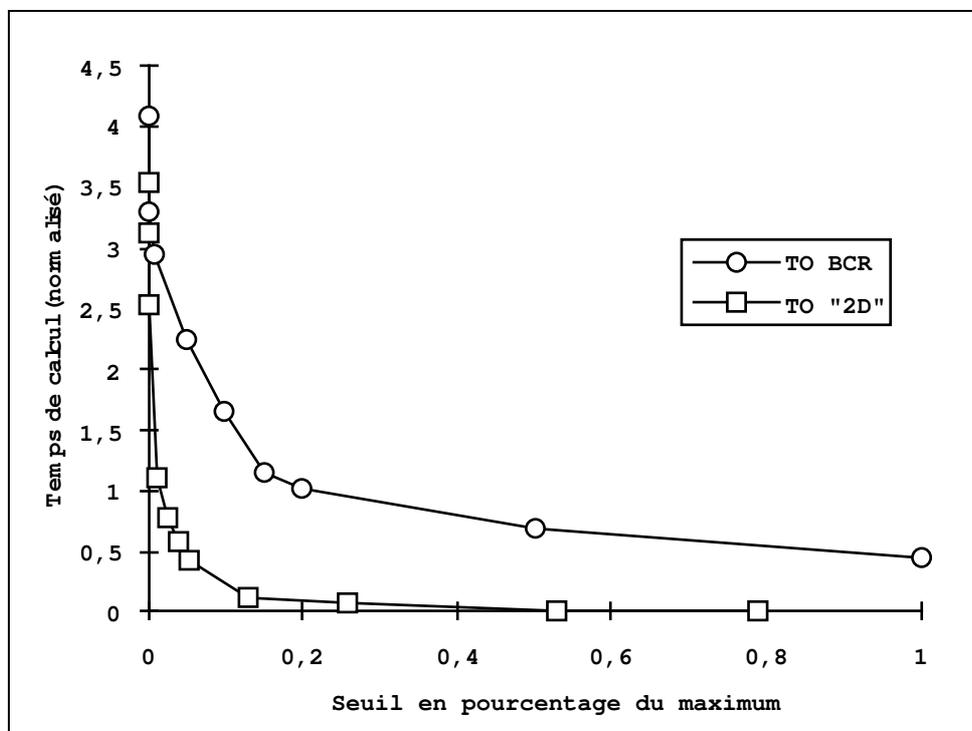


Figure 5-8: Temps de reconstruction en fonction du nombre de zéros dans la matrice

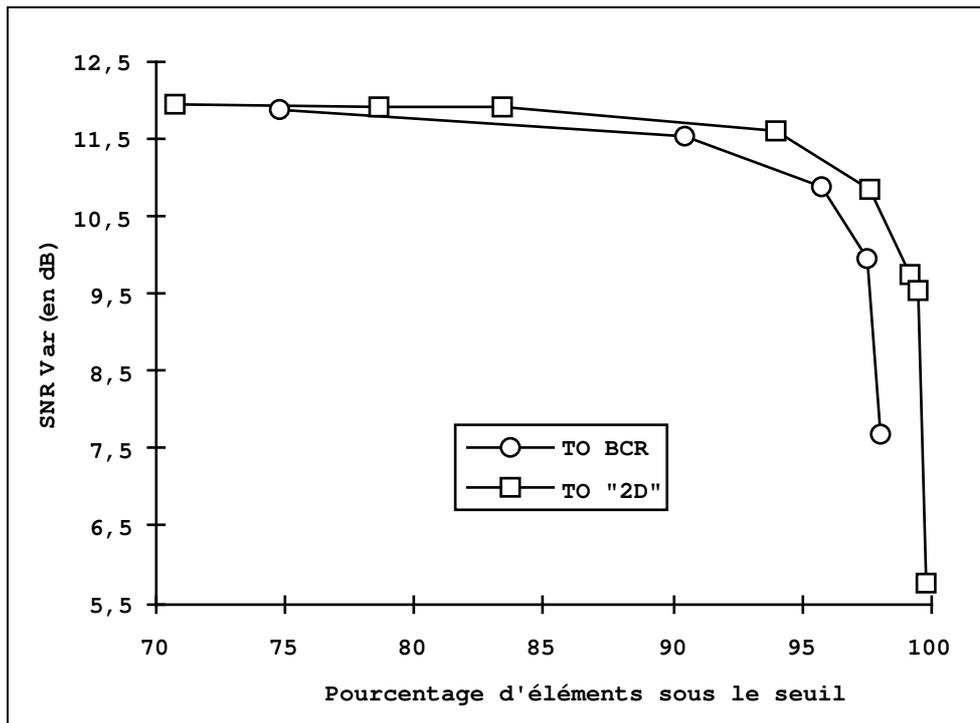


Figure 5-9: Qualité de la reconstruction en fonction du nombre de zéros dans la matrice

Au vu des résultats obtenus, on peut remarquer que les tendances observées sont les mêmes pour les deux transformées:

- Le nombre de zéros introduits dans la matrice augmente de façon très rapide en fonction du seuil utilisé, avant d'atteindre un plafond.
- Le temps de calcul diminue très rapidement en fonction du seuil. Ce résultat est logique. Il prouve que l'algorithme est bien adapté au problème. Lorsqu'on ne seuille pas, il est beaucoup plus lent que l'algorithme Gauss-Seidel standard. Par contre, la vitesse de reconstruction augmente très vite que l'on introduit des zéros dans la matrice.
- On peut introduire un nombre important de zéros dans la matrice sans trop nuire à la qualité de l'image reconstruite.

Ainsi, il apparaît que l'on peut gagner énormément au niveau du temps de calcul (un facteur 10 environ), sans que cela nuise à la qualité de l'image reconstruite.

Les résultats obtenus par la transformée "2D" sont toujours meilleurs que ceux obtenus par la transformée BCR. Cela confirme que la transformée 2D est mieux adaptée au problème que nous traitons que la transformée BCR.

3.3. Perspective: application aux algorithmes ARTUR et LEGEND

Nous avons remarqué au Chapitre III que la première étape d'ARTUR et LEGEND correspondaient à une régularisation de Tikhonov. Les résultats que nous avons obtenus en utilisant la résolution dans le domaine de la TO, pour la régularisation de Tikhonov nous incitent à aller plus loin: appliquer ces techniques à ARTUR et LEGEND. Nous rappelons la forme générale de ces algorithmes (figure 5-10).

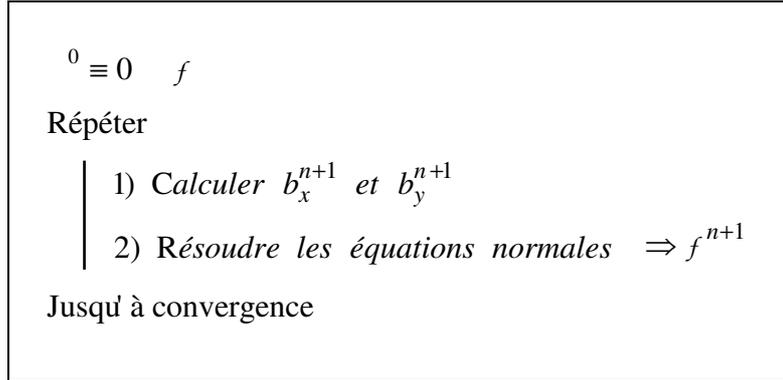


Figure 5-10: Forme générale des algorithmes ARTUR et LEGEND

A titre de perspective, nous donnons ici les versions des algorithmes ARTUR et LEGEND avec résolution des équations normales dans le domaine de la TO.

3.3.1. Cas d'ARTUR

Dans le cas d'ARTUR, les équations normales à l'étape $n + 1$ s'écrivent:

$$\left(\mathbf{R}^t \mathbf{R} - \lambda^2 \Delta_A^{n+1} \right) f^{n+1} = \mathbf{R}^t p \quad (5-29)$$

Ce qui s'écrit dans le domaine des ondelettes:

$$\left(\mathcal{T}O \cdot \mathbf{R}^t \mathbf{R} \cdot \tilde{\mathcal{T}}O^{-1} - \lambda^2 \mathcal{T}O \cdot \Delta_A^{n+1} \cdot \tilde{\mathcal{T}}O^{-1} \right) \mathcal{T}O \cdot f^{n+1} = \mathcal{T}O \cdot \mathbf{R}^t p \quad (5-30)$$

On doit donc résoudre à chaque étape un système de la forme:

$$\left(\mathcal{T}O \cdot \mathbf{S}^{n+1} \cdot \tilde{\mathcal{T}}O^{-1} \right) \mathcal{T}O \cdot f^{n+1} = \mathcal{T}O \cdot \mathbf{R}^t p \quad (5-31)$$

Avant de résoudre ce système, il faut commencer par former chacun de ses termes. La matrice

$\mathcal{TO} \cdot R^t R \cdot \tilde{\mathcal{TO}}^{-1}$ peut être calculée par avance, une fois pour toutes. Le vecteur $\mathcal{TO} \cdot R^t p$ peut être calculé au début de l'algorithme.

Par contre, la matrice $\mathcal{TO} \cdot \Delta_A^{n+1} \cdot \tilde{\mathcal{TO}}^{-1}$ doit être calculée à chaque étape, puisqu'elle dépend des nouvelles valeurs de la variable auxiliaire. Ce calcul implique la formation de Δ_A^{n+1} , puis sa transformée en ondelettes. On aurait donc à faire à chaque étape la transformée en ondelettes d'une matrice $N^2 \times N^2$. Cela serait trop coûteux en temps de calcul. Pour palier cette difficulté, on remarque que l'on a en fait:

$$-\Delta_A^{n+1} = D_x^t \Lambda_x^{n+1} D_x + D_y^t \Lambda_y^{n+1} D_y \quad (5-32)$$

avec

$$\Lambda_x^{n+1} = \text{diag} \left\{ \left(b_x^{n+1} \right)_k \right\} \text{ et } \Lambda_y^{n+1} = \text{diag} \left\{ \left(b_y^{n+1} \right)_k \right\} \quad (5-33)$$

En précalculant les matrices $\mathcal{TO} \cdot D_x^t$, $D_x \cdot \tilde{\mathcal{TO}}^{-1}$, $\mathcal{TO} \cdot D_y^t$ et $D_y \cdot \tilde{\mathcal{TO}}^{-1}$, le calcul à chaque étape de $\mathcal{TO} \cdot \Delta_A^{n+1} \cdot \tilde{\mathcal{TO}}^{-1}$ ne met plus en jeu que des multiplications de matrices diagonales par des matrices creuses, multiplications de matrices creuses entre elles et additions de matrices creuses. Ceci reste tout de même assez coûteux en temps de calcul. La forme générale de l'algorithme est donnée figure 5-11.

$^0 \equiv 0 \quad f$

Répéter

1) Calculer b_x^{n+1} et b_y^{n+1}	
2) Former la matrice $\mathcal{TO} \cdot S^{n+1} \cdot \tilde{\mathcal{TO}}^{-1}$	
3) Résoudre les équations normales	$\Rightarrow \mathcal{TO} \cdot f^{n+1}$
4) Faire la TO inverse de $\mathcal{TO} \cdot f^{n+1}$	$\Rightarrow f^{n+1}$

Jusqu'à convergence

Figure 5-11: Forme générale de l'algorithme ARTUR avec résolution des équations normales dans le domaine de la TO

On remarque qu'ARTUR nécessite la construction d'une matrice transformée à chaque étape.

Les éléments diagonaux de cette matrice sont modifiés à chaque fois, en fonction des pondérations calculées. Il est donc probable qu'on doive choisir un seuil différent à chaque étape. Il est également probable que plus on s'éloignera de la matrice associée à la régularisation de Tikhonov, plus on aura besoin de précision. Ceci risque donc de nous amener à utiliser un seuil de moins en moins bas, et donc à gagner de moins en moins de temps et de mémoire.

3.3.2. Cas de LEGEND

Dans le cas de LEGEND, les équations normales à l'étape $n + 1$ s'écrivent:

$$(R^t R - \lambda^2 \Delta) f^{n+1} = R^t p + \lambda^2 D_x^t b_x^{n+1} + \lambda^2 D_y^t b_y^{n+1} \quad (5-34)$$

Ce qui s'écrit, dans le domaine des ondelettes, sous la forme:

$$\left(\mathcal{TO} \cdot S \cdot \tilde{\mathcal{TO}}^{-1} \right) \mathcal{TO} \cdot f^{n+1} = \mathcal{TO} \cdot R^t p + \lambda^2 \mathcal{TO} \cdot D_x^t b_x^{n+1} + \lambda^2 \mathcal{TO} \cdot D_y^t b_y^{n+1} \quad (5-35)$$

Cette fois-ci, on n'a plus à calculer à chaque étape la matrice $\mathcal{TO} \cdot S \cdot \tilde{\mathcal{TO}}^{-1}$. Seul le second membre doit être recalculé. En précalculant les matrices $\mathcal{TO} \cdot D_x^t$ et $\mathcal{TO} \cdot D_y^t$, cela n'implique que des multiplications entre matrices creuses et vecteurs pleins. Chaque étape de l'algorithme LEGEND utilisant la résolution dans le domaine de la TO (cf. figure 5-12) est donc beaucoup plus simple que dans le cas d'ARTUR.

$f^0 \equiv 0$

Répéter

1) Calculer b_x^{n+1} et b_y^{n+1}	
2) Former $\lambda^2 \mathcal{TO} \cdot D_x^t b_x^{n+1} + \lambda^2 \mathcal{TO} \cdot D_y^t b_y^{n+1}$	
3) Résoudre les équations normales	$\Rightarrow \mathcal{TO} \cdot f^{n+1}$
4) Faire la TO inverse de $\mathcal{TO} \cdot f^{n+1}$	$\Rightarrow f^{n+1}$

Jusqu'à convergence

Figure 5-12: Forme générale de l'algorithme LEGEND avec résolution des équations normales dans le domaine de la TO

Dans ce cas, contrairement à ARTUR, la matrice peut être traitée une fois pour toutes, de façon

identique au cas de Tikhonov

4. Conclusion, perspectives de travail

Dans ce chapitre, nous avons présenté l'utilisation de la transformée en ondelettes pour la résolution de problèmes inverses de grandes dimensions. Les résultats obtenus lorsque l'on utilise une méthode de reconstruction mettant en œuvre une régularisation de Tikhonov sont très encourageants. Ils montrent qu'il est possible d'introduire un grand nombre de zéros dans la matrice sans perdre trop au niveau de la qualité de l'image reconstruite. Ils montrent également que l'utilisation d'une arithmétique spéciale, dédiée aux matrices "creuses" permet alors de gagner jusqu'à un facteur 10 au niveau du temps de calcul.

La technique consistant à résoudre le problème initial dans le domaine de la transformée en ondelettes semble donc répondre parfaitement à nos objectifs: diminuer l'encombrement-mémoire des matrices et le temps de reconstruction tout en gardant le contrôle sur la qualité d'image fournie. Les perspectives de travail sont encore nombreuses. En particulier, il reste à étudier cette technique avec nos algorithmes de reconstruction basés sur une régularisation avec prise en compte des discontinuités. Nous avons donné ici l'expression des algorithmes ARTUR et LEGEND dans le domaine de la transformée en ondelettes.

Dans l'algorithme ARTUR, on modifie à chaque étape la matrice transformée intervenant dans les équations normales à résoudre. De ce fait, on doit choisir un seuil différent à chaque étape. De plus, la formation de la matrice transformée reste une opération coûteuse. Par contre, dans le cas de LEGEND, la matrice peut être traitée une fois pour toutes, de façon identique au cas de Tikhonov. L'algorithme LEGEND paraît donc mieux adapté à la résolution dans le domaine de la TO.

CONCLUSION, PERSPECTIVES

Dans cette thèse, nous nous sommes intéressés à la reconstruction d'image. Pour reconstruire des images de bonne qualité, il est nécessaire d'avoir défini:

- un “bon” modèle de formation des données,
- un “bon” modèle d'image à reconstruire,
- un “bon” algorithme de reconstruction.

Dans nos travaux, nous nous sommes plus particulièrement intéressés aux deux derniers points.

- En ce qui concerne le modèle d'image à reconstruire, nous avons choisi de considérer qu'une image est formée de zones homogènes séparées par des bords francs. Ce type d'*a priori* définit la régularisation avec prise en compte des discontinuités.

Dans un premier temps, nous nous sommes posés la question suivante: “quelles conditions faut-il imposer à la fonction de potentiel, φ , pour définir une régularisation avec prise en compte des discontinuités?”. Une étude bibliographique nous a montré qu'il existait principalement deux approches. L'une travaille directement sur la forme de la fonction de potentiel, l'autre introduit un processus de ligne permettant de marquer les discontinuités afin de les préserver. Le théorème de Geman et Reynolds [GEMA92] fournit un passage entre les deux approches. L'étude des équations normales associées au critère du MAP nous a permis de constater que les conditions posées par Geman et Reynolds sont trop restrictives.

Í Il nous paraît plus judicieux de poser les conditions assurant la prise en compte des discontinuités sur la fonction de pondération, $\varphi'(u)/2u$, plutôt que sur la fonction de potentiel

elle-même. Les conditions que nous proposons sont plus générales que celles de Geman et Reynolds. En particulier, elles autorisent l'usage de potentiels convexes, rendant ainsi le critère du MAP facile à minimiser.

Dans un second temps, comme dans [GEMA93], nous avons proposé d'étendre le concept de processus de ligne au concept de variable auxiliaire. Cette variable auxiliaire b interagit avec la variable principale f , soit de façon multiplicative, soit de façon additive. C'est la régularisation semi-quadratique. Dans ce cas, le problème de la minimisation du critère du MAP (fonction de f) se trouve remplacé par celui de la minimisation d'une énergie duale (fonction de f et de b). Ce second problème est plus simple à résoudre. En effet, l'énergie duale est quadratique en f lorsque l'on fixe b , et convexe en b lorsque l'on fixe f .

Í Nous avons montré que la variable auxiliaire peut être introduite dès lors que l'on se trouve dans les conditions assurant la prise en compte des discontinuités. De plus, nous donnons l'expression de la valeur de b minimisant l'énergie duale à f fixé.

- En ce qui concerne la définition d'algorithmes de reconstruction, les propriétés de la régularisation semi-quadratique n'avaient pour l'instant été exploitées que dans une approche stochastique [GEMA93].

Í Au Chapitre III, nous avons mis à profit les propriétés de la régularisation semi-quadratique et l'expression de b à f fixé. En adoptant une stratégie de minimisations alternées en b et en f , nous avons défini deux algorithmes déterministes, ARTUR et LEGEND. Comparativement à la plupart des algorithmes connus à ce jour, ARTUR et LEGEND sont plus généraux, au sens où ils peuvent être mis en œuvre pour une variété importante de fonctions de potentiels.

ARTUR et LEGEND sont des algorithmes à introduction progressive de discontinuités. Le problème de la résolution des équations normales non linéaires associées au critère du MAP est remplacé par une suite de résolution de systèmes linéaires évoluant au fil des étapes. A chaque étape, le nouveau système est obtenu grâce aux valeurs de la variable auxiliaire, calculées à l'aide de l'image estimée à l'étape précédente. La variable auxiliaire marque les discontinuités détectées à chaque étape. Les algorithmes réalisent donc l'estimation conjointe de l'image et de la "carte" des discontinuités associée.

Í Nous avons proposé une étude théorique de la convergence des algorithmes. Nous avons donné plusieurs résultats valables dans le cas convexe comme dans le cas non convexe. Dans le cas convexe, on montre de plus que les algorithmes convergent vers la solution unique du MAP.

Nous avons également comparé notre approche avec les algorithmes MFA et GNC. Ces techniques reposent toutes sur le même principe: minimiser d'abord une fonctionnelle convexe, et traquer ensuite le minimum en modifiant doucement le critère. La différence entre les

méthodes réside dans la façon dont on fait évoluer le critère. Le GNC et le MFA utilisent la technique de variation de la fonction de potentiel. Pour minimiser chaque critère, le MFA utilise de plus une variable auxiliaire, comme ARTUR. Cette étude nous a permis de mettre à jour les similitudes et les différences entre ces algorithmes.

À titre de perspective, nous avons proposé l'algorithme MOISE, version d'ARTUR adaptée au cas de la vraisemblance Poissonienne. Cet algorithme est actuellement étudié par P.M. Koulibaly dans le cadre de sa thèse [KOUL95].

Au Chapitre IV, nous avons donné des exemples d'application de ces algorithmes à la reconstruction tomographique. Ceci nous a permis de mieux comprendre le fonctionnement et les propriétés de ces algorithmes. Nous avons pu vérifier leur convergence dans le cas convexe. Dans ce cas, la qualité d'image obtenue est cependant moins bonne que dans le cas où on utilise un potentiel non convexe. Lorsqu'on utilise un potentiel non convexe, la convergence n'est pas prouvée théoriquement, mais on l'observe expérimentalement. Dans tous les cas, les résultats obtenus, à partir de données simulées comme à partir de données réelles, sont visuellement très satisfaisants.

Les algorithmes ARTUR et LEGEND réalisent un bon compromis entre coût de calcul et qualité d'image reconstruite. De plus, ils sont assez simples à mettre en œuvre, et assez généraux pour être adaptés à différentes applications. En ce sens, ils nous paraissent être de bons algorithmes de reconstruction d'image.

- Dans cette thèse, notre travail a surtout porté sur le modèle d'image considéré. Un "bon" modèle d'image à reconstruire n'est pas le seul point important pour obtenir des images de très bonne qualité. Des améliorations importantes peuvent être obtenues en travaillant sur le modèle de formation des données.

Dans certaines applications, des techniques utilisées habituellement en estimation robuste peuvent permettre d'améliorer les résultats obtenus. Par exemple, M. Black et P. Anandan ont proposé dans [BLAC91] d'utiliser une modélisation Gibbsienne du bruit plutôt qu'une modélisation Gaussienne. Cela revient à remplacer le terme quadratique d'attache aux données par une fonctionnelle du même type que celles utilisées pour la régularisation. On peut, de façon analogue à ce que nous avons présenté ici, envisager d'introduire une variable auxiliaire permettant de marquer les données auxquelles on ne souhaite pas se fier pour la reconstruction [BLAC93]. L'adaptation des techniques développées dans cette thèse à ce type de modèle est actuellement étudiée par B. Rouchouze dans le cadre de sa thèse sur l'estimation de mouvement.

En ce qui concerne l'application à la tomographie par émission, une approximation Gaussienne du modèle Poissonien de formation des données permet de développer des algorithmes beaucoup

moins coûteux. Les avis demeurent cependant partagés quant à la validité pratique d'une telle approximation. Sur ce point, l'algorithme MOISE offre de bonnes perspectives de travail. Outre l'étude de l'algorithme lui-même, la comparaison avec les résultats obtenus par ARTUR pour divers taux de comptage de photons pourrait nous permettre d'apporter une réponse.

Enfin, toujours en reconstruction tomographique, il est important de prendre en compte des phénomènes physiques intervenant lors de l'acquisition des projections, au niveau de la construction de la matrice de Radon. En fournissant un modèle plus proche de la réalité, ceci devrait permettre une meilleure qualité d'image reconstruite, ce qui intéresse au premier chef les médecins. La correction des phénomènes physiques est actuellement expérimentée au Centre Antoine Lacassagne du CHR de Nice par P. M. Koulibaly sous la direction du Pr J. Darcourt. Les résultats obtenus en prenant simplement en compte le phénomène d'atténuation sont déjà très encourageants [KOUL95].

- Notre étude sur le modèle d'image à utiliser n'est pas terminée. Bien au contraire, plusieurs sujets de réflexion sont apparus à l'occasion de cette thèse.

En particulier, on peut remarquer que les modèles développés ici supposent qu'il n'y a pas d'interaction entre les éléments des variables auxiliaires. Bien que nous réalisions conjointement l'estimation d'une image et de la carte des discontinuités associée, nous ne posons des contraintes que sur l'image. Or, on sait que les lignes de contours (marquées par les variables auxiliaires) ont une certaine structure. On pourrait envisager de mettre à profit cette structure afin d'améliorer la qualité de l'estimation. Cela pourrait se faire en imposant une contrainte sur l'organisation spatiale de la variable auxiliaire. D'autre part, on souhaiterait que les lignes de contours obtenues soient relativement homogènes en intensité. Ceci pourrait également constituer un *a priori* intéressant.

D'autre part, les contraintes que nous avons posées ne portent que sur le *gradient* de l'image. Dans [GEMA93], Geman et Yang émettent l'hypothèse qu'une image réelle est plutôt formée de zones de gradient constant, plutôt que de zones d'intensité constante. De façon à améliorer la qualité des images produites, ils proposent d'imposer en même temps un *a priori* sur la dérivée et la dérivée seconde de l'image. Il serait intéressant d'expérimenter un tel modèle dans le cas de la reconstruction tomographique. Enfin, l'introduction d'une contrainte de positivité peut être envisagée.

- Au niveau des algorithmes de reconstruction, il reste également de bonnes perspectives de travail.

Comme nous l'avons évoqué à la fin du Chapitre III, d'autres stratégies sont envisageables pour

la minimisation de l'énergie duale. Ces solutions, comme par exemple "ARTUR explicite" méritent d'être étudiées, d'un point de vue théorique et expérimental. En particulier, il serait intéressant d'étudier les propriétés de convergence de ces algorithmes, dans le cadre de la régularisation semi-quadratique. Une étude des schémas dynamiques est actuellement réalisée par L. Lazaroia dans le cadre de sa thèse [LAZA95], sous la direction de G. Aubert.

En second lieu, l'application de la transformée en ondelettes semble être un bon moyen pour faciliter la résolution de systèmes linéaires de grandes dimensions. Nous avons vu au Chapitre V tout le bénéfice que l'on pouvait tirer de cette technique, dans le cas simple d'un algorithme de reconstruction basé sur une régularisation de Tikhonov. Le passage à la régularisation semi-quadratique demande une étude particulière. Il semble que LEGEND soit plus adapté qu'ARTUR à la résolution dans le domaine des ondelettes. Il reste encore à étudier l'effet du "creusement" de la matrice sur la vitesse et la qualité de la reconstruction. Pour l'application à la reconstruction tomographique, un tel travail est crucial dans l'optique du passage au cas 3D [LAUR94], et à des tailles d'image plus importantes.

Annexes

Annexe A: Calcul des équations normales associées à la minimisation du critère du MAP

1. Introduction

Le critère du MAP est défini par: $J(f) = J_1(f) + \lambda^2 J_2(f)$

avec $J_1(f) = \|p - Rf\|^2$ et $J_2(f) = \sum_{i,j} \varphi[(D_X f)_{i,j}] + \sum_{i,j} \varphi[(D_Y f)_{i,j}]$.

Un minimum de J vérifie nécessairement: $\frac{\partial J}{\partial f} = 0$

Nous allons maintenant calculer cette dérivée.

2. Dérivation du premier terme

La dérivée de $J_1(f) = \|p - Rf\|^2 = (p - Rf)^t (p - Rf)$ est donnée par:

$$\frac{\partial J_1}{\partial f} = 2(R^t Rf - R^t p)$$

3. Dérivation du second terme

Nous allons calculer l'expression locale de la dérivée de $J_2(f)$ au site (i,j), pour les sites "loins des bords" de l'image, les problèmes de bords nécessitant un traitement particulier qui ne sera pas abordé ici. Nous remarquons tout d'abord que dans $J_2(f)$, quatre termes seulement font apparaître la valeur de f au site (i,j):

$$\begin{array}{ll} \varphi(f_{i,j+1} - f_{i,j}) & \varphi(f_{i,j} - f_{i,j-1}) \\ \varphi(f_{i+1,j} - f_{i,j}) & \varphi(f_{i,j} - f_{i-1,j}) \end{array}$$

Nous avons donc:

$$\begin{aligned} \left(\frac{\partial J_2}{\partial f} \right)_{i,j} &= -\varphi'(f_{i,j+1} - f_{i,j}) + \varphi'(f_{i,j} - f_{i,j-1}) \\ &\quad -\varphi'(f_{i+1,j} - f_{i,j}) + \varphi'(f_{i,j} - f_{i-1,j}) \end{aligned}$$

ce qui peut se mettre sous la forme:

$$\left(\frac{\partial J_2}{\partial f}\right)_{i,j} = 2 \left[-\left(f_{i,j+1} - f_{i,j}\right) \frac{\varphi'(f_{i,j+1} - f_{i,j})}{2(f_{i,j+1} - f_{i,j})} + \left(f_{i,j} - f_{i,j-1}\right) \frac{\varphi'(f_{i,j} - f_{i,j-1})}{2(f_{i,j} - f_{i,j-1})} \right. \\ \left. - \left(f_{i+1,j} - f_{i,j}\right) \frac{\varphi'(f_{i+1,j} - f_{i,j})}{2(f_{i+1,j} - f_{i,j})} + \left(f_{i,j} - f_{i-1,j}\right) \frac{\varphi'(f_{i,j} - f_{i-1,j})}{2(f_{i,j} - f_{i-1,j})} \right]$$

Remarque: les termes de cette expression ne sont normalement pas définis lorsque la différence entre pixels voisins, u , est nulle. Toutefois, nous considérerons cette écriture valide pour $u \in [0, +\infty[$, moyennant éventuellement un prolongement par continuité en 0. Dans cette expression, on voit apparaître des coefficients de la forme: $\varphi'(u)/2u$. Nous noterons:

$\lambda^E = \frac{\varphi'(f_{i,j+1} - f_{i,j})}{2(f_{i,j+1} - f_{i,j})}$	$\lambda^W = \frac{\varphi'(f_{i,j} - f_{i,j-1})}{2(f_{i,j} - f_{i,j-1})}$
$\lambda^S = \frac{\varphi'(f_{i+1,j} - f_{i,j})}{2(f_{i+1,j} - f_{i,j})}$	$\lambda^N = \frac{\varphi'(f_{i,j} - f_{i-1,j})}{2(f_{i,j} - f_{i-1,j})}$

et $\Sigma = \lambda^E + \lambda^W + \lambda^S + \lambda^N$. On peut alors écrire:

$$\left(\frac{\partial J_2}{\partial f}\right)_{i,j} = 2 \left[\Sigma f_{i,j} - \lambda^E f_{i,j+1} - \lambda^W f_{i,j-1} - \lambda^S f_{i+1,j} - \lambda^N f_{i-1,j} \right]$$

Ceci revient à filtrer l'image par un Laplacien négatif pondéré:

$$\frac{\partial J_2}{\partial f} = -2\Delta_{pond} f,$$

les coefficients du Laplacien pondéré Δ_{pond} au site (i,j) étant donnés par:

$$\begin{pmatrix} 0 & \lambda^N & 0 \\ \lambda^W & -\Sigma & \lambda^E \\ 0 & \lambda^S & 0 \end{pmatrix}$$

On obtient donc finalement l'expression des équations normales associées à la minimisation du critère du MAP:

$$\frac{\partial J}{\partial f} = 0 \Leftrightarrow (R^t R - \lambda^2 \Delta_{pond}) f = R^t p$$

Annexe B: Preuve du théorème 2 “de Geman et Reynolds étendu”

La preuve de ce théorème est similaire à celle du théorème de Geman&Reynolds. Sans perdre en généralité, on peut supposer que φ est continûment dérivable.

1. Introduction

Notons $\theta(u) = \varphi(\sqrt{u})$. Dans les conditions du théorème, $\theta(u)$ est strictement concave. Cette propriété a deux conséquences:

i) D’une part, $\theta'(u) = \frac{\varphi'(\sqrt{u})}{2\sqrt{u}}$ est strictement décroissante sur $]0, \infty[$. Posons:

$$L = \lim_{x \rightarrow +\infty} \frac{\varphi'(x)}{2x} \quad \text{et} \quad M = \lim_{x \rightarrow 0^+} \frac{\varphi'(x)}{2x}$$

Si ces limites existent, alors θ' est bijective et admet un inverse $(\theta')^{-1}:]L, M] \rightarrow]0, +\infty[$.

ii) D’autre part, $\forall w, v, w \neq v \quad \theta(w) < \theta(v) + (w - v)\theta'(v)$, ce qui revient à dire que θ est l’inf d’une famille de fonctions paramétrées par v :

$$\theta(w) = \inf_v \left\{ \theta'(v)w + \theta(v) - v\theta'(v) \right\} \quad (1)$$

2. Preuve du 1):

Posons:

$$\boxed{\theta'(v) = b} \quad (2)$$

ou de façon équivalente:

$$v = (\theta')^{-1}(b) \quad (3)$$

Dans ce cas, (1) devient:

$$\theta(w) = \inf_b \left\{ bw + \theta((\theta')^{-1}(b)) - b(\theta')^{-1}(b) \right\} \quad (4)$$

En posant:

$$\boxed{\psi(b) = \theta((\theta')^{-1}(b)) - b(\theta')^{-1}(b)}$$

l'expression de $\theta(u)$ se met sous la forme:

$$\theta(w) = \inf_b \{bw + \psi(b)\} \quad (5)$$

• La fonction ψ est à valeurs sur l'intervalle défini par:

$$\psi(L) = \alpha = \lim_{v \rightarrow +\infty} (\theta(v) - v\theta'(v)) \quad \text{et} \quad \psi(M) = \beta = \lim_{v \rightarrow 0^+} (\theta(v) - v\theta'(v))$$

soit, en posant $v = u^2$:

$$\alpha = \lim_{u \rightarrow +\infty} \left(\varphi(u) - u^2 \frac{\varphi'(u)}{2u} \right) \quad \text{et} \quad \beta = \lim_{u \rightarrow 0^+} \left(\varphi(u) - u^2 \frac{\varphi'(u)}{2u} \right)$$

soit encore

$$\alpha = \lim_{u \rightarrow +\infty} \left(\varphi(u) - u^2 \frac{\varphi'(u)}{2u} \right) \quad \text{et} \quad \beta = \lim_{u \rightarrow 0^+} \varphi(u)$$

• Un calcul élémentaire montre que $\psi'(b) = -(\theta')^{-1}(b)$:

$$\begin{aligned} \psi'(b) &= \left[(\theta')^{-1}(b) \right]' \theta'((\theta')^{-1}(b)) - (\theta')^{-1}(b) - b \left[(\theta')^{-1}(b) \right]' \\ &= \left[(\theta')^{-1}(b) \right]' b - (\theta')^{-1}(b) - b \left[(\theta')^{-1}(b) \right]' \\ &= -(\theta')^{-1}(b) \end{aligned}$$

Comme $(\theta')^{-1}$ est à valeurs sur $]0, +\infty[$, cela montre que ψ est strictement décroissante sur $]L, M[$.

• D'autre part, il est facile de montrer que $\psi''(b) = \frac{-1}{\theta''(v)}$. En effet, on a par définition:

$$\begin{aligned} \psi''(b) &= \lim_{c \rightarrow 0} \frac{\psi'(b+c) - \psi'(b)}{c} \\ &= \lim_{c \rightarrow 0} \frac{\psi'(b+c) - \psi'(b)}{b+c-b} \quad \forall b \\ &= \lim_{c \rightarrow 0} \frac{-(\theta')^{-1}(b+c) + (\theta')^{-1}(b)}{\theta'[(\theta')^{-1}(b+c)] - \theta'[(\theta')^{-1}(b)]} \end{aligned}$$

En posant $e = (\theta')^{-1}(b + c)$ et en utilisant (3), il vient donc:

$$\begin{aligned}\psi''(b) &= \lim_{e \rightarrow v} -\frac{v-e}{\theta'[v]-\theta'[e]} \\ &= -\frac{1}{\theta''(v)}\end{aligned}$$

Or, dans les hypothèses du théorème, $\theta''(v)$ est strictement négative. La fonction ψ est donc strictement convexe.

• Enfin, en posant $w = u^2$, l'équation (5) devient:

$$\theta(u^2) = \varphi(u) = \inf_b \{bu^2 + \psi(b)\}$$

3. Preuve du 2):

D'après (1), il est évident qu'à w fixé, l'inf est atteint pour $v = w$, c'est-à-dire pour $b = \theta'(w)$. Avec $w = u^2$, nous avons donc bien $b_u = \frac{\varphi'(u)}{2u}$. On prolongera cette expression par continuité en 0.

Annexe C: Preuve du théorème 3 “Paires de Legendre”

1. Introduction: les paires de Legendre

Définition: Soit f une fonction convexe. La **transformée de Legendre** de f , notée f^* , est convexe et définie par:

$$f^*(b) = \sup_u [ub - f(u)]$$

On sait aussi [ROCK] que f et sa transformée de Legendre sont telles que

$$(f^*)^*(u) = f(u)$$

Définition: f et g forment une **paire de Legendre** si et seulement si

f et g sont convexes

et

$$f^*(b) = g(b) \text{ et } f(u) = g^*(u)$$

Si de plus f est strictement convexe et dérivable, alors g est strictement convexe et dérivable.

Propriété 1: Si f et g forment une paire de Legendre, et sont strictement convexes, alors à u fixé, la valeur b_u réalisant le sup en b de $[ub - g(b)]$ est unique et définie par

$$b_u = f'(u).$$

Démonstration: Soit b_u la valeur qui réalise le sup en b de $[ub - g(b)]$. La fonction g étant strictement convexe, b_u est unique et vérifie: $u - g'(b_u) = 0$.

D'autre part, on peut écrire $f(u) = g^*(u) = ub_u - g(b_u)$. Si $\partial b_u / \partial u$ existe, la dérivée de cette expression par rapport à u est:

$$f'(u) = b_u + u \frac{\partial b_u}{\partial u} - \frac{\partial b_u}{\partial u} g'(b_u).$$

En utilisant l'expression $u - g'(b_u) = 0$, on obtient $b_u = f'(u)$.

2. Preuve du 1):

- Si $f(u) = \frac{1}{2}[u^2 - \varphi(u)]$ est strictement convexe, sa transformée de Legendre est définie par:

$$g(b) = f^*(b) = \sup_u [ub - f(u)]$$

Soit:

$$g(b) = \sup_u \left[ub - \frac{u^2}{2} + \frac{\varphi(u)}{2} \right]$$

ou encore:

$$g(b) = \frac{1}{2} \left\{ b^2 + \sup_u \left[-(u-b)^2 + \varphi(u) \right] \right\}$$

En posant $\zeta(b) = \sup_u \left[-(u-b)^2 + \varphi(u) \right]$, on a donc

$$g(b) = \frac{1}{2} [b^2 + \zeta(b)].$$

- Par définition, f et g forment une paire de Legendre:

$$f(u) = g^*(u) = \sup_b [ub - g(b)]$$

Cette expression peut se mettre sous la forme:

$$\begin{aligned} f(u) &= \frac{1}{2} \left\{ u^2 + \sup_b \left[-(u-b)^2 - \zeta(b) \right] \right\} \\ &= \frac{1}{2} \left\{ u^2 - \inf_b \left[(u-b)^2 + \zeta(b) \right] \right\} \end{aligned}$$

- En conclusion, si $f(u) = \frac{1}{2}[u^2 - \varphi(u)]$ est strictement convexe, alors il existe une fonction $\zeta(b)$ définie par:

$$\boxed{\zeta(b) = \sup_u \left[-(u-b)^2 + \varphi(u) \right]}$$

et telle que

$$\boxed{\varphi(u) = u^2 - 2f(u) = \inf_b \left[(u-b)^2 + \zeta(b) \right] = \inf_b \varphi^*(u, b)}$$

avec $\varphi^*(u, b) = (u-b)^2 + \zeta(b)$.

- Au passage, on peut remarquer que la fonction $\varphi^*(u, b)$ est strictement convexe en b à u fixé.

En effet, un calcul simple montre qu'à u fixé $\frac{\partial \varphi^*(u,b)}{\partial b} = 2(b-u) + \zeta'(b) = -2u + g'(b)$ (d'après la définition de g). Comme g est strictement convexe en b , $g'(b)$ est strictement monotone, et donc $\frac{\partial \varphi^*(u,b)}{\partial b}$ l'est aussi, ce qui équivaut à dire que $\varphi^*(u,b)$ est strictement convexe en b à u fixé.

3. Preuve du 2):

La preuve du lemme 2 est basée sur le fait que les fonctions f et g définies ci-dessus forment une paire de Legendre.

• Notons b_u la valeur réalisant l'inf en b de $\varphi^*(u,b)$ à u fixé. On a donc $b_u = \arg \inf_b [\varphi^*(u,b)]$.

Or, on peut écrire:

$$\begin{aligned} \arg \inf_b [\varphi^*(u,b)] &= \arg \inf_b [(u-b)^2 + \zeta(b)] \\ &= \arg \sup_b [-(u-b)^2 - \zeta(b)] \\ &= \arg \sup_b [ub - \frac{1}{2}(b^2 + \zeta(b))] \end{aligned}$$

Dans notre cas, on a $g(b) = \frac{1}{2}[b^2 + \zeta(b)]$. L'égalité précédente devient alors:

$$\arg \inf_b [\varphi^*(u,b)] = \arg \sup_b [ub - g(b)]$$

• Les fonctions f et g formant une paire de Legendre, par application de la propriété 1, on sait que la valeur b_u réalisant ce sup est unique et donnée par $b_u = f'(u)$. Avec $f(u) = \frac{1}{2}[u^2 - \varphi(u)]$, on a donc:

$$b_u = u - \frac{\varphi'(u)}{2}$$

Soit pour $u \neq 0$:

$$\boxed{b_u = \left(1 - \frac{\varphi'(u)}{2u}\right)u}$$

En supposant que $\lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u}$ existe, on prolongera cette expression par continuité en $u = 0$.

Ceci conclut la démonstration du théorème 3.

• Au passage, nous signalons un petit résultat utile. L'application de la propriété 1 nous donne

$b_u = f'(u)$. D'autre part, on sait (cf. démonstration de la propriété 1) que $u = g'(b_u)$. En appliquant ces résultats avec $f(u) = \frac{1}{2}[u^2 - \varphi(u)]$ et $g(b) = \frac{1}{2}[b^2 + \zeta(b)]$ on obtient:

$$\boxed{\varphi'(u) = \zeta'(b_u)}$$

Ce résultat sera utilisé dans l'annexe D.

4. Conditions de préservations des discontinuités et hypothèses du théorème

Dans ce paragraphe, nous nous proposons de montrer que si la fonction de potentiel $\varphi(u)$ vérifie les conditions suivantes:

$$\begin{aligned} \text{i) } & \lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u} = 1 \\ \text{ii) } & \lim_{u \rightarrow +\infty} \frac{\varphi'(u)}{2u} = 0 \\ \text{iii) } & \frac{\varphi'(u)}{2u} \text{ est continue et strictement décroissante sur } [0, +\infty[. \end{aligned}$$

alors, les conditions d'application du théorème 3:

$$\begin{aligned} \text{a) } & f(u) = \frac{1}{2}(u^2 - \varphi(u)) \text{ soit strictement convexe sur }]0, +\infty[. \\ \text{b) } & \lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u} \text{ existe} \end{aligned}$$

sont automatiquement vérifiées.

Une condition suffisante pour que a) soit vérifiée est que $f''(u) > 0$ sur $]0, +\infty[$, soit:

$$1 - \frac{\varphi''(u)}{2} > 0.$$

Posons $h(u) = \frac{\varphi'(u)}{2u}$. On montre que:

$$\begin{aligned}
 h'(u) &= \frac{2u\varphi''(u) - 2\varphi'(u)}{4u^2} \\
 &= \frac{1}{u} \left[\frac{\varphi''(u)}{2} - \frac{\varphi'(u)}{2u} \right] \\
 &= \frac{1}{u} \left[\frac{\varphi''(u)}{2} - h(u) \right]
 \end{aligned}$$

Supposons que la condition iii) soit vérifiée. On a alors $h'(u) < 0 \quad \forall u > 0$, et donc:

$$\frac{\varphi''(u)}{2} < h(u)$$

Si de plus la condition i) est vérifiée, alors on a

$$\frac{\varphi''(u)}{2} < h(u) \leq 1$$

Ce qui entraîne que

$$1 - \frac{\varphi''(u)}{2} > 0$$

En conclusion, si les conditions i) et iii) sont vérifiées, alors a) est vérifiée.

Annexe D: Preuve du théorème 4

La preuve du théorème 4 suit le même cheminement logique pour les deux algorithmes. Dans cette annexe, nous donnons cette preuve dans le cas de LEGEND. A $b = (b_x, b_y)$ fixé, l'énergie duale s'écrit dans ce cas:

$$J^*(f, b_x, b_y) = \|p - Rf\|^2 + \lambda^2 \|b_x - D_x f\|^2 + \lambda^2 \|b_y - D_y f\|^2 + \sum_k \zeta(b_x)_k + \zeta(b_y)_k$$

Ce critère étant convexe en f , il admet un minimum unique, réalisé en $f = f_b$. On définit:

$$T(b_x, b_y) = \|p - Rf_b\|^2 + \lambda^2 \|b_x - D_x f_b\|^2 + \lambda^2 \|b_y - D_y f_b\|^2 + \sum_k \zeta(b_x)_k + \zeta(b_y)_k$$

et on s'intéresse au problème de minimisation suivant:

$$(P^*) \quad \min_{b_x, b_y} T(b_x, b_y)$$

1. Preuve du 1)

La preuve du 1) est basée sur les deux lemmes suivants:

Lemme 1: Si (P*) admet une solution $\hat{b} = (\hat{b}_x, \hat{b}_y)$ vérifiant

$$\forall k, (b_x)_k > 0 \text{ et } (b_y)_k > 0$$

alors:

$$\frac{\partial T(\hat{b}_x, \hat{b}_y)}{\partial (b_x)_k} = 0 \text{ et } \frac{\partial T(\hat{b}_x, \hat{b}_y)}{\partial (b_y)_k} = 0$$

Ce résultat est élémentaire car on suppose que \hat{b} appartient à un ensemble de contraintes ouvert.

Lemme 2: Les dérivées partielles de la fonction T sont données par:

$$\frac{\partial T(b_x, b_y)}{\partial (b_x)_k} = 2(b_x)_k - 2(D_x f_b)_k + \zeta'[(b_x)_k]$$

et

$$\frac{\partial T(b_x, b_y)}{\partial (b_y)_k} = 2(b_y)_k - 2(D_y f_b)_k + \zeta'[(b_y)_k]$$

Preuve: Nous donnons la preuve du lemme 2 pour la direction x seulement (la preuve pour la direction y est semblable). Soit q^k le vecteur de dimension N^2 tel que:

$$q_l^k = 1 \quad \text{si } l = k$$

$$q_l^k = 0 \quad \text{sinon}$$

Considérons le taux:

$$\delta_x^k T(b_x, b_y) = \frac{T(b_x + \theta q^k, b_y) - T(b_x, b_y)}{\theta} \quad \theta \in \mathfrak{R}, \theta \neq 0$$

• D'après la définition de T , si on note respectivement $f_b = f_{b_x, b_y}$ et $f_b^\theta = f_{b_x + \theta q^k, b_y}$ les solutions de (P_{b_x, b_y}) et $(P_{b_x + \theta q^k, b_y})$, on a alors:

$$\begin{aligned} \delta_x^k T(b_x, b_y) &= \left\{ \|p - Rf_b^\theta\|^2 + \lambda^2 \|b_x + \theta q^k - D_x f_b^\theta\|^2 + \lambda^2 \|b_y - D_y f_b^\theta\|^2 \right. \\ &\quad \left. + \|p - Rf_b\|^2 + \lambda^2 \|b_x - D_x f_b\|^2 + \lambda^2 \|b_y - D_y f_b\|^2 \right\} \\ &\quad + \frac{\zeta[(b_x + \theta q^k)_k] - \zeta[(b_x)_k]}{\theta} \end{aligned}$$

Après quelques manipulations formelles, en posant:

$$\delta \zeta[(b_x)_k] = \frac{\zeta[(b_x + \theta q^k)_k] - \zeta[(b_x)_k]}{\theta}$$

cette équation peut se mettre sous la forme suivante:

$$\begin{aligned}
\delta_x^k T(b_x, b_y) &= 2(b_x)_k + \theta - 2D_x f_b^\theta + \delta \zeta [(b_x)_k] \\
&+ \frac{1}{\theta} \left\{ \left\| R(f_b^\theta - f_b) \right\|^2 + 2 \left\langle R(f_b^\theta - f_b) R f_b - p \right\rangle \right\} \\
&+ \frac{\lambda^2}{\theta} \left\{ \left\| D_x(f_b^\theta - f_b) \right\|^2 + 2 \left\langle D_x(f_b^\theta - f_b) D_x f_b \right\rangle \right\} \\
&+ \frac{\lambda^2}{\theta} \left\{ \left\| D_y(f_b^\theta - f_b) \right\|^2 + 2 \left\langle D_y(f_b^\theta - f_b) D_y f_b \right\rangle \right\} \\
&- 2 \frac{\lambda^2}{\theta} \left\{ \left\langle D_x(f_b^\theta - f_b) b_x \right\rangle + \left\langle D_y(f_b^\theta - f_b) b_y \right\rangle \right\}
\end{aligned} \tag{1}$$

• Comme $f_b = f_{b_x, b_y}$ et $f_b^\theta = f_{b_x + \theta q^k, b_y}$ sont solutions de (P_{b_x, b_y}) et $(P_{b_x + \theta q^k, b_y})$, on a alors:

$$(R^t R + \lambda^2 D_x^t D_x + \lambda^2 D_y^t D_y) f_b = R^t p + \lambda^2 D_x^t b_x + \lambda^2 D_y^t b_y \tag{2}$$

$$(R^t R + \lambda^2 D_x^t D_x + \lambda^2 D_y^t D_y) f_b^\theta = R^t p + \lambda^2 D_x^t (b_x + \theta q^k) + \lambda^2 D_y^t b_y \tag{3}$$

En faisant le produit scalaire de (2) par $(f_b^\theta - f_b)$, on a:

$$\begin{aligned}
&\left\langle R(f_b^\theta - f_b) R f_b - p \right\rangle + \left\langle D_x(f_b^\theta - f_b) D_x f_b \right\rangle + \left\langle D_y(f_b^\theta - f_b) D_y f_b \right\rangle \\
&- \left\langle D_x(f_b^\theta - f_b) b_x \right\rangle - \left\langle D_y(f_b^\theta - f_b) b_y \right\rangle = 0
\end{aligned} \tag{4}$$

D'autre part, en soustrayant l'équation (3) à (2) et en faisant le produit scalaire par $(f_b^\theta - f_b)$, on obtient:

$$\left\| R(f_b^\theta - f_b) \right\|^2 + \lambda^2 \left\| D_x(f_b^\theta - f_b) \right\|^2 + \lambda^2 \left\| D_y(f_b^\theta - f_b) \right\|^2 = \theta \lambda^2 \left\langle D_x(f_b^\theta - f_b) q^k \right\rangle \tag{5}$$

• En utilisant les résultats (4) et (5), il vient:

$$\delta_x^k T(b_x, b_y) = 2(b_x)_k + \theta - 2D_x f_b^\theta + \delta \zeta [(b_x)_k] + \lambda^2 \left\langle D_x(f_b^\theta - f_b) q^k \right\rangle$$

En faisant tendre θ vers 0, on obtient:

$$\frac{\partial T(b_x, b_y)}{\partial (b_x)_k} = 2(b_x)_k - 2(D_x f_b)_k + \zeta'' [(b_x)_k]$$

Ceci conclut la preuve du lemme 2.

- Pour terminer la preuve du 1), l'application des lemmes 1 et 2 nous donne, dans le cas où $\forall k \hat{b}_k > 0$, on a:

$$\frac{\partial T(\hat{b}_x, \hat{b}_y)}{\partial (b_x)_k} = 0 \Leftrightarrow 2(\hat{b}_x)_k - 2(D_x f_{\hat{b}})_k + \zeta'[(\hat{b}_x)_k] = 0$$

Or, on a remarqué dans l'annexe C que: $\varphi'(u) = \zeta'(b_u)$. En appliquant ce résultat, il vient:

$$\frac{\partial T(\hat{b}_x, \hat{b}_y)}{\partial (b_x)_k} = 0 \Leftrightarrow 2(\hat{b}_x)_k - 2(D_x f_{\hat{b}})_k + \varphi'[(D_x f_{\hat{b}})_k] = 0$$

Moyennant un éventuel prolongement par continuité en 0, on a donc:

$$\frac{\partial T(\hat{b}_x, \hat{b}_y)}{\partial (b_x)_k} = 0 \Leftrightarrow (\hat{b}_x)_k = \left\{ 1 - \frac{\varphi'[(D_x f_{\hat{b}})_k]}{2(D_x f_{\hat{b}})_k} \right\} (D_x f_{\hat{b}})_k$$

- Enfin, dans le cas où pour un certain k_0 , $\hat{b}_{k_0} = 0$, on a alors seulement:

$$-2(D_x f_{\hat{b}})_{k_0} + \zeta''[0] \geq 0 \text{ ou encore } -2(D_x f_{\hat{b}})_{k_0} + \frac{\varphi'[(D_x f_{\hat{b}})_{k_0}]}{2} \geq 0$$

On a donc (moyennant un éventuel prolongement par continuité en 0):

$$1 - \frac{\varphi'[(D_x f_{\hat{b}})_{k_0}]}{2(D_x f_{\hat{b}})_{k_0}} \leq 0$$

Or, par définition:

$$1 - \frac{\varphi'[(D_x f_{\hat{b}})_{k_0}]}{2(D_x f_{\hat{b}})_{k_0}} \geq 0$$

On a donc nécessairement:

$$1 - \frac{\varphi'[(D_x f_{\hat{b}})_{k_0}]}{2(D_x f_{\hat{b}})_{k_0}} = 0 = \hat{b}_{k_0}$$

et la conclusion du théorème reste valable.

2. Preuve du 2)

• Si $\hat{b} = (\hat{b}_x, \hat{b}_y)$ est un minimum de $T(\hat{b}_x, \hat{b}_y)$ et que $\forall k, (b_x)_k > 0$ et $(b_y)_k > 0$, alors on a nécessairement pour tout k :

$$\frac{\partial T(\hat{b}_x, \hat{b}_y)}{\partial (b_x)_k} = 0 \text{ et } \frac{\partial T(\hat{b}_x, \hat{b}_y)}{\partial (b_y)_k} = 0$$

ou, de façon équivalente:

$$2(\hat{b}_x)_k - 2(D_x f_{\hat{b}})_k + \zeta'[(\hat{b}_x)_k] = 0 \text{ et } 2(\hat{b}_y)_k - 2(D_y f_{\hat{b}})_k + \zeta'[(\hat{b}_y)_k] = 0$$

Supposons qu'il existe un autre couple $\tilde{b} = (\tilde{b}_x, \tilde{b}_y)$ tel que: $\tilde{b} = \arg \min_{b_x, b_y} J^*(f_{\tilde{b}}, b_x, b_y)$, alors ce couple est unique (le critère étant convexe en b à f fixé) et vérifie:

$$\frac{\partial J^*(f_{\tilde{b}}, \tilde{b}_x, \tilde{b}_y)}{\partial (b_x)_k} = 0 \text{ et } \frac{\partial J^*(f_{\tilde{b}}, \tilde{b}_x, \tilde{b}_y)}{\partial (b_y)_k} = 0$$

ou de façon équivalente:

$$2(\tilde{b}_x)_k - 2(D_x f_{\tilde{b}})_k + \zeta'[(\tilde{b}_x)_k] = 0 \text{ et } 2(\tilde{b}_y)_k - 2(D_y f_{\tilde{b}})_k + \zeta'[(\tilde{b}_y)_k] = 0$$

Comme $g(b) = \frac{1}{2}[b^2 + \zeta(b)]$ est strictement convexe (voir Annexe C), on sait que $2b + \zeta'(b)$ est bijective et on a donc $\tilde{b} = \hat{b}$.

• Par définition on sait que $J(f_{\hat{b}}) = \min_{b_x, b_y} J^*(f_{\hat{b}}, b_x, b_y)$. Nous venons de montrer que ce minimum est atteint pour un unique couple $\hat{b} = (\hat{b}_x, \hat{b}_y)$, qui correspond au minimum de $T(\hat{b}_x, \hat{b}_y)$. En conséquence:

$$J(f_{\hat{b}}) = J^*(f_{\hat{b}}, \hat{b}_x, \hat{b}_y) = \min_{b_x, b_y} T(b_x, b_y)$$

Remarque: ce résultat est également valable dans le cas où $\exists k_0$ tel que $\hat{b}_{k_0} = 0$.

• Par application de la définition de $T(b)$ et du théorème 3, on obtient:

$$J(f_{\hat{b}}) = \min_f J(f)$$

Ceci conclut la démonstration du théorème 4.

Annexe E: Convergence d'ARTUR

L'énergie duale $J^*(f, b) = J^*(f, b_x, b_y)$ est définie par:

$$J^*(f, b_x, b_y) = \|p - Rf\|^2 + \lambda^2 \sum_k \left\{ (b_x)_k (D_x f)_k^2 + \psi[(b_x)_k] \right\} + \lambda^2 \sum_k \left\{ (b_y)_k (D_y f)_k^2 + \psi[(b_y)_k] \right\}$$

• En premier lieu, considérons la suite $J_n = J^*(f^n, b_x^{n+1}, b_y^{n+1})$, et montrons que J_n est convergente. ARTUR est basé sur le principe des minimisations alternées. Dans cette approche $b^{n+1} = (b_x^{n+1}, b_y^{n+1})$ est obtenu par minimisation de $J^*(f^n, b)$ et vérifie donc:

$$J^*(f^n, b^{n+1}) \leq J^*(f^n, b) \quad \forall b, \quad \forall n \quad (1)$$

De la même façon, f^n est tel que:

$$J^*(f^n, b^n) \leq J^*(f, b^n) \quad \forall f, \quad \forall n \quad (2)$$

On a donc:

$$\begin{aligned} J_{n-1} - J_n &= J^*(f^{n-1}, b^n) - J^*(f^n, b^{n+1}) \\ J_{n-1} - J_n &= [J^*(f^n, b^n) - J^*(f^n, b^{n+1})] + [J^*(f^{n-1}, b^n) - J^*(f^n, b^n)] \end{aligned} \quad (3)$$

En appliquant (1) pour $b = b^n$ et (2) pour $f = f^{n-1}$, on obtient:

$$J^*(f^n, b^n) - J^*(f^n, b^{n+1}) \geq 0 \quad (4)$$

$$J^*(f^{n-1}, b^n) - J^*(f^n, b^n) \geq 0 \quad (5)$$

On en déduit donc que la suite J_n est décroissante:

$$\boxed{J_{n-1} - J_n \geq 0 \quad \forall n}$$

De plus, J_n est tel que:

$$J_n \geq \lambda^2 \sum_k \psi[(b_x^{n+1})_k] + \lambda^2 \sum_k \psi[(b_y^{n+1})_k] \geq 2\lambda^2 N^2 \left[\inf_{0 \leq b \leq 1} \psi(b) \right] \quad \forall n$$

Comme ψ est décroissante (voir Annexe B), on a $J_n \geq 2\lambda^2 N^2 \psi(1) \quad \forall n$. Or, on sait que $\psi(1) = \lim_{u \rightarrow 0^+} \left[\varphi(u) - u^2 \frac{\varphi'(u)}{2u} \right]$. Il est immédiat de constater que si φ vérifie $\lim_{u \rightarrow 0^+} \frac{\varphi'(u)}{2u} = 1$ et $\varphi(0) = 0$, alors $\psi(1) = 0$. Donc $J_n \geq 0 \quad \forall n$

Comme elle est décroissante et minorée par 0, la suite J_n est convergente.

• Montrons maintenant que: $\left[(b_x^{n+1})_k - (b_x^n)_k \right]_{n \rightarrow +\infty} \rightarrow 0$ et $\left[(b_y^{n+1})_k - (b_y^n)_k \right]_{n \rightarrow +\infty} \rightarrow 0 \quad \forall k$.

De (3) et (5) on déduit:

$$J_{n-1} - J_n \geq J^*(f^n, b^n) - J^*(f^n, b^{n+1}) \quad (6)$$

Or, par définition:

$$\begin{aligned} J^*(f^n, b^n) - J^*(f^n, b^{n+1}) &= \lambda^2 \sum_k \left\{ (b_x^n)_k (D_x f^n)_k^2 + \psi \left[(b_x^n)_k \right] \right\} \\ &\quad + \lambda^2 \sum_k \left\{ (b_y^n)_k (D_y f^n)_k^2 + \psi \left[(b_y^n)_k \right] \right\} \\ &\quad - \lambda^2 \sum_k \left\{ (b_x^{n+1})_k (D_x f^n)_k^2 + \psi \left[(b_x^{n+1})_k \right] \right\} \\ &\quad - \lambda^2 \sum_k \left\{ (b_y^{n+1})_k (D_y f^n)_k^2 + \psi \left[(b_y^{n+1})_k \right] \right\} \end{aligned}$$

Définissons $g_k^n(b) = b (D_x f^n)_k^2 + \psi(b)$ et $h_k^n(b) = b (D_y f^n)_k^2 + \psi(b)$. Nous avons alors:

$$J^*(f^n, b^n) - J^*(f^n, b^{n+1}) = \lambda^2 \sum_k \left(g_k^n \left[(b_x^n)_k \right] - g_k^n \left[(b_x^{n+1})_k \right] \right) + \lambda^2 \sum_k \left(h_k^n \left[(b_y^n)_k \right] - h_k^n \left[(b_y^{n+1})_k \right] \right) \quad (7)$$

En utilisant un développement de Taylor des fonctions $g_k^n(\cdot)$ et $h_k^n(\cdot)$, nous écrivons:

$$\begin{aligned} \sum_k \left(g_k^n \left[(b_x^n)_k \right] - g_k^n \left[(b_x^{n+1})_k \right] \right) &= \sum_k \left(\left[(b_x^n)_k - (b_x^{n+1})_k \right] (g_k^n)' \left[(b_x^{n+1})_k \right] \right) \\ &\quad + \frac{1}{2} \sum_k \left(\left[(b_x^n)_k - (b_x^{n+1})_k \right]^2 (g_k^n)'' \left(c_k^n \right) \right) \end{aligned}$$

et

$$\sum_k \left(h_k^n \left[\left(b_y^n \right)_k \right] - h_k^n \left[\left(b_y^{n+1} \right)_k \right] \right) = \sum_k \left(\left[\left(b_y^n \right)_k - \left(b_y^{n+1} \right)_k \right] \left[h_k^n \right]' \left[\left(b_y^{n+1} \right)_k \right] \right) \\ + \frac{1}{2} \sum_k \left(\left[\left(b_y^n \right)_k - \left(b_y^{n+1} \right)_k \right]^2 \left[h_k^n \right]'' \left(d_k^n \right) \right)$$

où $c_k^n \in]b_k^n, b_k^{n+1}[$ [et $d_k^n \in]b_k^n, b_k^{n+1}[$. Comme b^{n+1} réalise le minimum de $J^*(f^n, b)$, nous avons:

$$\left(g_k^n \right)' \left[\left(b_x^{n+1} \right)_k \right] = \left(D_x f^n \right)_k^2 + \psi' \left[\left(b_x^{n+1} \right)_k \right] = 0 \text{ et } \left(h_k^n \right)' \left[\left(b_y^{n+1} \right)_k \right] = \left(D_y f^n \right)_k^2 + \psi' \left[\left(b_y^{n+1} \right)_k \right] = 0.$$

En conséquence,

$$J^*(f^n, b^n) - J^*(f^n, b^{n+1}) = \frac{1}{2} \lambda^2 \sum_k \left(\left[\left(b_x^n \right)_k - \left(b_x^{n+1} \right)_k \right]^2 \left(g_k^n \right)'' \left(c_k^n \right) \right) \\ + \frac{1}{2} \lambda^2 \sum_k \left(\left[\left(b_y^n \right)_k - \left(b_y^{n+1} \right)_k \right]^2 \left(h_k^n \right)'' \left(d_k^n \right) \right) \quad (8)$$

On a $\left(g_k^n \right)''(b) = \psi''(b)$ et $\left(h_k^n \right)''(b) = \psi''(b)$. On sait par ailleurs (cf. annexe B) que l'on a $\psi''(b) = -1/\theta''(t)$. Un calcul rapide montre que:

$$-\theta''(t^2) = \frac{1}{2t^2} \left(\frac{\varphi'(t)}{2t} - \frac{\varphi''(t)}{2} \right)$$

Cette expression est bornée partout sauf peut-être en 0. Un développement de Taylor en 0 permet d'écrire:

$$\frac{1}{2t^2} \left(\frac{\varphi'(t)}{2t} - \frac{\varphi''(t)}{2} \right) = \frac{1}{4t^2} \varphi'(0) - \frac{1}{8t} \varphi'''(0) - \frac{1}{12} \varphi^{(4)}(0) + \varepsilon(t)$$

On sait qu'en 0, φ' et φ''' tendent vers 0. On voit donc que l'on peut majorer $-\theta''(t^2)$ pour tout t , ce qui revient à minorer $\psi''(b)$. On peut donc dire qu'il existe $c > 0$ tel que:

$$\left(g_n^k \right)''(b) \geq c \quad \forall t.$$

(idem pour $\left(h_n^k \right)''$). On a donc finalement:

$$J^*(f^n, b^n) - J^*(f^n, b^{n+1}) \geq \frac{1}{2} \lambda^2 c \left\| b_x^n - b_x^{n+1} \right\|^2 + \frac{1}{2} \lambda^2 c \left\| b_y^n - b_y^{n+1} \right\|^2$$

et d'après (6):

$$J_{n-1} - J_n \geq \frac{1}{2} \lambda^2 c \|b_x^n - b_x^{n+1}\|^2 + \frac{1}{2} \lambda^2 c \|b_y^n - b_y^{n+1}\|^2$$

Comme la suite J_n est convergente, on en déduit que $(b_x^{n+1} - b_x^n) \xrightarrow{n \rightarrow +\infty} 0$ et $(b_y^{n+1} - b_y^n) \xrightarrow{n \rightarrow +\infty} 0$. Comme $\varphi'(u)/2u$ est bijective, on a aussi: $(D_x f^n - D_x f^{n-1}) \xrightarrow{n \rightarrow +\infty} 0$ et $(D_y f^n - D_y f^{n-1}) \xrightarrow{n \rightarrow +\infty} 0 \quad \forall k$

• Finalement nous montrons que dans le cas convexe, f^n converge et donc que l'algorithme ARTUR converge vers le minimum global de $J(f)$.

Quand b^n est fixé, $J^*(f, b^n)$ est convexe en f , et le minimum est atteint pour $f = f^n$. Donc, f^n est solution des équations normales:

$$(R^t R + \lambda^2 D_x^t B_x^n D_x + \lambda^2 D_y^t B_y^n D_y) f^n - R^t p = 0 \quad (9)$$

avec $B_x^n = \text{diag} \left\{ (b_x^n)_k, k = 1 \dots N^2 \right\}$ et $(b_x^n)_k = \frac{\varphi' \left[(D_x f^{n-1})_k \right]}{2 (D_x f^{n-1})_k} \quad \forall k$ (la définition pour la

direction y est semblable). D'autre part, lorsque $n \rightarrow +\infty$ nous avons:

$$(R^t R + \lambda^2 D_x^t B_x^0 D_x + \lambda^2 D_y^t B_y^0 D_y) f^0 - R^t p = 0 \quad (10)$$

En soustrayant (10) à (9) et en faisant le produit scalaire avec $(f^n - f^0)$, on obtient:

$$\left| R(f^n - f^0) \right|^2 + \lambda^2 \left\langle B_x^n D_x f^n - B_x^0 D_x f^0, D_x (f^n - f^0) \right\rangle + \lambda^2 \left\langle B_y^n D_y f^n - B_y^0 D_y f^0, D_y (f^n - f^0) \right\rangle = 0$$

Introduisons $B_x^{n+1} D_x f^n$ et $B_y^{n+1} D_y f^n$. Nous obtenons:

$$\begin{aligned} & \left| R(f^n - f^0) \right|^2 + \lambda^2 \left\langle B_x^{n+1} D_x f^n - B_x^0 D_x f^0, D_x (f^n - f^0) \right\rangle \\ & + \lambda^2 \left\langle B_y^{n+1} D_y f^n - B_y^0 D_y f^0, D_y (f^n - f^0) \right\rangle \\ & + \lambda^2 \left\langle (B_x^n - B_x^{n+1}) D_x f^n, D_x (f^n - f^0) \right\rangle \\ & + \lambda^2 \left\langle (B_y^n - B_y^{n+1}) D_y f^n, D_y (f^n - f^0) \right\rangle = 0 \end{aligned} \quad (11)$$

Nous rappelons que:

$$\left(b_x^{n+1}\right)_k \left(D_x f^n\right)_k = \frac{1}{2} \varphi' \left[\left(D_x f^n\right)_k \right] \text{ et } \left(b_x^0\right)_k \left(D_x f^0\right)_k = \frac{1}{2} \varphi' \left[\left(D_x f^0\right)_k \right]$$

(les mêmes résultats sont valables pour la direction y) D'autre part, φ est convexe par hypothèse et vérifie donc: $[\varphi'(u) - \varphi'(v)](u - v) \geq 0$. En appliquant ce résultat avec $u = \left(D_x f^n\right)_k$ et $v = \left(D_x f^0\right)_k$, on voit que le produit scalaire:

$$\left\langle \left(B_x^{n+1}\right) \left(D_x f^n\right) - \left(B_x^0\right) \left(D_x f^0\right), \left(D_x f^n\right) - \left(D_x f^0\right) \right\rangle$$

est positif car chacun de ses termes est positif. Le même raisonnement est valable pour la direction y et on obtient:

$$\left\langle \left(B_y^{n+1}\right) \left(D_y f^n\right) - \left(B_y^0\right) \left(D_y f^0\right), \left(D_y f^n\right) - \left(D_y f^0\right) \right\rangle \geq 0$$

En conséquence, d'après (11) on a:

$$\begin{aligned} \left| R \left(f^n - f^0 \right) \right|^2 - \lambda^2 \left\langle \left(B_x^{n+1} - B_x^n \right) D_x f^n, D_x \left(f^n - f^0 \right) \right\rangle \\ - \lambda^2 \left\langle \left(B_y^{n+1} - B_y^n \right) D_y f^n, D_y \left(f^n - f^0 \right) \right\rangle \leq 0 \end{aligned}$$

ou encore:

$$\left| R \left(f^n - f^0 \right) \right|^2 \leq \lambda^2 \left\| B_x^{n+1} - B_x^n \right\| \cdot \left[\left\| D_x f^n \right\| \cdot \left\| D_x \left(f^n - f^0 \right) \right\| \right] + \lambda^2 \left\| B_y^{n+1} - B_y^n \right\| \cdot \left[\left\| D_y f^n \right\| \cdot \left\| D_y \left(f^n - f^0 \right) \right\| \right]$$

On peut montrer que la norme du gradient est toujours majorée par une constante M. On a donc:

$$\left| R \left(f^n - f^0 \right) \right|^2 \leq 2 \lambda^2 M^2 \left[\left\| B_x^{n+1} - B_x^n \right\| + \left\| B_y^{n+1} - B_y^n \right\| \right]$$

et nous concluons que $R \left(f^n - f^0 \right) \xrightarrow[n \rightarrow +\infty]{} 0$ car $\left(b_x^{n+1} - b_x^n \right) \xrightarrow[n \rightarrow +\infty]{} 0$ et $\left(b_y^{n+1} - b_y^n \right) \xrightarrow[n \rightarrow +\infty]{} 0$. Si R est inversible, on a donc $f^n \xrightarrow[n \rightarrow +\infty]{} f^0$.

• Même si R n'est pas inversible, on a également: $D_x f^n \xrightarrow[n \rightarrow +\infty]{} D_x f^0$ et $D_y f^n \xrightarrow[n \rightarrow +\infty]{} D_y f^0$. En effet, d'après (11) on a:

$$\left\langle B_x^{n+1} D_x f^n - B_x^0 D_x f^0, D_x \left(f^n - f^0 \right) \right\rangle \rightarrow 0 \quad (12)$$

Nous allons minorer cette expression, qui peut aussi s'écrire:

$$\sum_k \left\{ \left[\varphi'(D_x f^n)_k \right] - \left[\varphi'(D_x f^0)_k \right] \right\} \left\{ (D_x f^n)_k - (D_x f^0)_k \right\} \rightarrow 0$$

Or on sait (formule de Taylor) qu'il existe un $e_n^k \in \left] (D_x f^0)_k, (D_x f^n)_k \right[$ tel que:

$$\left\{ \left[\varphi'(D_x f^n)_k \right] - \left[\varphi'(D_x f^0)_k \right] \right\} = \left\{ (D_x f^n)_k - (D_x f^0)_k \right\} \varphi''(e_n^k)$$

Soit:

$$\left\{ \left[\varphi'(D_x f^n)_k \right] - \left[\varphi'(D_x f^0)_k \right] \right\} \left\{ (D_x f^n)_k - (D_x f^0)_k \right\} = \left\{ (D_x f^n)_k - (D_x f^0)_k \right\}^2 \varphi''(e_n^k)$$

En supposant que le gradient reste majoré par une constante M, on sait que φ'' est minorée par un certain $A > 0$. Ceci nous permet d'écrire:

$$\sum_k \left\{ \left[\varphi'(D_x f^n)_k \right] - \left[\varphi'(D_x f^0)_k \right] \right\} \cdot \left\{ (D_x f^n)_k - (D_x f^0)_k \right\} \geq A \left\| D_x (f^n - f^0) \right\|^2$$

D'après (12), on voit donc que $\left\| D_x (f^n - f^0) \right\|^2 \xrightarrow{n \rightarrow +\infty} 0$, ce qui nous permet de conclure que $D_x f^n \xrightarrow{n \rightarrow +\infty} D_x f^0$. La preuve est semblable pour la direction y.

Références Bibliographiques

- [AMAR94] V. Amar, "Reconstruction tomographique d'images 3D en médecine nucléaire", Rapport de DEA Informatique, Université de Nice-Sophia Antipolis, Juin 1994.
- [ANTO91] M. Antonini, "Transformée en Ondelettes et Compression Numérique des Images", *Thèse de Doctorat en Sciences*, Université de Nice-Sophia Antipolis, 17 Septembre 1991.
- [ANTO92] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding using Wavelet Transform", *IEEE Trans. on Image Processing*, Vol. 1, N°2, pp. 205-220, 1992.
- [AUBE94] G. Aubert, M. Barlaud and P. Charbonnier, "Theorems for Edge-Preserving Regularization in Computed Imaging", *Submitted to IEEE Trans. on Image Processing*, tech. rep., Research Report N° 94-01, I3S, University of Nice-Sophia Antipolis, 1994.
- [AUBE94B] G. Aubert, L. Blanc-Féraud, M. Barlaud and P. Charbonnier, "A deterministic algorithm for edge-preserving computed imaging using Legendre transform", *to appear in Proceedings of the 12th ICPR*, Jerusalem, Israël, October 1994.
- [BARL92] M. Barlaud, L. Blanc-Féraud and P. Charbonnier, "Image coding using multiresolution Markov Random Fields", *Image Processing Algorithms and Techniques III*, SPIE Proceedings, Vol. 1657, pp. 24-35, 1992.
- [BATH93A] M. Bathia, W. C. Karl, A. S. Willsky, "Using Natural Wavelet Bases and Multiscale Stochastic Models for Tomographic Reconstruction", *MIT Tech. Report N°LIDS-P-2196*, Submitted to *IEEE Trans. on Signal Processing*, 1993.
- [BATH93B] M. Bathia, W. C. Karl, A. S. Willsky, "A Fast Filtered Back-Projection Based Multiscale Tomographic Reconstruction Technique from Complete but Possibly Noisy Projection data", *MIT Tech. Report N°LIDS-P-2182*, Submitted to *IEEE Trans. on Signal Processing*, 1993.
- [BERT92] M. Berthod, G. Giraudon, J.P. Stromboni, "Deterministic pseudo-annealing: optimization in Markov-Random-Fields. An application to pixel classification", in *Proc. 2nd ECCV*, pp. 67-71, Gènes Italy, 18-23 Octobre 1992.
- [BESA86] J. Besag, "On the Statistical Analysis of Dirty Pictures", *J. Roy. Statist. Soc.*, N°3, pp. 259-302, 1986.
- [BEYL91] G. Beylkin, R. R. Coiffman and V. Rokhlin, "Fast wavelet transforms and numerical algorithms", *Comm. Pure and Appl. Math.*, 44, pp 141-183, 1991.
- [BLAC91] M. J. Black and P. Anandan, "Robust dynamic motion estimation over time", in

Proc. CVPR'91, pp. 286-302, Hawaï, June 1991.

- [BLAC93] M. J. Black and P. Anandan, "A framework for the robust estimation of optical flow", *4th ICCV*, pp. 231-236, Berlin, May 1993.
- [BLAK87] A. Blake, A. Zisserman, *Visual Reconstruction*, London: The M.I.T. Press Series in artificial intelligence, 1987.
- [BLAN92] L. Blanc-Féraud, M. Barlaud and P. Charbonnier, "Problèmes inverses et multirésolution", *Progress in Wavelet Analysis and Applications*, edited by Y. Meyer and S. Roques, ed. *Frontières, Proceedings of the international conference "Wavelets and applications"*, pp. 387-398, Toulouse, France, June, 1992.
- [BLAN93] L. Blanc-Féraud, M. Barlaud and T. Gaidon, "Motion estimation involving discontinuities in a multiresolution scheme", *Optical Engineering*, Vol. 32, N°7, pp 1475-1482, July, 1993.
- [BLAN94] L. Blanc-Féraud, P. Charbonnier, P. Lobel and M. Barlaud, "A fast tomographic reconstruction algorithm in the 2D wavelet transform domain", *Proc. ICASSP 1994*, Adelaide, Vol. V, pp. 305-308, April 1994.
- [BOUM91] C. Bouman and K. Sauer, "An Edge-preserving Method for Image Reconstruction from Integral Projections", in *Proc. Conf. on Info. Sci. and Syst.*, The Johns Hopkins University, Baltimore, MD, pp 382-387, March, 1991.
- [BOUM93] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation", *IEEE Trans. on Image Processing*, Vol 2, N°3, pp 296-310, July, 1993.
- [BOUT92] F. Bouthemy and F. Heitz, "Champs Markoviens: fondements et application à l'analyse de séquences d'image", in *Journées thématiques Approches Markoviennes en Signal et Image, GDR 134 TDSI du CNRS*, pp. 20-43, Paris, September 1992.
- [BRUN93] J.M. Bruneau, "Résolution de problèmes inverses mal-posés en analyse multirésolution. Application à la restauration d'images et à la stéréovision.", *Thèse de Doctorat en sciences*, Université de Nice-Sophia Antipolis, Octobre 1993.
- [CEA89] J. Cea, *Méthodes numériques*, Support de cours, Ecole Supérieure en Sciences informatiques, Université de Nice-Sophia Antipolis, 1989.
- [CEA] J. Cea, *Optimisation, Théorie et Algorithmes*, Paris: Dunod, 1971.
- [CHAL88] B. Chalmond, "Image Restoration using an Estimated Markov Model", *Signal Processing*, 15, pp. 115-129, 1988.
- [CHAL89] B. Chalmond, "An iterative Gibbsian Technique for Reconstruction of m-ary images", *Pattern Recognition*, Vol. 22, N° 6, pp. 747-761, 1989
- [CHAM93] A. Chambolle, "Analyse mathématique de trois problèmes de reconstruction

visuelle”, *Thèse de doctorat en Sciences*, Université de Paris IX Dauphine, 1993.

- [CHAR92A] P. Charbonnier, M. Barlaud et L. Blanc-Féraud, “*Méthode adaptative de reconstruction tomographique préservant les discontinuités*”, Rapport de synthèse GDR TDSI “Reconstruction 3D en tomographie X”, Juillet 1992.
- [CHAR92B] P. Charbonnier, L. Blanc-Féraud et M. Barlaud, “*An adaptive reconstruction method involving discontinuities*”, rapport interne I3S N°92-61, Novembre 92.
- [CHAR92C] P. Charbonnier, L. Blanc-Féraud, M. Barlaud, Ivan Ltte, Jacques Darcourt and Françoise Lapalus, “*Iterative SPECT reconstruction using a regularization which preserves sharp edges: comparison to classical methods*”, *Proc. 14th EMBS*, Vol. 5, pp. 1823-1824, Paris, November 1992.
- [CHAR92D] P. Charbonnier, L. Blanc-Féraud and M. Barlaud, “*Noisy image restoration using multiresolution Markov random fields*”, *Journal of Visual Communication and Image Representation*, Vol. 3, N° 4, pp. 338-346, December 1992.
- [CHAR93A] P. Charbonnier, L. Blanc-Féraud, M. Barlaud, “*An adaptive reconctruction method involving discontinuities*”, *Proc. ICASSP 1993*, Vol V, pp. 491-494, Minneapolis, April 1993.
- [CHAR93B] P. Charbonnier, L. Blanc-Féraud and M. Barlaud, “*ARTUR: An Adaptive Deterministic Relaxation Algorithm for Edge-Preserving Tomographic Reconstruction*”, *Submitted to IEEE Trans. on Image Processing*, tech. rep., Research Report N° 93-76, I3S, University of Nice-Sophia Antipolis, 1993.
- [CHAR93C] P. Charbonnier, L. Blanc-Féraud et M. Barlaud, “*A Fast Iterative Tomographic Reconstruction Algorithm*”, *Proceedings of the heighth workshop on Image and Multidimensional Signal Processing*, pp. 104-105, Cannes, September 1993.
- [CHAR94A] P. Charbonnier, G. Aubert, L. Blanc-Féraud et M. Barlaud, “*Two deterministic half-quadratic regularization algorithms for computed imaging*”, *to appera in Proceedings of the first IEEE ICIP*, Austin, USA, November 1994.
- [CIAR] P.G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Paris: Masson, 1994.
- [COHE90] A. Cohen, I. Daubechies, J. C. Feauveau, “*Biorthogonal Bases of Compactly Supported Wavelets*”, AT&T Bell laboratories, Tech. Report, N°TM 11217-90052907, 1990.
- [DARC92] J. Darcourt, “*Méthodes de reconstruction tomographique à partir des projections: application à l'imagerie médicale*”, *Thèse de doctorat en sciences*, Université de Nice-Sophia Antipolis, 1992.
- [DEMO89] G. Demoment, “*Image Reconstruction and Restoration: Overview of Common Estimation Structures and Problems*”, *IEEE Trans. on ASSP*, Vol. 37, N°12, pp. 2024-2036, December 1989.

- [DEMP77] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *J. Roy. Statist. Soc.*, Vol. 39, pp. 1-38, 1977.
- [DINT90] J.M. Dinten, "Tomographie à partir d'un nombre limité de projections: régularisation par des champs Markoviens", *Thèse de doctorat en sciences*, Université de Paris-Sud, Janvier 1990.
- [FLET] R. Fletcher, *Practical Methods of Optimization*, 2nd edition, New York: Wiley, 1987.
- [GEIG90] D. Geiger and F. Girosi, "Parallel and Deterministic Algorithms from MRF's: Surface Reconstruction and Integration", in *Proc. First ECCV*, Antibes, France, Lect. Notes in Comp. Science 427, pp.89-97, 1990.
- [GEIG91] D. Geiger and F. Girosi, "Parallel and Deterministic Algorithms from MRF's: Surface Reconstruction", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-13, N°5, pp. 401-412, May 1991.
- [GEMA84] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, pp. 721-741, Nov. 1984
- [GEMA85] S. Geman and D.E. McClure, "Bayesian image analysis: an application to single photon emission tomography", in *Proc. Statist. Comput. Sect.*, Amer. Statist. Assoc., Washington, DC, pp. 12-18, 1985.
- [GEMA92] S. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol PAMI-14, N°3, pp. 367-383, March 1992.
- [GEMA93] D. Geman and C. Yang, "Nonlinear Image Recovery with Half-Quadratic Regularization and FFTs", *preprint*, March 1993
- [GREE90A] P. J. Green, "Bayesian reconstructions from emission tomography data using a modified EM algorithm", *IEEE Trans. Med. Imaging*, Vol. MI-9, N°1, pp. 84-93, March 1990.
- [GREE90B] P. J. Green, "On use of the EM algorithm for penalized likelihood estimation", *J. Roy. Statist. Soc. B*, 52, N° 3., pp. 443-452, 1990.
- [GUYO] X. Guyon, *Champs aléatoires sur un réseau. Modélisation, statistique et applications*, Paris: Masson, 1993.
- [HANS83] K.M. Hanson and G.W. Wecksung, "Bayesian approach to limited angle reconstruction in computed tomography", *J. Opt. Soc. Amer.*, Vol. 73, N° 11, pp. 1501-1509, November 1983.
- [HEBE89] T. Hebert and R. Leahy, "A generalized EM algorithm for 3-D Bayesian

- reconstruction from Poisson data using Gibbs priors”, *IEEE Trans. Med. Imaging*, Vol. MI-8, N°2, pp. 194-202, June 1989.
- [HORN] B.L.P. Horn, *Robot Vision*, the MIT Press, 1986.
- [HURW75] H. Hurwitz Jr., "Entropy reduction in Bayesian analysis of measurements", *Phys. Rev., A*, Vol. 12, pp. 678-706, 1975.
- [JAIN] A.K. Jain, *The Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
- [JENG91] F. C. Jeng and J. W. Woods, "Compound Gauss-Markov Random Fields for Image Estimation”, *IEEE Trans. on Signal Processing*, Vol 39, N°3, pp. 683-697, March 1991.
- [KAK] S. Haykin, J.H. Justice, N.L. Owsley, J.L. Yen, A.C. Kak, *Array Signal Processing*, Prentice Hall, 1985.
- [KATO93] Z. Kato, J. Zerubia, M. Berthod, "Bayesian Image Classification using Markov Random Fields”, Maximum Entropy and Bayesian Methods, A. Mohammad-Djafari and G. Demoment editors, pp. 375-381, Kluwer Academic Publishers, 1993.
- [KOUL91] P.M. Koulibaly, "Reconstruction d'images tomographique par des méthodes itératives”, Rapport de DEA Imagerie et Sciences de l'Univers, Université de Nice-Sophia Antipolis, 1991.
- [KOUL94] P.M. Koulibaly, P. Charbonnier, M. Barlaud, J. Darcourt and L. Blanc-Féraud, "EM-MAP algorithms versus ARTUR: theoretical and practical comparisons”, to appear in *proc. of SPIE Mathematical Methods in Medical Imaging III*, San Diego USA, July 1994.
- [KOUL95] P.M. Koulibaly, "Reconstruction d'images tomographiques par des méthodes itératives régularisées avec correction des paramètres physiques”, *Thèse de doctorat*, Université de Nice-Sophia Antipolis, 1995.
- [LALU92] D. S. Lalush, B. M. W. Tsui, "Simulation evaluation of Gibbs prior distributions for use in maximum a posteriori SPECT reconstruction”, *IEEE Trans. Med. Imaging*, Vol. MI-11, N°2, pp. 267-275, June 1992.
- [LANG84] K. Lange, R. Carson, "EM reconstruction algorithms for emission and transmission tomography”, *J. Comput. Assist. Tomogr.*, Vol. 8, N°2, pp 306-316, April 1984.
- [LANG90] K. Lange, "Convergence of EM Image Reconstruction Algorithms with Gibbs Smoothing”, *IEEE Trans. Med. Imaging*, Vol. MI-9, N°4, December 1990.
- [LAUR92] I. L. Ltte, "Reconstruction tomographique en imagerie médicale par des méthodes itératives utilisant la régularisation: construction de la matrice de Radon et comparaison de méthodes”, Rapport de DEA Imagerie et Sciences de l'Univers,

Université de Nice-Sophia Antipolis, 1991.

- [LAUR94] I. L. Lte, “Reconstruction tomographique 3D en imagerie médicale”, *Rapport d’avancement des travaux N°1*, Région PACA, 1994.
- [LAZA95] L. Lazaroaia, “Sur quelques problèmes d’équations aux dérivées partielles non linéaires et du calcul des variations en reconstruction et en restauration d’images”, *Thèse de doctorat*, Université de Nice-Sophia Antipolis, 1995.
- [LEVI87] E. Levitan and G.T. Herman, “A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography”, *IEEE Trans. Med. Imaging*, Vol. MI-6, pp. 185-192, 1987.
- [LEWI86] R.M. Lewitt and G. Muehllehner, “Accelerated Iterative Reconstruction for Positron Emission Tomography Based on the EM Algorithm for Maximum Likelihood Estimation”, *IEEE Trans. Med. Imaging*, Vol. MI-5, N° 1, pp. 16-22, March 1986.
- [LIAN88] Z. Liang, “Statistical models for a priori information for image processing”, *SPIE Medical Imaging II*, Vol. 914, pp. 677-683, 1988.
- [LOBE93] P. Lobel, L. Blanc-Féraud, P. Charbonnier, M. Barlaud, “Reconstruction Tomographique dans le Domaine de la Transformée en Ondelettes bidimensionnelle”, I3S CNRS-URA 1376, Tech. Report N° 93-60, October 1993.
- [MALL89] S. Mallat, “A theory for Multiscale Signal Decomposition: The Wavelet Representation”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, N°7, pp. 674-693, 1989.
- [MARC92] R. March, “Visual Reconstruction with Discontinuities using Variational methods”, *Image and Vision Computing*, Vol 10, N°1, pp. 30-38, 1992.
- [MARR84] J.L. Marroquin, “Surface reconstruction preserving discontinuities”, Artificial Intell. Lab, Memo N°792, Massachusetts Institute of Technology, August 1984.
- [MARR85] J.L. Marroquin, “Probabilistic Solution of Inverse Problems”, Phd Thesis, Massachusetts Institute of Technology, September 1985.
- [MEYE] Y. Meyer, *Ondelettes et Opérateurs I: Ondelettes*, Paris:Hermann, 1990.
- [NATT] F. Natterer, *The mathematics of computerized tomography*, New York: Wiley, 1986.
- [NIKO94] M. Nikolova, A. Mohammad-Djafari and J. Idier, “Inversion of Large-support ill-conditioned linear operators using a Markov model with a line process”, *Proc. ICASSP 1994*, Adelaide, Vol. V, pp. 357-360, April 1994.
- [NOUM93] R. Noumeir, G.E. Mailloux and R. Lemieux, “Accelerated iterative reconstruction based on the maximum *a posteriori* expectation maximization”, *Proc. ICASSP 1993*, Vol V, pp. 469-472, Minneapolis 1993.

- [NUMC] W. H. Press, S.A. Teulosky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C, The Art of Scientific Computing*, 2nd edition, Cambridge University Press, 1992.
- [PELL94] C. Pellot, A. Herment, M. Sigelle, P. Horain, H. Maître and P. Peronneau, "A 3D reconstruction of vascular structures from two X-rays angiograms using an adapted simulated annealing algorithm", *IEEE Trans. Med. Imaging*, Vol. MI-6, N° 1, pp. 48-60, March 1994.
- [PENT91] A. P. Pentland, "Spatial and Temporal Surface Interpolation Using Wavelet Bases", *Geometric Methods in Computer Vision*, SPIE Vol. 1570, pp. 43-62, 1991.
- [POGG85] T. Poggio, V. Torre and C. Koch, "Computational Vision and Regularization theory", *Nature*, Vol. 317, pp. 314-319, September 1985.
- [POUG88] J. Pouget, *Probabilités*, Support de cours, Ecole Supérieure en Sciences informatiques, Université de Nice-Sophia Antipolis, 1988.
- [PRIN93] J. L. Prince and A.S. Willsky, "Hierarchical reconstruction using geometry and sinogram restoration", *IEEE Trans. on Image Processing*, Vol 2, N°3, pp 401-416, July, 1993.
- [ROCK] R. Rockafellar, *Convex Analysis*, Princeton: Princeton University Press, 1970.
- [ROUC94] B. Rouchouze, L. Blanc-Féraud et M. Barlaud, "Méthodes pel-récurrentes et d'appariement de blocs pour l'estimation de mouvement dans une séquence d'image", *rapport interne I3S*, 1994.
- [SCHU93] R. R. Schultz and R. L. Stevenson, "Stochastic Modeling and Estimation of Multispectral Image Data", *Submitted to IEEE Trans. on Image Processing*, preprint, May 1993.
- [SHEP74] L. A. Shepp and B.F. Logan, "The Fourier reconstruction of a head section", *IEEE Trans. on Nuclear Science*, Vol. NS-21, pp. 21-43, 1974.
- [SHEP82] L.A. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography", *IEEE Trans. Med. Imaging*, Vol. MI-1, N°2, pp. 113-122, October 1982.
- [SMIT92] M. F. Smith, C. E. Floyd Jr., R. J. Jaszczak and R. E. Coleman, "Reconstruction of SPECT images using generalized matrix inverses", *IEEE Trans. Med. Imaging*, Vol. MI-11, N° 2, pp. 165-175, June 1992.
- [STRA] G. Strang, *Linear algebra and its applications*, Second edition, New York: Academic Press, 1980.
- [TERZ86] D. Terzopoulos, "Regularization of Inverse Problems Involving Discontinuities", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-8, N°4, pp. 413-424, July 1986.

- [TIKH63] A.N. Tikhonov, "Regularization of incorrectly posed problems", *Sov. Math. Dokl.*, Vol. 4, pp. 1624-1627, 1963.
- [TITT85] D. Titterington, "General Structure of Regularization Procedures in Image Reconstruction", *Astron. Astrophys.*, Vol. 144, pp. 381-387, 1985.
- [WAKU93] J. Waku, "Ondelettes et applications en imagerie et en calcul de surfaces", *Thèse de doctorat en sciences*, Université Joseph Fourier, Grenoble, 1993.
- [ZERU90] J. Zerubia et C. Graffigne, "Détection de contours et restauration d'image par des algorithmes déterministes de relaxation", Rapport Segmentation du GDR TDSI, Juillet 1990, à paraître chez Masson en 1995.
- [ZERU91] J. Zerubia et F. Ployette, "Détection de contours et lissage d'image par deux algorithmes déterministes de relaxation. Mise en œuvre sur la machine à connexions CM2", *Traitement du Signal*, Vol. 8 N°3, pp. 165-179, 1991.
- [ZERU92] J. Zerubia and Marc Berthod, "Markovian modelization in computer vision. Different techniques of optimization", in *Journées thématiques Approches Markoviennes en Signal et Image, GDR 134 TDSI du CNRS*, pp. 167-184, Paris, September 1992.
- [ZERU93] J. Zerubia, and R. Chellappa, "Mean field annealing using compound Gauss-Markov random fields for edge detection and image estimation", *IEEE Trans. Neural Networks*, Vol. 4, N°4, pp. 703-709, July 1993.

Liste des figures et tableaux

Chapitre I

Figure 1-1 : De l'image réelle à l'image observée.....	11
Figure 1-2 : Voisinages d'ordre 1 et 2 (d'après [GEMA86]).....	17
Figure 1-3 : Cliques associées aux voisinages d'ordre 1 (a) et d'ordre 2 (a) et (b)	18
Figure 1-4 : Cliques utilisées pour les voisinages d'ordres 1 (a) et 2 (b).....	19
Figure 1-5 : La dynamique de Metropolis permet de “sortir” des minima locaux (d'après [BLAKE])	23
Figure 1-6 : Tirage aléatoire selon une loi, par inversion de la fonction de répartition.....	24
Figure 1-7 : Principe de l'algorithme GNC (d'après [BLAKE])	26

Chapitre II

Figure 2-1 : Localisation du processus de ligne.....	32
Figure 2-2 : Valeurs de potentiel de régularisation de b définies dans [GEMA84] (en gras) et [MARR84] (en italique).....	34
Figure 2-3 : Fonction quadratique (pointillés) et quadratique tronquée (trait plein)	35
Figure 2-4 : Quadratique tronquée (trait pointillé) et fonction de Geman et McClure (trait plein)	36
Figure 2-5 : Fonction de Hebert et Leahy	38
Figure 2-6 : Fonction $ u ^\alpha$ pour: $\alpha = 1$ (a); $\alpha = 1,2$ (b); $\alpha = 1,4$ (c); $\alpha = 1,6$ (d); $\alpha = 1,8$ (e).....	39

Figure 2-7 : Fonction de Huber (a), fonction logcosh (b) et fonction des surfaces minimales (c) 40

Figure 2-8 : Courbe $z=f(x)$ 41

Figure 2-9 : Diversité des fonctions de potentiel censées correspondre à une régularisation avec prise en compte des discontinuités.....43

Figure 2-10 : Pondérations du Laplacien au voisinage du site (i,j).....45

Figure 2-11 : Fonction de pondération correspondant à la quadratique (a), la fonction de Huber (b) et la quadratique tronquée (c).46

Tableau 2-1 :Fonctions de potentiel et fonctions de pondération.....47

Figure 2-12 : Fonction de pondération correspondant aux fonctions de Geman et McClure (a), d’Hebert et Leahy (b), des surfaces minimales (c) et de Green (d).48

Figure 2-13 : Fonction de pondération correspondant à la fonction $|u|^\alpha$ préconisée par Bouman et Sauer pour: $\alpha = 1$ (a); $\alpha = 1,2$ (b); $\alpha = 1,4$ (c); $\alpha = 1,6$ (d); $\alpha = 1,8$ (e).49

Tableau 2-2 : Tableau récapitulatif des fonctions de potentiel et fonctions de pondération utilisées50

Tableau 2-3 : Expression de $\psi(b)$ pour les potentiels utilisés.....56

Tableau 2-4 : Du non-quadratique au semi-quadratique58

Tableau 2-5 : Expression de b pour f fixé59

Figure 2-14 : La régularisation semi-quadratique60

Chapitre III

Figure 3-1 : Principe des minimisations alternées64

Tableau 3-1 : Calcul de b^{n+1} (f^n fixé)66

Tableau 3-2 : Calcul de f^{n+1} (b^{n+1} fixé)67

Figure 3-2 : Forme générale des algorithmes ARTUR et LEGEND.....68

Figure 3-3 :	Interprétation des algorithmes ARTUR et LEGEND.....	69
Figure 3-4 :	Algorithme SOR modifié pour le calcul de f^{n+1}	75
Figure 3-5 :	Algorithme de gradient pour le calcul de f^{n+1}	76
Tableau 3-3 :	Tableau récapitulatif des fonctions de pondération utilisées en pratique.....	78
Figure 3-6 :	Fonctions de pondérations utilisées en pratique, correspondant à: (a) φ_{GM} , (b) φ_{HL} , (c) φ_{HS}	78
Figure 3-7:	Distance entre pixels voisins.....	81
Figure 3-8 :	Fonction $\varphi_{BZ}^{(p)}$ pour plusieurs valeurs de p ($c^* = 1/4$).....	85
Figure 3-9 :	Fonction φ_{β} pour plusieurs valeurs de β	88
Tableau 3-4 :	Classification des principaux algorithmes déterministes dans le cas de la vraisemblance Gaussienne.....	91
Figure 3-10 :	Algorithme MAP-EM-OSL	95
Figure 3-11 :	Algorithme MAP-EM utilisé à chaque étape pour le calcul de \hat{f}^{n+1}	97
Tableau 3-5 :	Exemples d'algorithmes déterministes pour la régularisation semi-quadratique (avec vraisemblance Gaussienne).	101
Figure 3-12 :	A gauche: trois étapes d'un schéma implicite: à chaque étape, à b fixé, on minimise un critère convexe. A droite: cas d'un schéma explicite: à chaque étape, à b fixé, on fait un pas de descente.	102

Chapitre IV

Figure 4-1 :	Tomographie par émission: acquisition des données.....	106
Figure 4-2 :	Géométrie de l'acquisition des projections (d'après [JAIN])	106
Figure 4-3 :	Discretisation de la distribution d'activité du traceur (N=16)	108
Figure 4-4 :	Projection d'une image 4x4 pour un angle de $\pi/4$. En grisé: raie de sommation du premier pixel	108

Figure 4-5 : Modèle du disque concave	109
Figure 4-6 : Projection d'un pixel sur le détecteur.....	110
Tableau 4-1 : Taille-mémoire de R en fonction de N (stockage en flottant sur 4 octets)...	110
Tableau 4-2 : Etude de la quantification de R	111
Tableau 4-3 : Etude de la quantification de $R'R$	111
Figure 4-7 : Noyaux de projection du dixel 5, pour les angles :(a) 0, (b) $\pi/4$, (c) $\pi/2$, (d) $3\pi/4$. Les pixels participant à la raie de projection n°5 sont en foncé	112
Figure 4-8 : Spectre des valeurs singulières de R pour $N=16$ (échelle logarithmique).....	114
Figure 4-9 : Conditionnement c pour N compris entre 1 et 24 (d'après [LAUR94]).....	114
Tableau 4-4 : Conditionnement en fonction de N : étude de l'effet de la quantification	115
Figure 4-10 : Objet Nice (d'après [DARC92])	116
Figure 4-11 : Fantôme de Jaszczak et ses projections acquises au Centre Antoine Lacassagne de Nice	117
Figure 4-12 : Données réelles: projections d'un cerveau acquises au Centre Antoine Lacassagne de Nice	117
Figure 4-13 : Evolution de la solution au fil des étapes (niveaux de gris, inversée).....	119
Figure 4-14 : Evolution du processus de ligne vertical au fil des étapes (niveaux de gris) 121	
Figure 4-15 : Evolution du processus de ligne horizontal au fil des étapes (niveaux de gris) 122	
Figure 4-16 : Evolution des variables auxiliaires diagonales au fil des étapes (niveaux de gris) 123	
Figure 4-17 : Evolution de la solution entre deux itérations successives (à gauche), entre deux étapes successives (à droite).....	124
Figure 4-18 : Evolution de la RMS de la solution (à gauche) et des processus de ligne (à droite) au fil des étapes	125

Figure 4-19 : Evolution des énergies (à gauche) et de la NMSE par rapport à la solution exacte (à droite) au fil des itérations	126
Figure 4-20 : Solutions obtenues pour différentes valeurs des paramètres.....	128
Tableau 4-5 : Valeurs des paramètres utilisés	129
Figure 4-21 : Objet Nice (a) et images reconstruites avec différentes fonctions de potentiel. On utilise en (b) le potentiel quadratique (Tikhonov); en (c) la fonction φ_{GR} ; en (d) la fonction φ_{HL} et en (e) φ_{GM}	130
Figure 4-22 : Coupes de l'objet Nice (en pointillés) et des images reconstruites (traits pleins). On utilise: (a) le potentiel quadratique (Tikhonov); (b), le potentiel convexe φ_{GR} ; (c) φ_{HL} et (d) φ_{GM} (potentiels non convexes).....	131
Figure 4-23 : Comportement de la fonction log cosh dans le cas d'un échelon	132
Figure 4-24 : Profils des images reconstruites pour deux conditions initiales différentes avec: le potentiel non convexe $\varphi = \varphi_{HL}$ (a) et le potentiel convexe $\varphi = \varphi_{GR}$ (b).....	133
Tableau 4-6 : Résultats obtenus à partir de différentes conditions initiales. Nb étapes, Nb itérations: nombre d'étapes et nombre total d'itérations effectuées par l'algorithme; SNR/Nice: rapport signal à bruit (en variance) par rapport à l'image originale.....	134
Figure 4-25 : Conditions initiales utilisées	135
Figure 4-26 : Résultats obtenus avec les différentes conditions initiales (table de couleurs: niveaux de gris, inversée).....	135
Figure 4-27 : Résultat obtenu avec 4 voisins (4V) et 8 voisins (8V). Au centre: objet Nice (niveaux de gris, inversée). Fonction utilisée: φ_{GM}	136
Tableau 4-7 : SNR par rapport à l'image originale avec différents potentiels, et avec les deux types de voisinage utilisés.....	137
Figure 4-28 : Fantôme de Jaszczak: reconstructions avec ARTUR (en haut). On utilise les potentiels φ_{GR} (a), φ_{HL} (b) et φ_{GM} (c). En bas: reconstruction avec le MAP-EM-OSL pour $\varphi = \varphi_{GM}$ (d) et l'algorithme de filtrage-rétroprojection (e) utilisé couramment à l'hôpital (niveau de gris).	138

Figure 4-29: Données réelles: reconstruction d'une coupe de cerveau. A gauche: filtrage-rétroprojection, à droite ARTUR avec φ_{HL} (niveau de gris, inversée).138

Figure 4-30 : Reconstruction avec LEGEND: évolution de la solution au fil des étapes. En bas à droite: solutions finales obtenues avec LEGEND et ARTUR. On utilise le potentiel convexe φ_{GR} 140

Figure 4-31 : Reconstruction avec LEGEND: évolution de $\varphi'(u)/2u$ au fil des étapes (gradient pris dans la direction verticale). On utilise le potentiel convexe φ_{GR}141

Figure 4-32 : Evolution des énergies (à gauche) et de la NMSE par rapport à la solution exacte (à droite) au fil des itérations, pour ARTUR et LEGEND dans les mêmes conditions expérimentales141

Figure 4-8 : Etude de l'influence du seuil d'arrêt entre étapes sur le nombre d'étapes effectuées et la qualité de reconstruction, pour les deux algorithmes ($\varphi = \varphi_{GR}$)142

Figure 4-33 : Reconstruction avec ARTUR (à gauche) et LEGEND (à droite) avec les mêmes paramètres, en utilisant le potentiel non-convexe φ_{GM} 143

Tableau 4-9 : Effet de la factorisation préalable de la matrice A sur le temps de calcul nécessaire à la reconstruction.....144

Figure 4-34 : Résultat de reconstruction obtenu avec MOISE 4 voisins (à gauche) et objet Nice (à droite)145

Figure 4-35 : Un essai d'algorithme ARTUR explicite. Courbe de la NMSE par rapport à la solution exacte en fonction des itérations pour ARTUR et ARTUR explicite...146

Chapitre V

Figure 5-1 : Analyse-synthèse d'un signal 1D sur un niveau151

Figure 5-2 : Transformée BCR de la matrice B sur 3 niveaux de résolution. En noir: valeurs supérieures à 4‰ du maximum, en blanc: valeurs inférieures à ce seuil (représentant 92,8% de la matrice).....155

Figure 5-3 : Transformée “2D” de la matrice B sur 3 niveaux de résolution. En noir: valeurs supérieures à 4‰ du maximum, en blanc: valeurs inférieures à ce seuil (représentant 90,7% de la matrice)	157
Figure 5-4 : Matrice $R^t R - 5\Delta$ ($N=16$). Valeurs faibles en blanc, valeurs importantes en noir	159
Figure 5-5 : Transformée BCR de la matrice $R^t R - 5\Delta$, seuillée à 4‰ (62,3% de zéros introduits dans la matrice). L’image est binarisée pour l’affichage	160
Figure 5-6 : Transformée “2D” de la matrice $R^t R - 5\Delta$, seuillée à 4‰ (91,2% de zéros introduits dans la matrice). L’image est binarisée pour l’affichage	160
Figure 5-7 : Nombre de zéros introduits dans la matrice en fonction du seuil	161
Figure 5-8 : Temps de reconstruction en fonction du nombre de zéros dans la matrice....	161
Figure 5-9 : Qualité de la reconstruction en fonction du nombre de zéros dans la matrice	162
Figure 5-10 : Forme générale des algorithmes ARTUR et LEGEND	163
Figure 5-11 : Forme générale de l’algorithme ARTUR avec résolution des équations normales dans le domaine de la TO.....	164
Figure 5-12 : Forme générale de l’algorithme LEGEND avec résolution des équations normales dans le domaine de la TO.....	165

Résumé: Cette thèse est consacrée à la reconstruction d'image régularisée. Nous nous intéressons au cas où l'image à reconstruire est formée de zones homogènes séparées par des bords francs. C'est la *régularisation avec prise en compte des discontinuités*. L'image reconstruite est obtenue par minimisation d'une fonctionnelle d'énergie, ou critère.

Dans un premier temps, nous proposons des conditions suffisantes pour définir un critère assurant la prise en compte des discontinuités. Nous montrons ensuite que sous ces conditions, il est possible de transformer le critère en introduisant une variable auxiliaire interagissant avec la variable principale, soit de manière additive, soit de manière multiplicative. Le rôle de la variable auxiliaire est double: marquer les discontinuités et faciliter l'optimisation du critère. Le critère augmenté est quadratique en la variable principale lorsqu'on fixe la variable auxiliaire et convexe par rapport à la variable auxiliaire lorsqu'on fixe la variable principale. Dans ce dernier cas, nous donnons de plus l'expression littérale de la valeur optimale de la variable auxiliaire.

Dans un second temps, nous exploitons d'un point de vue algorithmique les propriétés de cette *régularisation semi-quadratique*. La stratégie d'optimisation adoptée consiste à minimiser alternativement le critère par rapport à la variable principale et par rapport à la variable auxiliaire. Ceci nous conduit à deux *algorithmes à introduction progressive de discontinuités*: ARTUR et LEGEND. Nous proposons une étude théorique et pratique de ces algorithmes. Nous appliquons les algorithmes à la reconstruction tomographique.

ARTUR et LEGEND réalisent un bon compromis entre temps de calcul et qualité d'image reconstruite. Néanmoins, ils impliquent la résolution de systèmes linéaires de grandes tailles, lourds à manipuler d'un point de vue informatique. Nous proposons, à titre de perspective, une technique permettant de limiter la quantité d'information nécessaire à la reconstruction et, par là même, d'envisager des dimensions d'image plus importantes.

Mots-clefs: Régularisation, Champs de Markov, Discontinuités, Fonctions de potentiel, Régularisation semi-quadratique, Processus de ligne, Minimisations alternées, Algorithmes de reconstruction déterministes, Reconstruction tomographique, Ondelettes.

Abstract: This thesis deals with regularized image reconstruction. We seek images consisting of smooth regions separated by sharp edges, or discontinuities. This is called *edge-preserving regularization*. The reconstructed image is found by minimizing an objective function, which we call the criterion.

Firstly, we propose sufficient conditions for the design of an edge-preserving objective function. Then, we show that, under these conditions, it is possible to transform the criterion by introducing an auxiliary variable, which interacts with the main variable either in an additive or in a multiplicative way. The role of the auxiliary variable is twofold: it marks discontinuities and facilitates the optimization of the criterion. The augmented objective function is quadratic with respect to the main variable when the auxiliary variable is fixed, and convex with respect to the auxiliary variable when the main variable is fixed. Moreover, in the last case, we give the expression of the optimal value of the auxiliary variable.

Secondly, we exploit the properties of such a *half-quadratic regularization* to design new reconstruction algorithms. Our optimization strategy is based on alternated minimizations with respect to the auxiliary and the main variable. This leads to two *progressive discontinuity introduction* algorithms: ARTUR and LEGEND. We propose a theoretical and practical study of these algorithms. We apply them to tomographic reconstruction.

ARTUR and LEGEND offer a good trade-off between computation time and image quality. However, huge linear systems have to be solved. We propose, as a perspective, a means of reducing the volume of information involved for reconstruction. Thereby, greater image dimensions can be envisioned.

Keywords: Regularization, Markov random fields, Discontinuities, Potential functions, Half-quadratic regularization, Line process, Alternate minimizations, Deterministic reconstruction algorithms, Tomographic reconstruction, Wavelets.