



Université Sétif 1 Ferhat Abbas
Faculté des Sciences
Département de Mathématiques

جامعة سطيف 1 فرحات عباس
كلية العلوم
قسم الرياضيات

Thèse de Doctorat

présentée en vue de l'obtention du diplôme de
Docteur en Mathématiques

Spécialité : Mathématiques et Applications

Thème :

ETUDES COMPARATIVES DE QUELQUES ALGORITHMES D'IMAGERIE

Présentée par

Mme. **SOULEF BOUGUEROUA**

Directeur de thèse : Pr. **Nourreddine DAILI**

Thèse soutenue le 08 Juin 2024 devant le jury composé de :

Mr. Nacerdine HEMICI	Prof.	Université Sétif 1 Ferhat Abbas	Président
Mr. Nourreddine DAILI	Prof.	Université Sétif 1 Ferhat Abbas	Rapporteur
Mme. Sabrina GHERAIBIA	MCA	Université Skikda 20 Août 1955	Co-rapporteur
Mr. Feteh ELLAGGOUNE	Prof.	Université Guelma 8 Mai 1945	Examineur
Mr. Mohammed KARA	MCA	Université Sétif 1 Ferhat Abbas	Examineur
Mr. Messaoud MAOUNI	Prof.	Université Skikda 20 Août 1955	Examineur
Mr. Salim MESBAHI	Prof.	Université Sétif 1 Ferhat Abbas	Invité

ملخص

تعد المشاكل العكسية مجالاً واسعاً جداً، من بين المشاكل العكسية التي تهمني في عملي معالجة الصور. استعادة الصور هي مشكلة مثيرة للاهتمام وسببها الطرح ذات أهمية حاسمة في مفهوم معالجة الصور. الضوضاء تلحق الضرر بالصور لذلك تم تطوير العديد من الخوارزميات لمعالجة الصور: تنظيم تاكونوف؛ نموذج رودين-أوشر-فاطمي المستمر؛ نموذج إيف ماير ونموذج أوشر-سوليه-فيبيسي؛ نموذج غرايبية-دايلي سأتناول في أطروحتي نماذج الترميم المذكورة أعلاه؛ لقد أجريت دراسات مقارنة من خلال إجراء تطبيقات نظرية وعددية، قمنا بدراسة نظرية إسقاط برجمان والإسقاط المائل. نقدم طريقة تحسين جديدة لحل مشاكل استعادة الصور المشوشة بسبب الضوضاء الغوسية البيضاء المضافة، وتعتمد طريقة الحل هذه على خوارزمية عقوبة بديلة. تمت مقارنة النتائج العددية التي تم الحصول عليها بواسطة طريقة الجزاء الوكيل، وخوارزمية برجمان المنقسمة لمشاكل تقليل الضوضاء التلفزيونية متباينة الخواص من حيث جودة الصورة والتقارب ومعدل ضوضاء الإشارة في أطروحتي.

الكلمات الرئيسية: الخوارزميات؛ التطبيقات؛ دراسة مقارنة لترميم الصور؛ نماذج الترميم؛ معالجة الصور.

Résumé

Les problèmes inverses sont un domaine trop vaste. Parmi les problèmes inverses qui m'intéressent dans mon travail le traitement d'image. La restauration d'images est un problème mal posé intéressant et d'importance cruciale dans la notion du traitement d'image. Le bruit endommage les images, c'est pourquoi plusieurs algorithmes ont été développés pour le traitement des images: la régularisation de Tychonov; le modèle continu de Rudin-Osher-Fatemi; le modèle d'Yves Meyer; le modèle d'Osher-Solé-Vese; le modèle de Gheraibia Daili...

Dans cette thèse, on a traité les modèles de restauration cités ci-dessus. On a fait des études comparatives en faisant des implémentations théoriques et numériques. Nous avons étudié le théorème de projection de Bregman et la projection oblique. Nous présentons une nouvelle méthode d'optimisation pour résoudre les problèmes de restauration d'images perturbées par bruit blanc Gaussien additif. Cette méthode de résolution est basée sur un algorithme de proximale pénalité. Les résultats numériques obtenus par la méthode proximale pénalité, l'algorithme de split Bregman pour les problèmes de débruitage TV anisotrope et isotrope en termes de qualité d'image, de convergence et de taux de bruit de signal (SNR) est comparé dans ma thèse.

Mots clés: Algorithmes; Applications; Etude comparatives de restauration d'image; Modèles de restauration; Traitement d'images.

Abstract

Inverse problems are too broad a field. Among the inverse problems in which I am interested in image processing in my work. Image restoration is a problem that is ill-posed, interesting and of crucial importance to the notion of image processing. The noise damages images, which is why several algorithms have been developed for processing images: the regularization of Tychonov; the Rudin-Osher-Fatemi continuous model; the Model of Yves Meyer; the Osher-Solé-Vese model; the Gheraibia-Daili model; ...

In this thesis, we treated the restoration models cited above; we did comparative studies by making theoretical and numerical implementations. We studied the Bregman projection theorem and oblique projection. We present a new optimization method to solve the problems of restoring images disturbed by additive white Gaussian noise. This resolution method is based on an algorithm of prox-penalty. The numerical results obtained by the prox-penalty method, the algorithm split Bregman for anisotropic and isotropic TV denoising problems in terms of image quality, convergence and signal noise ratio (SNR), these are compared in my thesis.

Key words: Algorithms; Application; Image restoration comparative study; Restoration models; Image processing.

AMS 2020 : 68W40; 65K05; 65Z05; 65S99.

Remerciements

Pour chaque début il y a une fin, et ce qui est beau dans toute fin c'est la réussite et d'atteindre le but.

Avant tout, je remercie notre Dieu le tout-puissant de m'avoir donnée la force d'atteindre mon but et d'accomplir mon travail. Je tiens à adresser mes plus sincères remerciements à monsieur Nourreddine DAILI mon directeur de thèse, Professeur de l'enseignement supérieur.

Je le remerciais d'avoir donné accepter de diriger ce travail de recherche, encadrer, orienter, encourager, aider et conseiller. Je lui suis profondément reconnaissante de m'avoir accompagné pendant les années de préparation de cette thèse. Il a toujours apporté sa rigueur scientifique à certains points-clés de ce travail. Je lui suis particulièrement reconnaissante pour son soutien en me donnant sa grande liberté de travail.

J'adresse mes sincères remerciements à tous les membres de jury pour l'intérêt qu'ils ont bien voulu porter à ce travail de thèse:

- Monsieur Nacerdine HEMICI, Professeur à l'université Ferhat Abbas, Sétif 1 pour l'honneur qui me fait de présider ce jury, je le remercie très vivement;
- Madame Sabrina GHERAIBIA, Maître de conférences à l'université 20 Août 1955, Skikda;
- Monsieur Feteh ELLAGGOUNE, Professeur à l'université 8 Mai 1945 Guelma;
- Monsieur Salim MESBAHI, Professeur à l'université Ferhat Abbas, Sétif 1;
- Monsieur Mohammed KARA, Maître de conférences à l'université Ferhat Abbas, Sétif 1;
- Monsieur Messaoud MAOUNI, Professeur à l'université 20 Août 1955, Skikda;

d'avoir accepté et examiner ce travail, je leurs remercient très vivement pour leurs aides et leurs encouragements durant mes études. Ceci fut pour moi un honneur, je les remercie chaleureusement pour leurs critiques constructives, les remarques pertinentes et les corrections qui ont amélioré la présentation de ce manuscrit.

Je remercie beaucoup mes très chers parents pour tout ce qu'ils ont fait pour moi. Ils ont toujours été là pour moi.

Un grand merci à mon mari qui a toujours été là pour moi et merci pour ses encouragements et son aide. Je remercie mes sœurs et mes enfants " Iline et Iyed " pour leurs encouragements.

Merci à tous ceux qui ont contribué de près ou de loin pour la réalisation de ce travail.

Dédicace



Je dédie ce modeste travail :

*A mon cher père **Riad**♥*

Pour être le bon exemple de père avec ses sacrifices, et qui m'a encouragé et motivé depuis ma naissance.

*A ma chère mère **Boudaoud Nassima**♥*

Pour son amour, sa tendresse, son soutien et sa prière tout au long de mes études.

*Un grand merci à mon mari **Redjem Mouhamed**♥ qui a toujours été là pour moi et merci pour ses encouragements et son aide, merci à mes enfants " **Iline**♥ et **Iyed** ♥".*

*A mes chères sœurs ♥**Dounia** ♥ **Hana** ♥ **Malek** pour leurs encouragements permanents, et leur soutien moral.*

Ma petite famille Je ferai de mon mieux pour rester un sujet de fierté à vos Yeux avec l'espoir de ne jamais vous décevoir.

A toutes les personnes que j'aime Merci d'être toujours là pour moi.

B.SOULEF



Liste des tableaux

N°	Titre	Page
Tableau 2.1	Résultats obtenus par l'algorithme de régularisation de Tychonov, $\sigma = 0.01$.	33
Tableau 2.2	Résultats obtenus par l'algorithme de régularisation de Tychonov, $\sigma = 0.08$.	34
Tableau 2.3	Résultats obtenus par l'algorithme de régularisation de Tychonov, $\sigma = 0.2$.	35
Tableau 2.4	Mesures de performance par l'algorithme Tykhonov, $\sigma=0.01$.	36
Tableau 2.5	Mesures de performance par l'algorithme Tykhonov, $\sigma=0.08$.	37
Tableau 2.6	Mesures de performance par l'algorithme Tykhonov, $\sigma=0.2$.	37
Tableau 3.1	Résultats obtenus par l'algorithme de régularisation de ROF, $\sigma = 0.01$.	47
Tableau 3.2	Résultats obtenus par l'algorithme de régularisation de ROF, $\sigma = 0.08$.	48
Tableau 3.3	Résultats obtenus par l'algorithme de régularisation de ROF, $\sigma = 0.2$.	49
Tableau 3.4	Mesures de performance par l'algorithme ROF, $\sigma=0.01$.	50
Tableau 3.5	Mesures de performance par l'algorithme ROF, $\sigma=0.08$.	50
Tableau 3.6	Mesures de performance par l'algorithme ROF, $\sigma=0.2$.	50
Tableau 6.1	Résultats obtenus par l'algorithme de débruitage TV anisotrope, $\sigma = 0.08$.	84
Tableau 6.2	Résultats obtenus par l'algorithme de débruitage TV isotrope, $\sigma = 0.08$.	84
Tableau 6.3	Mesures de performance par l'algorithme de débruitage TV anisotrope, $\sigma = 0.08$.	84
Tableau 6.4	Mesures de performance par l'algorithme de débruitage TV isotrope, $\sigma = 0.08$.	84
Tableau 6.5	Résultats obtenus par l'algorithme de débruitage TV anisotrope pour l'image flower, $\sigma = 0.08$.	91

Tableau 6.6	Résultats obtenus par l'algorithme de débruitage TV isotrope pour l'image flower, $\sigma = 0.08$.	91
Tableau 6.7	Les différentes valeurs de SNR pour image débruitée par : Prox ; TV anisotrop ; TV isotrop.	91

Liste des figures

N°	Titre	Page
Figure 1	Historique du traitement d'image .	6
Figure 1.1	Mesures de qualité d'image.	21
Figure 2.1	Les images de textes " cameraman, girl, university".	34
Figure 2.2	Image bruitée de cameraman et image débruitée par Tykhonov, sigma 0.08.	35
Figure 2.3	Image bruitée de girl et image débruitée par Tykhonov, sigma 0.08.	36
Figure 2.4	Image bruitée de University et image débruitée par Tykhonov, sigma 0.08.	37
Figure 3.1	Décomposition dans BV.	40
Figure 3.2	Les images de textes " animaux, fleur, camera".	46
Figure 3.3	Image bruitée de animaux et image débruitée par ROF, sigma 0.08.	47
Figure 3.4	Image bruitée de fleur et image débruitée par ROF, sigma 0.08.	48
Figure 3.5	Image bruitée de camera et image débruitée par ROF, sigma 0.08.	49
Figure 5 .1	Distance de Bregman.	64
Figure 5.2	Projection oblique.	67
Figure 6 .1	Résultats obtenus de Split Bregman en utilisant l'image de cameraman.	85
Figure 6 .2	Résultats obtenus de l'erreur de Split Bregman en utilisant l'image de cameraman.	86
Figure 6 .3	Résultats obtenus de Split Bregman en utilisant l'image de university.	87
Figure 6 .4	Résultats obtenus de l'erreur de Split Bregman en utilisant l'image de university.	88
Figure 6 .5	Résultats obtenus de Split Bregman en utilisant l'image de Iline.	89
Figure 6 .6	Résultats obtenus de l'erreur de Split Bregman en utilisant l'image de Iline.	90
Figure 6 .7	Image originale et image débruitée pour sigma = 0.08.	92
Figure 6 .8	Le SNR pour image débruitée par Prox ; TV anisotrop ; TV isotrop pour sigma = 0.08.	93
Figure 6 .9	Image originale et image débruitée pour sigma = 0.501.	94
Figure 6 .10	Le SNR pour image débruitée par Prox ; TV anisotrop ; TV isotrop pour sigma = 0.501.	95

Travaux réalisés par la candidate

- S. Bougueroua and N. Daili, Comparative Analysis of the Prox Penalty and Bregman Algorithms for Image Denoising, Journal of Applied Mathematics, Volume 2023, Article ID 6689311, 15 pages, <https://doi.org/10.1155/2023/6689311>.
- S. Bougueroua and N. Daili, Numerical Comparisons of Different Imaging Algorithms, Scientific Research Publishing, Journal of Applied Mathematics and Physics, 2023, 11, 2671-2690; Article ID:1723299, <https://www.scirp.org/journal/jamp>.
- S. Bougueroua and N. Daili, Comparaison of Some Algorithms In Image Processing, held on "March 28-30-2022" at India, "2nd International Conference on Computational Sciences-Modelling Computing and Soft Computing".
- S. Bougueroua and N. Daili, Comparative Theoretical and Practical Study of Some Imaging Algorithm, held on "May 10-13-2022" at Konya/Turkey, "1st International Conference on Engineering and Applied Natural Sciences".
- S. Bougueroua and N. Daili, Image restoration by the Prox Penalty Method with Bregman Projection, held on "29 November 2022" at Mila - Algeria, " National Conference on Mathematics and Applications NCMA2022".
- S. Bougueroua and N. Daili, Modern methods in image processing and their relationship with ROF model, le "04-07-2023" à l'université Echahid Cheikh Larbi Tbessi-Tbessa, "Deuxième Conférence Nationale Des Mathématiques Pures et Appliquées en ligne CMPA2023".
- S. Bougueroua and N. Daili, Study of Some Algorithm in Image processing, held on "15-16 May 2023" at Ferhat Abbas University Setif 1- Algeria, " National Conference on Mathematics and Applications NCMA2023".

Table des matières

0.1	Introduction Générale	4
0.1.1	L'Histoire du Traitement d'Image	4
0.1.2	Domaines d'Applications du Traitement d'Image	7
0.1.3	Algorithmes d'Imagerie	8
0.1.4	L'Objectif et le Contenu du Mémoire	8
1	Généralités	10
1.1	Eléments de la théorie d'Analyse Fonctionnelle	10
1.1.1	Opérateur Linéaire Borné	11
1.1.2	Le Spectre et Résolvante d'un Opérateur	12
1.2	Généralités sur les Espaces de Sobolev	12
1.3	Optimisation dans les Espaces de Banach	13
1.3.1	Semi-continuité et Convexité de Fonctionnelles sur V	14
1.3.2	Gâteaux-Différentiabilité des Fonctionnelles Convexes	14
1.3.3	Minimisation dans un Banach Réflexif	15
1.3.4	Les Projections	15
1.4	Problème Inverse et Problème Mal Posé	18
1.5	Traitement Numérique des Images	18
1.5.1	Les Images Numérique	18
1.5.2	L'Imagerie Médicale	20
1.6	Des Mesures	21
1.6.1	Erreur quadratique moyenne	21
1.6.2	Rapport signal/bruit de crête	21
1.6.3	Erreur absolue normalisée	21
1.6.4	Différence moyenne	21
1.6.5	Différence maximale	22
1.6.6	Corrélation croisée normalisée	22
1.6.7	Contenu structurel	22
2	La Méthode de Régularisation de Tychonov	24
2.1	Introduction Historique	24
2.2	Régularisation Linéaire (Tykhonov)	25
2.2.1	La Méthode de Tykhonov en EDP	26
2.3	La Méthode de Tykhonov Généralisé	27
2.3.1	Deux Méthodes de Tikhonov Généralisées	28
2.4	La Méthode de Tykhonov en Imagerie	31

2.4.1	Résultats Numériques pour le Modèle de Tykhonov	33
2.5	Conclusions	33
3	Modèle de Rudin-Osher-Fatemi	38
3.1	Introduction Historique	38
3.2	Problème de ROF	39
3.2.1	Discrétisation	43
3.2.2	Image Débruitée	45
3.3	Résultats Numériques	45
3.4	Conclusions	46
4	Modèle de Yves Meyer et Vese Osher	51
4.1	Introduction	51
4.2	Problème de Meyer	52
4.2.1	La relation entre ROF et Meyer	53
4.2.2	Discrétisation du Modèle de Meyer	54
4.3	Problème de Vese Osher	55
4.3.1	Discrétisation Numérique du Modèle de Meyer	55
4.3.2	Résolution du Problème de Vese Osher	57
5	Théorèmes de Projection de Bregman et Projection Oblique	59
5.1	Introduction Historique	59
5.2	Théorème de Projection de Bregman	60
5.2.1	Distance de Bregman	61
5.3	Théorème de Projection Oblique	64
5.3.1	Introduction au projection oblique	64
5.3.2	Projection Oblique	65
5.3.3	Projection oblique itérative sur des ensembles convex et problèmes de faisabilité divisée (SFP)	67
5.3.4	Problèmes (SFP) et l'algorithme CQ	67
5.3.5	Convergence de l'algorithme CQ	69
5.3.6	Les méthodes Landweber	70
5.3.7	Le CFP et l'algorithme MSFP	70
5.4	Algorithmes Itérative de Bregman	71
5.4.1	Théorème de Convergence	73
6	Analyses Comparatives des Algorithmes de Proximale Pénalité et de Bregman pour le Débruitage D'image	74
6.1	Méthodes Proximale-Pénalité	74
6.2	Algorithmes de Bregman et Imagerie	77
6.2.1	Algorithme de Split Bregman	77
6.2.2	Problème de Débruitage TV Anisotrope	78
6.2.3	Problème de Débruitage TV Isotrope	79
6.2.4	Combinaison des Problèmes Débruitage TV Anisotrope et Isotrope	79

6.2.5 Théorème de Convergence	80
6.3 Résultats Numériques	83
6.4 Conclusion	91
Bibliographie	93

0.1 Introduction Générale

En 1923, le mathématicien français J. HADAMARD a écrit son célèbre livre sur les équations aux dérivées partielles et leurs implications physiques. Ce livre a servi de base pour la croissance de l'idée d'un problème bien posé en physique mathématique. Il s'agit d'un problème dont la solution existe, unique et dépend continûment des données (stabilité). J. HADAMARD a affirmé (comme l'a fait I.G. PETROVSKY) dans le même livre que seul un problème bien posé pouvait modéliser correctement un phénomène physique. Le fait qu'il ait des réponses multiples (la non-unicité de la solution) est un grand défi parce qu'il faut découvrir une technique pour choisir la meilleure, celle qui est physiquement la plus parfaite. L'instabilité est la plus grande difficulté, surtout lorsqu'on approche ou utilise la résolution numérique. Les données fournies seront bruyantes et donc proches mais différentes des données réelles, ce qui rend impossible une approche satisfaisante de la solution du problème (quelle que soit la méthode numérique).

Les problèmes mal posés ont longtemps été ignorés par la physique mathématique parce qu'elle estime que soit elles manquent de toute signification physique ou que leur modélisation est insuffisante. Par conséquent, un problème mal posé est un problème pour lequel au moins une des trois conditions n'est pas confirmée ; néanmoins, l'incapacité de satisfaire à la troisième condition pose la plus grande difficulté. Il s'agit donc de problèmes pour lesquels une petite perturbation des données peut provoquer un changement important de la solution. La réalité actuelle est toute autre (significativement différente) : le caractère fondamentalement mal posé de certains problèmes pratiques est reconnu et se manifeste dans une très large classe de problèmes, dits "Problèmes inverses", et motive de nombreuses recherches en mathématiques.

Les problèmes inverses sont un domaine trop vaste pour que nous puissions en donner un exposé exhaustif. D'après J.B Keller un problème inverse consiste à déterminer des causes connaissant des effets. Ainsi, ce problème est l'inverse de l'autre appelé problème direct, consistant à déduire les effets, les causes étant connues. Les problèmes inverses mal-posés sont multiples, et leurs applications se retrouvent dans de nombreux domaines par exemples pour les EDP et imagerie ([52],[40] et [37]). Parmi les problèmes inverse qui m'intéressent dans mon travail le traitement d'image. Toutes les manipulations effectuées sur une image afin d'accroître la lisibilité et l'interprétation sont appelées traitement d'image. On fait également référence à un groupe d'activités de prétraitement qui utilisent toutes des méthodes pour améliorer la qualité d'une image. En conséquence, l'image originale est la donnée de départ, et le résultat est également une image.

0.1.1 L'Histoire du Traitement d'Image

L'histoire du traitement d'image remonte au début de la photographie et a évolué en parallèle avec les progrès technologiques dans ce domaine. Voici une brève chronologie des développements clés dans l'histoire du traitement d'image :

- 1826 : La première photographie permanente est réalisée par Joseph Nicéphore Niépce en utilisant une plaque recouverte de bitume de Judée exposée à la lumière.

- 1839 : Louis Daguerre présente le procédé du daguerréotype, une méthode de photographie utilisant des plaques de cuivre recouvertes de vapeurs d'iode pour capturer des images.
- 1861 : James Clerk Maxwell réalise la première photographie couleur en combinant des images prises à travers des filtres rouge, vert et bleu.
- 1888 : George Eastman lance le premier appareil photo grand public, le Kodak, qui utilise des rouleaux de pellicule souple.
- 1960 : La NASA développe le premier scanner d'images, utilisé pour convertir les photographies en signaux numériques pour les transmettre depuis l'espace.
- 1973 : Le chercheur Thomas Knoll développe le programme "Display" qui, des années plus tard, deviendra Adobe Photoshop, le logiciel de retouche d'image le plus populaire au monde.
- 1987 : La première conférence internationale sur le traitement d'image et sa qualité (ICIP) est organisée, marquant la reconnaissance du traitement d'image comme un domaine de recherche et de développement à part entière.
- **Années 1990** : L'avènement de la photographie numérique et des capteurs CCD/CMOS ouvre la voie à des avancées majeures dans le traitement d'image. Les images numériques peuvent être manipulées, corrigées et améliorées de manière plus flexible que les photographies analogiques.
- **Années 2000** : Les algorithmes de vision par ordinateur et d'apprentissage automatique (machine learning) sont appliqués au traitement d'image, permettant des avancées dans la reconnaissance d'objets, la détection de visages, la segmentation d'image, etc.
- **Aujourd'hui**, le traitement d'image est un domaine multidisciplinaire qui englobe des techniques telles que la restauration d'image, la fusion d'image, la super-résolution, la réalité augmentée, l'imagerie médicale avancée, la reconnaissance d'images, et bien plus encore. Le traitement d'image joue un rôle crucial dans de nombreux domaines, allant de la médecine à l'industrie, en passant par les médias, la surveillance, l'art et le divertissement.

Le traitement d'image fait référence à l'ensemble des techniques et des méthodes utilisées pour modifier, améliorer ou analyser des images numériques. Cela peut inclure des opérations de base telles que le redimensionnement, le recadrage, la rotation, la correction de la luminosité et du contraste, ainsi que des opérations plus avancées telles que la détection d'objets, la segmentation d'images, la restauration d'images, et bien plus encore. Il existe de nombreux outils et bibliothèques logicielles disponibles pour effectuer des traitements d'images, notamment des logiciels tels que Adobe Photoshop, GIMP, et des bibliothèques de programmation comme Open CV, Scikit-image, MATLAB Image Processing Toolbox, et d'autres encore.

Les étapes générales impliquées dans le traitement d'image comprennent généralement :

1. **Acquisition de l'image** : Capture ou importation de l'image dans un format numérique.
2. **Prétraitement** : Cela peut inclure des opérations telles que la normalisation, la réduction du bruit, le filtrage, la correction des défauts d'acquisition, etc.
3. **Amélioration** : Il s'agit d'opérations visant à améliorer les caractéristiques de l'image, telles que l'ajustement des niveaux de contraste, la correction de la balance des couleurs, la netteté, etc.
4. **Transformation** : Application de techniques spécifiques pour changer la représentation de l'image, comme la transformation de Fourier, la transformation en ondelettes, etc.
5. **Analyse et traitement** : Utilisation de méthodes d'analyse d'image pour extraire



Figure 1 – Historique du traitement d'image.

des informations spécifiques, telles que la détection de contours, la reconnaissance d'objets, la classification, etc.

6. **Compression** : Réduction de la taille de l'image pour économiser de l'espace de stockage et faciliter le transfert.
7. **Affichage ou sortie** : Visualisation de l'image traitée ou création d'une nouvelle image résultante.

Le traitement d'image est une discipline issue du mariage de théorie de l'information et de l'informatique qui se concentre sur la manipulation, l'analyse et l'amélioration des images numériques à l'aide d'algorithmes et de techniques informatiques. Il vise à extraire des informations utiles, à améliorer la qualité visuelle, à détecter des motifs ou des objets spécifiques, et à effectuer diverses tâches liées à l'imagerie. Le traitement d'image trouve des applications dans de nombreux domaines, tels que la photographie numérique, la télévision, la médecine, la vision par ordinateur, la reconnaissance de formes, la robotique, la surveillance, la sécurité et la surveillance, la géographie, etc. Il joue un rôle essentiel dans la compréhension et l'exploitation des informations visuelles contenues dans les images. Le traitement d'image peut inclure des opérations de base telles que le redimensionnement, la rotation, le filtrage, ainsi que des opérations plus avancées telles que la segmentation, la détection d'objets, la reconnaissance de formes, etc.

En résumé, le traitement d'image est un domaine dynamique et en évolution constante, qui utilise des techniques et des algorithmes pour manipuler, analyser et améliorer les images numériques. Il joue un rôle important dans de nombreux domaines et continue d'avoir un

impact significatif sur notre vie quotidienne.

0.1.2 Domaines d'Applications du Traitement d'Image

Les exemples et domaines d'applications du traitement d'image sont nombreux. Les deux principaux domaines, qui ont permis au traitement d'image de se développer sont :

- 1- Le domaine militaire : missiles en tous genres (autodirecteur (courte portée), de croisière (longue portée), etc.) ; le renseignement (télédétection à partir d'images satellites dont la précision peut aller aujourd'hui jusqu'à quelques centimètres, ou aériennes, la photo-interprétation) ; les simulateurs réels (de vol avion, de char, etc).
- 2- Le domaine médical : l'imagerie médicale (Ultrasons, IRM, tomographie ; angiographie ; radiographie ; échographie ; scanner ; etc).

Une multitude d'applications apparaissent dans des domaines divers allant :

- * du domaine industriel : le contrôle de la qualité des produits en bout de chaîne (état de surface, dimensionnement, forme, couleur, présence des comprimés par exemple dans l'industrie pharmaceutique) ; le contrôle non destructif ; les manipulations automatiques par des robots ; etc ;
- * allant jusqu'au domaine du multimédia : la compression / décompression pour la transmission d'images ou le stockage ; l'amélioration de la qualité ; la TV haute définition (Standard MPEG4) ; la TV en relief ; le coloriage des dessins animés ; le remplacement des panneaux publicitaires dans des retransmissions sportives ; la recherche d'images par le contenu dans des banques d'images ; la vidéo-conférence.

En passant par de nombreux domaines dont :

- * l'imagerie civile satellitaire et aérienne : la météorologie ; la cartographie ; l'aménagement du territoire ;
- * la reconnaissance de caractères, de documents : le photocopieur intelligent ; le tri automatique du courrier postal ; la reconnaissance automatique des plaques d'immatriculation de véhicules ;
- * la robotique mobile et autonome militaire : le robot aidant le fantassin sur le champ de bataille ; les robots sans pilote : martien, avion (drone), sous-marin ;
- * la conduite assistée par ordinateur, à la conduite automatique : suivi de routes ; reconnaissance des panneaux ; reconnaissance des voies de circulation ; la détection d'obstacles ; détection de piétons ;
- * la régularisation de flux par le biais de comptage et suivi d'objets : automobiles : temps de parcours, indication de vitesse ou de proximité par rapport au véhicule précédent ; personnes : réduction des files d'attente ;
- * la surveillance de sites ou locaux stratégiques, protection de données : éléments biométriques : empreintes, iris, visage pour la reconnaissance de personnes autorisées, la validation d'identité et la détection d'intrus ; la vidéoprotection : détection automatique de situation à problèmes.

0.1.3 Algorithmes d'Imagerie

Le traitement d'image comprend une variété de techniques et d'algorithmes. Il existe de nombreux algorithmes utilisés en imagerie pour traiter et analyser les images. Voici quelques exemples d'algorithmes couramment utilisés :

1. **Filtrage** : Les algorithmes de filtrage sont utilisés pour supprimer le bruit, améliorer le contraste, lisser les contours et mettre en évidence certaines caractéristiques de l'image. Les filtres linéaires (passe-bas, passe-haut, passe-bande) et les filtres non linéaires (médian, bilatéral) sont largement utilisés ;
2. **Segmentation** : Les algorithmes de segmentation sont utilisés pour diviser une image en régions ou objets distincts. Parmi les méthodes populaires, on trouve le seuillage, la croissance de région, les contours actifs, les méthodes basées sur les graphes (comme la coupe de graphes) et les algorithmes de clustering (comme le k-means) ;
3. **Détection de contours** : Les algorithmes de détection de contours identifient les changements significatifs d'intensité dans une image, permettant de détecter les contours et les bords ;
4. **Transformation de Fourier** : La transformation de Fourier est utilisée pour analyser les fréquences présentes dans une image. Elle permet de passer de l'espace de l'image à l'espace des fréquences. Des filtres fréquentiels peuvent être utilisés pour supprimer ou accentuer certaines fréquences dans l'image ;
5. **Descripteurs et caractéristiques** : Les algorithmes de calcul de descripteurs et de caractéristiques extraient des informations spécifiques de l'image pour la reconnaissance, la classification ou la correspondance. Des exemples incluent les histogrammes de couleur, les descripteurs de texture, les descripteurs de forme, les points d'intérêt (SIFT, SURF) et les descripteurs basés sur l'apprentissage en profondeur (comme les réseaux de neurones convolutifs) ;
6. **Restauration d'image** : Les algorithmes de restauration d'image sont utilisés pour réduire le bruit, éliminer les artefacts, améliorer la résolution ou la qualité globale de l'image. Parmi les méthodes courantes, on trouve les techniques de déconvolution, la restauration basée sur les modèles, l'utilisation de filtres adaptatifs, etc.

Il est important de noter que la sélection de l'algorithme dépendra du problème spécifique, des caractéristiques de l'image, des contraintes computationnelles et des objectifs souhaités. Différentes techniques peuvent être combinées pour obtenir de meilleurs résultats ou pour répondre à des besoins spécifiques dans le domaine de l'imagerie.

0.1.4 L'Objectif et le Contenu du Mémoire

Ce travail sert d'introduction à la restauration d'image, qui est un problème mal-posé intéressant et a une importance cruciale dans la notion de traitement d'image. Le processus de récupération d'une image presque identique à l'image originale à partir d'une observation,

généralement une image bruyante ou floue de l'image réelle, est connu sous le nom de restauration d'image. Plusieurs modèles de restauration ont été appliqués à divers problèmes mal posé dans la littérature mathématique. Parmi elles, on cite :

- La méthode de régularisation de Tychonov ;
- Le modèle Continu de Rudin-Osher-Fatemi ;
- Le modèle de Yves Meyer ;
- Le modèle de Osher-Solé-Vese ;
- Modèle de Gheraibia-Daili ; ...

Dans ma thèse, j'ai étudié problèmes mal posés engendrés par traitement et restauration d'image. L'examen des modèles précédents est le but de ce travail. J'ai mené des études comparatives en utilisant des implémentations numériques, en créant de nouvelles techniques de modélisation et d'optimisation pour l'imagerie, en envisageant de nouvelles approches, en identifiant certains critères de convergence, et l'introduction d'algorithmes pour résoudre ces problèmes.

Ce mémoire est composée d'une introduction et de six chapitres, d'une conclusion et de quelques annexes.

Dans l'introduction, on décrit les motivations et les objectifs de l'étude abordée. On développe la problématique et on donne une description détaillée du mémoire.

Le premier chapitre est consacré à la présentation des concepts de base ainsi que le rappel des outils d'analyse fonctionnelle et l'optimisation dans les espaces de Banach. On introduit les notions de problème mal posé et de problème inverse, le traitement numérique des images avec des illustrations des mesures pour la programmation dans MATLAB.

Dans le deuxième chapitre, on donne une stratégie de régularisation basée sur la méthode de régularisation de Tykhonov en EDP, Tykhonov généralisée et la méthode de Tykhonov en imagerie. Un script MATLAB pour les résultats numériques du modèle Tykhonov.

Dans le troisième chapitre, nous étudions le modèle Continu de Rudin-Osher-Fatemi ; on donne les résultats numériques du modèle ROF.

Dans le quatrième chapitre, nous étudions le modèle de Yves Meyer et problème de Vese Osher avec la discrétisation numérique du modèle de Meyer.

Au chapitre cinq, nous étudions les théorèmes de Projection de Bregman et Projection Oblique.

Au chapitre six, on présente des analyses comparatives des algorithmes de proximale pénalité et de Bregman pour le débruitage d'image.

Chapitre 1

Généralités

1.1 Éléments de la théorie d'Analyse Fonctionnelle

Définition 1.1.1 [57] [30] [34] Soit E un espace vectoriel sur \mathbb{k} . Une norme sur E est une application de E dans \mathbb{R}^+ telle que

1. $\forall x \in E \quad \|x\|_E = 0 \Leftrightarrow x = 0$;
2. $\forall x \in E, \forall \alpha \in \mathbb{R} \quad \|\alpha x\|_E = |\alpha| \cdot \|x\|_E$;
3. $\forall (x, y) \in E^2 \quad \|x + y\| \leq \|x\|_E + \|y\|_E$;

Un tel espace E muni d'une norme $\|\cdot\|$ est dit espace vectoriel normé.

Définition 1.1.2 [57] [30] [34] Soit E un espace vectoriel sur \mathbb{k} . Un produit scalaire sur E est une application de $E \times E$ dans \mathbb{k} telle que pour tout $(x, y, z) \in E^3$ et $(\alpha, \beta) \in \mathbb{k}^2$, on a :

1. $\langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$;
2. $\langle x, y \rangle = \overline{\langle y, x \rangle}$;
3. $\langle x, x \rangle \geq 0$;
4. $\langle x, x \rangle = 0 \Leftrightarrow x = 0$.

Un espace vectoriel muni d'un produit scalaire est appelé un espace préhilbertien.

Théorème 1.1.1 [57] [30] [34] Dans un espace préhilbertien E , l'application $\|\cdot\| : E \rightarrow \mathbb{R}$ est une norme, donnée par

$$\|x\|_E = \sqrt{\langle x, x \rangle}, \text{ pour tout } x \in E.$$

Définition 1.1.3 [57] [30] [34] Un espace de Hilbert est un espace complet par rapport à la norme induite par un produit scalaire. En d'autres mots, un espace de Hilbert est un espace de Banach dont la norme est induite par un produit scalaire.

Définition 1.1.4 [57] [30] [34] On appelle base Hilbertienne d'un espace de Hilbert, toute suite $(e_n)_{n \in \mathbb{N}}$ tel que :

1. $\|e_n\| = 1 \quad \forall n \in \mathbb{N}$;
2. $\langle e_n, e_m \rangle = 0 \quad \forall n, m \in \mathbb{N}, \quad n \neq m$.

L'espace vectoriel engendré par cette base est dense dans H . Tout élément $u \in H$, s'écrit

$$u = \sum_{n=1}^{\infty} \langle u, e_n \rangle e_n.$$

1.1.1 Opérateur Linéaire Borné

Soient H_1, H_2 deux espaces vectoriels normés et $A : H_1 \rightarrow H_2$ est un opérateur linéaire, le théorème suivant caractérise la continuité d'un opérateur linéaire.

Théorème 1.1.2 [57] [30] [34] *Les propriétés suivantes sont équivalentes :*

1. A est continu.
2. A est continu en 0.
3. $\exists c > 0$ tel que $\|Ax\| \leq c \|x\| \quad \forall x \in E$.

Un opérateur A est linéaire continu d'un espace de Hilbert H_1 dans un espace de Hilbert H_2 est une application linéaire continue de H_1 dans H_2 , s'il vérifie :

1. $\forall x \in H_1, Ax \in H_2$
2. $\forall (x, y) \in H_1 \times H_1, \forall (\alpha, \beta) \in \mathbb{R}^2, A(\alpha x + \beta y) = \alpha Ax + \beta Ay$
3. $\exists c > 0, \forall x \in H_1 \quad \|Ax\|_{H_2} \leq c \|x\|_{H_1}$

Le plus petite nombre c qui vérifie le 3^{ème} point ci-dessus s'appelle la norme de A .

$$\|A\| = \sup_{x \in H_1} \frac{\|Ax\|_{H_2}}{\|x\|_{H_1}};$$

$$\|A\| = \sup_{\|x\|=1} \|Ax\|_{H_2};$$

$$\|A\| = \sup_{\|x\| \leq 1} \|Ax\|_{H_2};$$

$$\|A\| = \inf \{c > 0, \quad \forall x \in H_1 \quad \|Ax\|_{H_2} \leq c \|x\|_{H_1}\};$$

Théorème 1.1.3 [57] [30] [34] (*Théorème des isomorphismes de Banach*)
Toute application bijectif linéaire continue $A \in \mathcal{L}(H_1, H_2)$ est inversible.

Théorème 1.1.4 [57] [30] [34] (*L'adjoint d'un opérateur*)

Soit A un opérateur linéaire continu de H_1 dans H_2 , il existe un unique opérateur de H_2 dans H_1 noté A^* tel que

$$\forall x \in H_1, \forall y \in H_2 \quad \langle Ax, y \rangle = \langle x, A^*y \rangle,$$

cet opérateur est appelé l'adjoint de A . Il vérifie $(A^*)^* = A$ et $\|A^*\| = \|A\|$;

1. $\ker(A^*) = (\text{Im}A)^\perp$.
2. $(\ker A)^\perp = \overline{\text{Im}A^*}$.

1.1.2 Le Spectre et Résolvante d'un Opérateur

Définition 1.1.5 [57] [30] [34] Soit $A \in L(E)$.

1. On appelle spectre de A , l'ensemble

$$\sigma(A) = \{\lambda \in \mathbb{k}, (\lambda I - A) \text{ non inversible}\},$$

tout scalaire $\lambda \in \sigma(A)$ est dit valeur spectrale.

Le rayon spectrale de A noté $r(A)$, est défini par

$$r(A) = \sup \{\lambda \mid \lambda \in \sigma(A)\}.$$

Examinons à présent de plus près la structure du spectre.

2. Le premier sous-ensemble important du spectre est le spectre ponctuel

$$\sigma_p(A) = \{\lambda \in \mathbb{k}, (\lambda I - A) \text{ n'est pas injectif}\}.$$

3. On appelle spectre continue de A l'ensemble

$$\sigma_c(A) = \{\lambda \in \mathbb{k}, (\lambda I - A) \text{ est injectif et } \overline{\text{Im}(\lambda I - A)} \text{ est dense dans } E\}.$$

4. On appelle spectre résiduel de A l'ensemble

$$\sigma_r(A) = \{\lambda \in \mathbb{k}, (\lambda I - A) \text{ est injectif, } \overline{\text{Im}(\lambda I - A)} \text{ n'est pas dense dans } E\},$$

et on a $\sigma(A) = \sigma_p(A) \cup \sigma_c(A) \cup \sigma_r(A)$.

5. On appelle l'ensemble résolvant de A , l'ensemble

$$\rho(A) = \{\lambda \in \mathbb{k}, (\lambda I - A) \text{ inversible}\}.$$

$$\sigma(A) = \mathbb{k} \setminus \rho(A).$$

Si $\lambda \in \rho(A)$, alors on note $R_\lambda(A) = (\lambda I - A)^{-1} \in L(E)$ la résolvante de A .

1.2 Généralités sur les Espaces de Sobolev

Définition 1.2.1 ([6]) Soit Ω un ouvert borné de \mathbb{R}^n , (en pratique $n \leq 3$) de frontière régulière Γ . On appelle $D(\Omega)$ l'espace des fonctions C^∞ à support compact dans Ω . Son dual $D'(\Omega)$ est l'espace des distributions sur Ω . Pour toute distribution $u \in D'(\Omega)$, la dérivée $\frac{\partial u}{\partial x_i}$ est définie de la manière suivante :

$$\forall \varphi \in D(\Omega) \quad \left\langle \frac{\partial u}{\partial x_i}, \varphi \right\rangle_{D'(\Omega), D(\Omega)} \equiv - \left\langle u, \frac{\partial \varphi}{\partial x_i} \right\rangle_{D'(\Omega), D(\Omega)}.$$

On notera la dérivée de u au sens des distributions $D_i u = \frac{\partial u}{\partial x_i} = \partial_i u$.

Si $a \in \mathbb{N}^n$, on note $D^a u = \partial_1^{a_1} u \dots \partial_n^{a_n} u$ avec $|a| = a_1 + \dots + a_n$, on obtient :

$$\forall \varphi \in D(\Omega) \quad \langle D^a u, \varphi \rangle_{D'(\Omega), D(\Omega)} = (-1)^{|a|} \langle u, D^a \varphi \rangle_{D'(\Omega), D(\Omega)}.$$

Définition 1.2.2 ([6]) On définit les espaces de Sobolev $H^m(\Omega)$ de la manière suivante :

$$\begin{aligned} H^0(\Omega) &= L^2(\Omega), \\ H^1(\Omega) &= \left\{ u \in L^2(\Omega) \mid \frac{\partial u}{\partial x_i} \in L^2(\Omega), i = 1 \dots n \right\}, \\ H^m(\Omega) &= \left\{ u \in D'(\Omega) \mid D^a u \in L^2(\Omega), |a| \leq m \right\}. \end{aligned}$$

Proposition 1.2.1 ([6])

1. $H^m(\Omega)$ muni du produit scalaire :

$$\langle u, v \rangle_m = \sum_{|a| \leq m} \int_{\Omega} D^a u(x) D^a v(x) dx,$$

est un espace de Hilbert.

2. $H^m(\Omega) \subset H^{m'}(\Omega)$ et l'injection est continue, pour $m \geq m'$.

Définition 1.2.3 ([6])

1. Soit

$$H_0^1(\Omega) = \{ u \in H^1(\Omega) \mid u|_{\Gamma} = 0 \},$$

C'est aussi l'adhérence de $D(\Omega)$ dans $H^1(\Omega)$.

2. Soit

$$H_0^m(\Omega) = \left\{ u \in H^1(\Omega) \mid \frac{\partial^j u}{\partial n_{|\Gamma}^j} = 0, j = 1, \dots, m-1 \right\},$$

où $\frac{\partial u}{\partial n}$ est la dérivée de u suivant la normale extérieure à Γ la frontière de Ω :

$$\frac{\partial u}{\partial n} = \sum_{i=1}^n \frac{\partial u}{\partial x_i} \cos(\vec{n}, \vec{e}_i)$$

où \vec{n} est la normale extérieure à Γ et Ω est supposé "régulier" (de frontière C^∞ par exemple).

3. **La Dualité** : Pour tout $m \in \mathbb{N}$, on note $H^{-m}(\Omega)$ le dual de $H_0^1(\Omega)$.

1.3 Optimisation dans les Espaces de Banach

On considère dans toute cette section un espace de Banach réflexif V de dual (topologique) V' . On note $\|V\|$ la norme de V et $\langle \cdot, \cdot \rangle$ le crochet de dualité entre V et V' .

1.3.1 Semi-continuité et Convexité de Fonctionnelles sur V

Définition 1.3.1 ([6]) Une fonction J de V dans $\mathbb{R} \cup \{+\infty\}$ est semi-continue inférieurement (sci) sur V si elle satisfait aux conditions équivalentes :

- 1 $\forall a \in \mathbb{R}, \{u \in V \mid J(u) \leq a\}$ est fermé.
- 2 $\forall \bar{u} \in \mathbb{R}, \liminf_{u \rightarrow \bar{u}} J(u) \geq J(\bar{u})$.

Théorème 1.3.1 ([6]) Toute fonction convexe sci pour la topologie forte (celle de la norme) de V est encore (sci) pour la topologie faible de V .

En pratique ce résultat s'utilise sous la forme du corollaire suivant :

Corollaire 1.3.1 ([6]) Soit J une fonctionnelle convexe de V dans $\mathbb{R} \cup \{+\infty\}$ sci pour la topologie forte. Si v_n est une suite de V faiblement convergente vers v alors

$$J(v) \leq \liminf_{n \rightarrow +\infty} J(v_n).$$

1.3.2 Gâteaux-Différentiabilité des Fonctionnelles Convexes

Définition 1.3.2 ([6]) Soit J une fonctionnelle de V dans $\mathbb{R} \cup \{+\infty\}$. On dit que J est Gâteaux-différentiable en $u \in \text{dom}(J)$ si la dérivée directionnelle

$$J'(u; v) = \lim_{t \rightarrow 0^+} \frac{J(u + tv) - J(u)}{t},$$

existe dans toute direction v de V et si l'application $v \mapsto J'(u; v)$ est linéaire continue.

De manière générale on notera $\nabla J(u)$ le Gâteaux-différentiel de J en u . C'est un élément du dual V' .

Si V est un espace de Hilbert, avec le théorème de représentation de Riesz on identifie V et son dual ; on note alors

$$J'(u; v) = (\nabla J(u), v),$$

où (\cdot, \cdot) désigne le produit scalaire de V . L'élément $\nabla J(u)$ de V est le gradient de J en u .

Théorème 1.3.2 ([6]) Soit $J : C \subset V \rightarrow \mathbb{R}$, Gâteaux-différentiable sur C , avec C convexe. J est convexe si et seulement si

$$\forall (u, v) \in C \times C \quad J(v) \geq J(u) + \langle \nabla J(u), v - u \rangle.$$

Théorème 1.3.3 ([6]) Soit $J : C \subset V \rightarrow \mathbb{R}$, Gâteaux-différentiable sur C , avec C convexe. J est convexe si et seulement si ∇J est un opérateur monotone, c'est-à-dire

$$\forall (u, v) \in C \times C \quad \langle \nabla J(u) - \nabla J(v), u - v \rangle \geq 0.$$

1.3.3 Minimisation dans un Banach Réflexif

Définition 1.3.3 ([6]) Soit J une fonctionnelle semi-continue sur un ensemble fermé de V . On dit que $J : V \rightarrow \mathbb{R}$ est coercive si

$$\lim_{\|x\|_V \rightarrow +\infty} J(x) = +\infty.$$

Théorème 1.3.4 ([6]) On suppose que V est un Banach réflexif. Soit J une fonctionnelle de V dans \mathbb{R} , semi-continue inférieurement pour la topologie faible de V . Soit K un sous-ensemble non vide et faiblement fermé de V . On suppose que J est propre (qu'il existe un élément v_0 de K tel que $J(v_0) < +\infty$). Alors le problème de minimisation suivant :

$$(P) \quad \begin{cases} \text{Trouver } u \text{ tel que} \\ J(u) = \inf \{J(v) \mid v \in K\}, \end{cases}$$

admet au moins une solution dans l'un des cas suivants :

- 1 soit J est coercive c'est à dire $\lim_{\|v\|_V \rightarrow +\infty} J(v) = +\infty$;
- 2 soit K est borné.

Corollaire 1.3.2 ([6]) On suppose que V est un Banach réflexif. Soit J une fonctionnelle de V dans \mathbb{R} , convexe semi-continue inférieurement sur V propre et K un sous-ensemble convexe non vide et fermé de V . Si J est coercive ou si K est borné, le problème de minimisation admet une solution. Si, de plus, J est strictement convexe la solution est unique.

Théorème 1.3.5 ([6]) Soient K un sous-ensemble convexe, non vide de V et J une fonctionnelle de K vers \mathbb{R} Gâteaux-différentiable sur K . Soit u dans V une solution du problème (P). Alors

$$\forall v \in K \quad \langle \nabla J(u), v - u \rangle \geq 0.$$

1.3.4 Les Projections

Projection sur un sous-espace vectoriel

Dans le cas où C est un sous-espace vectoriel fermé de V , c'est bien sûr un convexe fermé non vide. L'opérateur de projection est dans ce cas linéaire (c'est faux dans le cas général). Le projeté x^* (d'un élément x sur C , est caractérisé par $\forall y \in C \quad \langle x - x^*, y \rangle = 0$.) Cela signifie que $x - x^* \in C^\perp$ (l'orthogonal de C). On retrouve ainsi la classique projection orthogonale sur un sous-espace vectoriel fermé.

Projection sur un convexe fermé

Soit V un espace de Hilbert muni d'un produit scalaire $\langle \cdot, \cdot \rangle$ et de la norme associée $\|\cdot\|$ et C est un sous-ensemble convexe et fermé de V .

Théorème 1.3.6 [57] [30] [34] (*La projection*)

Soit C une partie non-vide convexe fermée d'un espace de Hilbert H . Alors, pour tout $x \in H$, il existe $x_0 \in C$ unique tel que

$$\|x - x_0\|_H = \inf \{\|x - a\|, a \in C\},$$

le point x_0 s'appelle la projection de x sur C .

Théorème 1.3.7 ([6]) Etant donné C un sous-ensemble convexe, fermé et non vide de V et x un élément quelconque de V . Alors le problème

$$\min \{\|x - y\|^2, y \in C\};$$

a une solution unique $x^* \in C$. De plus $x^* \in C$ est caractérisé par :

$$\forall y \in C \quad \langle x - x^*, y - x^* \rangle \leq 0.$$

Le point x^* est le projeté de x sur C . L'application $\pi_C : V \rightarrow C$ qui à x associe son projeté x^* est la projection sur C . Le projeté $\pi_C(x)$ est donc le point de C qui est le plus près de x .

On définit de manière standard la fonction distance d'un point x à l'ensemble C par

$$d(x, C) = \inf_{y \in C} \|x - y\|,$$

Dans le cas où C est un convexe fermé, on vient de démontrer que

$$d(x, C) = \|x - \pi_C(x)\|.$$

Définition 1.3.4 ([6]) (*L'opérateur de projection métrique*) Soit X un espace réel normé et $C \subset X$ non vide; l'opérateur de projection métrique sur C est noté par :

$$\begin{aligned} \pi_C & : X \rightarrow C \\ x & \rightarrow \pi_C(x) = \{z \in C, \|x - z\| = \inf_{y \in C} \|x - y\|\} \forall x \in X. \end{aligned}$$

(1.2)

Proposition 1.3.1 ([6]) La projection π_C est continue de V dans C . Plus précisément, π_C est une contraction dans le sens

$$\forall (x, y) \in V \times V \quad \|\pi_C(x) - \pi_C(y)\| \leq \|x - y\|.$$

Remarque 1.3.1 Si $x \in C$ alors $\pi_C(x) = x$. Plus généralement si $C = V$ alors $\pi_C(x) = Id_V$. La projection π_C n'est pas différentiable en général, mais l'application $x \rightarrow \|x - y\|^2$ l'est.

Projection de Bregman et projection oblique

Définition 1.3.5 ([38]) Soit $g : X \rightarrow \mathbb{R}$ une fonction convexe, $u \in X, p \in X^*$. On appelle p le sous-gradient de g de u si :

$$\forall v \in X : J(v) - J(u) - \langle p, v - u \rangle \geq 0.$$

L'ensemble de tous les sous-gradients de g en u , est appelé le sous-différentiel de g noté $\partial g(u)$.

Définition 1.3.6 ([38]) (**Distance de Bregman**) Supposons $g : X \rightarrow \mathbb{R}$ une fonction convexe, $u, v \in X$, et $p \in \partial g(u)$ est un sous-gradient du sous-différentiel de g au point u , alors la distance de Bregman entre les points u, v est définie par

$$D_g^p(u, v) = g(u) - g(v) - \langle p, u - v \rangle.$$

Définition 1.3.7 ([38]) La fonction distance de Bregman sur C est définie par

$$D_C^g(x) := \inf_{y \in C} D_g(y, x), \quad \forall x \in \text{int}(\text{dom}(g)). \quad (1.3)$$

Définition 1.3.8 ([38, 4]) (**Fonction de Legendre**) Supposons que g est une fonction convexe sur $(X \subset \mathbb{R}^d \text{ espace euclidien})$ avec $\text{int}(\text{dom}(g)) \neq \emptyset$. Alors g est une fonction convexe de type Legendre, si elle remplit chacune des conditions suivantes :

1. sur $\text{int}(\text{dom}(g))$, g est différentiable ;
2. $\lim_{\lambda \rightarrow 0^+} \langle \nabla g(x + \lambda(y - x)), y - x \rangle = -\infty, \forall x \in \text{bd}(\text{dom}(g)), y \in \text{int}(\text{dom}(g))$;
3. sur $\text{int}(\text{dom}(g))$, g est strictement convexe.

Définition 1.3.9 ([38]) (**Projection de Bregman**) Soit $g : X \rightarrow \mathbb{R}$ une fonction de Legendre. Soit C un ensemble convexe fermé dans X avec $C \cap \text{int}(\text{dom}(g)) \neq \emptyset$. Alors le problème d'approximation $\inf_{y \in C} D_g(y, x)$ admet une solution unique $\text{proj}_C^g(x)$ appelé la projection de Bregman de x sur C qui est définie par

$$\begin{aligned} \text{proj}_C^g(x) &:= \arg \min_{y \in C} D_g(y, x) \in \text{int}(\text{dom}(g)), \\ \text{proj}_C^g(x) &:= \{y \in C : D_g(y, x) = D_C^g(x)\} \quad \forall x \in \text{int}(\text{dom}(g)). \end{aligned} \quad (1.4)$$

Définition 1.3.10 ([16]) (**La projection oblique**) Soit A une matrice réelle de taille $M \times N$ et soient C et Q des ensembles convexes fermés non vides dans \mathbb{R}^N et \mathbb{R}^M , respectivement. Lorsque $M = N$ et A est inversible, l'application

$$P_{A^{-1}(Q)}^{A^T A} = A^{-1} P_Q A; \quad (1.5)$$

est une projection oblique sur l'ensemble $A^{-1}(Q)$; c'est-à-dire :

$$A^{-1} P_Q (Ax) := \min \{f(z) = (z - x)^T A^T A (z - x)\}; \quad \forall z \in A^{-1}(Q).$$

1.4 Problème Inverse et Problème Mal Posé

Soient E, F deux espaces de Banach et $A : D(A) \subset E \rightarrow F$ un opérateur linéaire ou non. Le problème $Ax = y$ est bien posé au sens de Hadamard si les conditions suivantes sont vérifiées :

1. **existence** : pour tout $y \in F$, il existe $x \in E$ tel que $Ax = y$ (A surjectif) $Im(A) = F$ pour tout $y \in F$;
2. **unicité** : pour tout $y \in F$, il existe au plus une solution $x \in E$ (A injectif) $\ker(A) = 0$. C'est-à-dire pour tout $x_1, x_2 \in E$ et $Ax_1 = Ax_2$ alors $x_1 = x_2$;
3. **stabilité** : la solution x dépend continument de la donnée y . $Im(A) = \overline{Im(A)} = \ker(A^*)^\perp$. C'est-à-dire faible perturbation de donnée y donne faible perturbation de la solution x .

Pour tout $(x_n)_n \subset E$ si $y_n = Ax_n$ alors $Ax = y$ implique $x_n \rightarrow x$ c'est-à-dire A^{-1} borné.

problème direct : Etant donné les causes, trouver l'effet $Ax = y$.

problème inverse : Etant donné les l'effets, trouver les causes $x = A^{-1}y$.

1.5 Traitement Numérique des Images

Aujourd'hui, pratiquement toutes les interprétations et analyses d'images impliquent un certain type de traitement numérique car la majorité des données sont stockées en format numérique. L'amélioration numérique pour faciliter l'interprétation visuelle, le formatage et la correction des données, ou même la classification automatique des cibles et des structures entièrement par ordinateur ne sont que quelques-unes des méthodes qui peuvent être utilisées dans le traitement d'images numériques.

1.5.1 Les Images Numérique

Les images sont des représentations visuelles de scènes, d'objets ou d'informations capturées à l'aide de différents dispositifs, tels que des caméras ou des scanners. Le format d'une image peut varier en fonction de l'application et des exigences spécifiques. Certains formats d'image couramment utilisés comprennent *JPEG*, *PNG*, *GIF*, *TIFF*, *BMP*, etc. Chaque format peut avoir des caractéristiques et des propriétés spécifiques, tels que la compression, la transparence, la gestion des couleurs, etc. Une image numérique est composée de pixels, qui sont les plus petites unités d'information dans une image. Chaque pixel contient des valeurs qui représentent l'intensité lumineuse ou la couleur d'un point spécifique de l'image. Les images en niveaux de gris ont un seul canal de couleur représentant les niveaux de gris, tandis que les images en couleur ont plusieurs canaux de couleur (généralement rouge, vert et bleu) qui se combinent pour former une image couleur.

Les images sont omniprésentes dans notre vie quotidienne et jouent un rôle essentiel dans la communication visuelle, la documentation, la création artistique et de nombreux autres

domaines. Grâce aux avancées technologiques, les images numériques sont devenues plus accessibles, plus faciles à manipuler et à partager, et elles jouent un rôle central dans notre société moderne.

Définition 1.5.1 ([6]) Une image numérique est composée d'unités de base (appelées pixels) qui représentent chacune une zone spécifique de l'image. Une image est définie par

- * le nombre total de pixels dans sa largeur et sa hauteur (qui peut varier à l'infini);
- * l'étendue des teintes de gris ou des couleurs différentes dont chaque pixel est capable (appelé dynamique d'image).

Il existe trois catégories d'images numériques :

1. **les images binaires** : Images les plus simples, un pixel ne peut avoir qu'une valeur de noir ou de blanc dans les graphiques les plus basiques. Lorsque le texte est composé d'une seule couleur, c'est souvent le type d'image utilisé pour l'analyser;
2. **les images en teintes de gris** : En général, il y a 256 nuances de gris différentes dans une image typique. Chacune de ces 256 couleurs est définie dans la gamme des gris. Par convention la valeur 0 représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale);
3. **les images couleurs** : Pour représenter les couleurs rouge, vert et bleu, une image couleur est en fait composée de trois images (R,V,B). Ces trois images sont appelées chacune un canal.

Définition 1.5.2 ([1]) Une image numérique est définie comme une fonction $f(m, n, \dots)$ à valeurs discrètes et finies (mais pas forcément scalaires), et dont les coordonnées m, n, \dots sont également à valeurs discrètes et finies.

$$\begin{cases} f : \mathbb{N}^d \longrightarrow \mathbb{R}^B \\ (m, n, \dots) \longmapsto f(m, n, \dots) \end{cases}$$

1. Une image à niveaux de gris correspond à $d = 2$ et $B = 1$ (il n'y a qu'une valeur par pixel : le niveau de gris).
2. Une image couleur classique et à $d = 2$ et $B = 3$ (il y a trois bandes : Rouge, Vert, Bleu).
3. Une image IRM "cas médical" correspond à $d = 3$ (l'image est 3D) et $B = 1$.

Définition 1.5.3 ([1]) Le pixel est l'abréviation de picture element; élément d'image. C'est la primitive de bas niveau la plus pauvre en information car ses seuls attributs sont la position dans la matrice image (n de ligne, m de colonne) et la valeur numérique indiquant sa couleur, ou son niveau de gris. Il peut être représenté en mémoire sur :

- a- un bit (0 ou 1) pour les images monochromes : 0 pour le noir et 1 pour blanc;
- b- un octet, soit 256 niveaux de gris pour une image à niveaux de gris : 0 noir et 255 blanc.

Définition 1.5.4 ([1]) Le bruit est une information parasite s'ajoutant de façon aléatoire à la scène. Particulièrement visible dans :

- * les zones où le rapport signal /bruit est faible (ex zones peu éclairées);
- * les zones uniformes.

Le bruit est de nature et d'origine variées et donc se modélise de différentes façons par exemple : **bruit additif gaussien** (centré) d'écart-type σ souvent égal au SNR. C'est le modèle le plus courant.

Si u représente l'image originale, l'image dégradée u_b s'obtient par $u_b = u + b_\sigma$, où b_σ est une variable aléatoire de loi gaussienne centrée et d'écart-type σ .

Définition 1.5.5 ([1]) *Les contours d'une image numérique sont les lieux de variations significatives de l'information niveau de gris (il y a très peu de travaux sur les contours dans les images couleurs ou multi-spectrales).*

1.5.2 L'Imagerie Médicale

L'imagerie médicale est un domaine de la médecine qui utilise diverses techniques d'imagerie pour visualiser l'intérieur du corps humain, permettant ainsi de diagnostiquer des maladies, de surveiller leur progression et d'évaluer l'efficacité des traitements.

Voici quelques-unes des principales techniques d'imagerie médicale utilisées :

1. **radiographie** : Les radiographies utilisent des rayons X pour produire des images bidimensionnelles des structures internes du corps, telles que les os et les poumons. Elles sont couramment utilisées pour détecter les fractures, les infections pulmonaires, les tumeurs osseuses, etc ;
2. **tomodensitométrie (TDM)** : La TDM, également appelée scanner, utilise des rayons X pour obtenir des images transversales détaillées du corps. Elle est utile pour détecter les anomalies dans les organes internes, les vaisseaux sanguins, les tumeurs, les caillots sanguins, etc ;
3. **Imagerie par résonance magnétique (IRM)** : L'IRM utilise des champs magnétiques et des ondes radio pour créer des images détaillées des organes et des tissus du corps. Elle est particulièrement efficace pour visualiser le cerveau, la moelle épinière, les articulations, les organes abdominaux, etc ;
4. **échographie** : L'échographie utilise des ondes sonores pour produire des images en temps réel des organes et des tissus internes. Elle est souvent utilisée pour examiner les organes abdominaux, le cœur, les vaisseaux sanguins, le fœtus pendant la grossesse, etc ;
5. **imagerie par résonance magnétique nucléaire (IRMN)** : L'IRMN est une technique d'imagerie avancée qui utilise des isotopes radioactifs pour marquer des molécules spécifiques dans le corps. Elle est utilisée pour étudier le métabolisme, la perfusion sanguine, la fonction cérébrale, etc ;
6. **tomographie par émission de positons (TEP)** : La TEP utilise des traceurs radioactifs pour visualiser les processus métaboliques dans le corps. Elle est principalement utilisée pour la détection précoce du cancer, l'évaluation de l'activité tumorale, etc.

Ces techniques d'imagerie médicale fournissent des informations précieuses aux médecins pour poser des diagnostics, planifier des traitements et surveiller l'évolution des maladies. Chaque technique a ses avantages et ses applications spécifiques, et leur utilisation dépend de la condition clinique et des besoins du patient.

1.6 Des Mesures

1.6.1 Erreur quadratique moyenne

L'erreur quadratique moyenne est la méthode où nous pouvons comparer image restaurée à l'image d'origine, et calculer les différentes erreurs entre elles. L'erreur quadratique moyenne généralement définie comme erreur quadratique cumulée entre l'image restaurée et image originale. Elle a la forme :

$$MSE = \frac{1}{MN} \sum_{j=1}^M \sum_{k=1}^N (x_{j,k} - x'_{j,k})^2.$$

1.6.2 Rapport signal/bruit de crête

Fondamentalement, le rapport signal-bruit de crête défini comme l'expression de rapport entre la puissance maximale du signal et la puissance du signal image bruitée. Le résultat ou l'expression doit être en décibel échelle (dB).

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} = 10 \log_{10} \left(\frac{R^2}{MSE} \right),$$

où R est la puissance maximale de l'image du signal. Le $PSNR$ est calculé sur la base de la MSE.

1. $PSNR < 0 \implies$ la restauration est mauvaise.
2. $PSNR = 0 \implies$ il n'y a pas de restauration (on reste proche de l'image bruitée).
3. $PSNR > 0 \implies$ la restauration est de qualité bonne.

1.6.3 Erreur absolue normalisée

L'erreur absolue normalisée généralement définie comme :
 $NAE = \text{sum}(\text{sum}(\text{abs}(\text{error}))) / \text{sum}(\text{sum}(\text{origImg}))$. Elle a la forme :

$$NAE = \frac{\sum_{j=1}^M \sum_{k=1}^N |x_{j,k} - x'_{j,k}|}{\sum_{j=1}^M \sum_{k=1}^N |x_{j,k}|}.$$

1.6.4 Différence moyenne

Différence moyenne généralement défini comme :
 $AD = \text{sum}(\text{sum}(\text{error})) / (M \times N)$. Elle a la forme :

$$AD = \frac{\sum_{j=1}^M \sum_{k=1}^N (x_{j,k} - x'_{j,k})}{M \times N}.$$

1.6.5 Différence maximale

Différence maximale généralement définie comme :

$MD = \max(error)$. Elle a la forme :

$$MD = \max \left(\left| x_{j,k} - x'_{j,k} \right| \right).$$

1.6.6 Corrélation croisée normalisée

Corrélation croisée normalisée généralement définie comme :

$NK = \frac{\text{sum}(\text{sum}(\text{origImg} * \text{distImg}))}{\text{sum}(\text{sum}(\text{origImg} * \text{origImg}))}$. Elle a la forme :

$$NK = \frac{\sum_{j=1}^M \sum_{k=1}^N (x_{j,k} \times x'_{j,k})}{\sum_{j=1}^M \sum_{k=1}^N x_{j,k}^2}.$$

1.6.7 Contenu structurel

Contenu structurel généralement défini comme :

$SC = \frac{\text{sum}(\text{sum}(\text{origImg} * \text{origImg}))}{\text{sum}(\text{sum}(\text{distImg} * \text{distImg}))}$. Il a la forme :

$$SC = \frac{\sum_{j=1}^M \sum_{k=1}^N x_{j,k}^2}{\sum_{j=1}^M \sum_{k=1}^N x'_{j,k}^2}.$$

Afin de quantifier la qualité de la restauration pour une image bruitée, on utilise parfois une mesure. On note $x_{j,k}$ l'image originale et $x'_{j,k}$ l'image restaurée avec $[M \ N]$ la taille des images.

Mean Square Error	$MSE = \frac{1}{MN} \sum_{j=1}^M \sum_{k=1}^N (x_{j,k} - x'_{j,k})^2$
Peak Signal to Noise Ratio	$PSNR = 10 \log \frac{(2^n - 1)^2}{MSE} = 10 \log \frac{255^2}{MSE}$
Normalized Cross-Correlation	$NK = \frac{\sum_{j=1}^M \sum_{k=1}^N x_{j,k} \cdot x'_{j,k}}{\sum_{j=1}^M \sum_{k=1}^N x_{j,k}^2}$
Average Difference	$AD = \frac{\sum_{j=1}^M \sum_{k=1}^N (x_{j,k} - x'_{j,k})}{MN}$
Structural Content	$SC = \frac{\sum_{j=1}^M \sum_{k=1}^N x_{j,k}^2}{\sum_{j=1}^M \sum_{k=1}^N x'_{j,k}^2}$
Maximum Difference	$MD = \text{Max} \left(x_{j,k} - x'_{j,k} \right)$
Laplacian Mean Square Error	$LMSE = \frac{\sum_{j=1}^M \sum_{k=1}^N [O(x_{j,k}) - O(x'_{j,k})]^2}{\sum_{j=1}^M \sum_{k=1}^N [O(x_{j,k})]^2}$ $O(x_{j,k}) = x_{j+1,k} + x_{j-1,k} + x_{j,k+1} + x_{j,k-1} - 4x_{j,k}$
Normalized Absolute Error	$NAE = \frac{\sum_{j=1}^M \sum_{k=1}^N x_{j,k} - x'_{j,k} }{\sum_{j=1}^M \sum_{k=1}^N x_{j,k} }$

Figure 1.1 – Mesures de qualité d'image.

Chapitre 2

La Méthode de Régularisation de Tychonov

2.1 Introduction Historique

La régularisation de Tykhonov (ou Tichonov) a été inventée indépendamment dans divers contextes. Elle est largement connue grâce à son application aux équations intégrales grâce aux travaux (d'Andrey Tikhonov et de David L. Phillips). Dans la littérature mathématique, plusieurs méthodes de régularisation ont été utilisées pour résoudre certains problèmes mal posés. Parmi elles, on a la méthode de régularisation de Tykhonov ([55]) ; c'est la plus ancienne méthode régulière utilisée pour résoudre des problèmes mal posés ainsi que des problèmes inverses. Il s'agit de convertir le problème original mal-posé en un problème de minimisation. Elle a été imaginée par le mathématicien russe Andreï Nikolaïevitch Tykhonov.

Un certain nombre de mathématiciens tels que Tykhonov, John, Lavrentiev, Ivanov et d'autres ont travaillé pour développer des théories et des méthodes de résolution des problèmes difficiles mal posés. La régularisation des problèmes mal posés de Tykhonov cherche à redéfinir les notions de solutions (quasi-solution, solution approchée, image, ...), de sorte que "la solution régularisée" obtenue est une solution stable et proche de la solution exacte (en supposant que la solution suivante existe pour des données proches des valeurs effectivement obtenues par la mesure). En d'autres termes, nous remplaçons le problème original mal posé par un autre problème d'approximation bien posé.

Les problèmes inverses forment un domaine vaste de recherche depuis de nombreuses années. L'importance de ce domaine est due à une large classe d'applications telles que le traitement d'images. Tykhonov a initié la recherche sur les méthodes stables pour la résolution numérique des problèmes inverses. La théorie de la régularisation de Tykhonov a été développée systématiquement. Jusqu'aux années 1980, on a réussi une analyse rigoureuse et assez complète des méthodes de régularisation pour la résolution de problèmes non linéaires. Vers 1989, une théorie de régularisation pour les problèmes inverses non linéaires a été développée. À peu près à la même époque qu'a été développée la régularisation par TV pour le débruitage de données discontinues. La règle principale de Tykhonov a été inventée indépendamment dans un certain nombre de contextes différents. Elle est devenue très populaire grâce à sa demande d'équations intégrales des œuvres d'Erandy Tykhonov et de David L. Phillips. Certains au-

teurs utilisent les termes réguliers de Tykhonov-Phillips. Le cas de la taille finie de problème est expliqué par Arthur E. Hoerl, qui a mis en œuvre la méthode statistique, et Mans Foster, qui a utilisé la méthode comme filtre de Wiener-Kolmogorov. Ainsi, selon Hoerl, il est connu dans la littérature statistique sous le nom de régression de crête.

2.2 Régularisation Linéaire (Tykhonov)

La méthode de régularisation de Tykhonov est la plus ancienne méthode régulière utilisée pour résoudre des problèmes inverses. Il s'agit de convertir le problème de négligence initial en un problème d'atténuation. Il a été imaginé par le mathématicien russe Andreï Nikolaïevitch Tykhonov. La régularisation des faux problèmes de Tykhonov vise à redéfinir les notions d'inversion et d'approximation afin que la solution de régularisation obtenue par régularisation dépende continûment des données et proche de la solution exacte (en supposant qu'elle existe pour des données proches des valeurs réelles obtenues par la mesure). En d'autres termes, nous remplaçons le problème mal posé d'origine par un autre problème d'approximation bien posé. C'est l'une des méthodes de régularisation la plus connue, tant en statistique qu'en analyse numérique. On résout un problème de type général :

$$y = \min_{x \in H} \left\{ \underbrace{\|Au - f\|}_{\text{Attaché aux données}}^2 + \lambda \underbrace{\|\nabla u\|}_{\text{Terme de régularisation}}^2 \right\}.$$

Considérons le problème inverse $Kh_1 = h_2$ où $K : H_1 \rightarrow H_2$ est un opérateur compact injectif. On suppose que $h_2 \in \text{Im}(K)$, c'est-à-dire le problème inverse possède une solution unique.

Définition 2.2.1 (*Une Stratégie de Régularisation*) Une famille d'opérateurs linéaires et bornés

$$\{R_\alpha\}_{\alpha \geq 0} : H_2 \rightarrow H_1,$$

est dite "famille régularisante" pour l'opérateur K si

$$\forall h_1 \in H_1, \lim_{\alpha \rightarrow 0} (R_\alpha K)h_1 = h_1, \text{ i.e. } R_\alpha K \text{ converge simplement vers } I. \quad (2.1)$$

Si $\{R_\alpha\}_{\alpha \geq 0}$ est une famille régularisante pour l'opérateur $K : H_1 \rightarrow H_2$, H_1 de dimension infinie, alors les opérateurs R_α ne sont pas uniformément bornés c'est-à-dire, il existe une suite $(\alpha_n) \subset \mathbb{R}_+$ telle que $\lim_{n \rightarrow \infty} \|R_{\alpha_n}\| = +\infty$.

Définition 2.2.2 Une stratégie de régularisation $\delta \rightarrow \alpha(\delta)$ est admissible si pour tout $h_1 \in H_1$

$$\lim_{\delta \rightarrow 0} \alpha(\delta) = 0 \text{ et } \lim_{\delta \rightarrow 0} \left(\sup_{h_2^\delta \in H_2} \{ |R_{\alpha(\delta)} h_2^\delta - h_1| \text{ tel que } \|K h_1 - h_2^\delta\| \leq \delta \} \right) = 0. \quad (2.2)$$

Les stratégies de régularisation sont variées. Chaque problème nécessite un traitement spécifique selon son degré de complexité.

2.2.1 La Méthode de Tykhonov en EDP

Parmi les méthodes les plus connues en problème inverse, on a la méthode de Tykhonov. Soient H_1 et H_2 deux espaces de Hilbert, supposons que $K : H_1 \rightarrow H_2$, $\|A\| < \infty$ est linéaire, $y \in \text{Im}(A)$ telle que $\|y_\delta - y\| \leq \delta$, y_δ n'est pas nécessaire dans $\text{Im}(A)$, et soit le problème

$$Ax = y.$$

Etant donné y_δ , on veut donc construire une approximation raisonnable x_α^δ de la solution exacte x de l'équation non perturbée $Ax = y$. On veut aussi que cette approximation soit stable, i.e, x_α^δ dépende continuellement des données y_δ . Donc on cherche une approximation de l'opérateur inverse non borné $A^{-1} : \text{Im}(A) \rightarrow H_1$ par un opérateur linéaire borné $R_\alpha : H_2 \rightarrow H_1$.

Théorème 2.2.1 ([44]) Soient H_1, H_2 deux espaces de Hilbert et $A : H_1 \rightarrow H_2$ un opérateur linéaire compact et soit $\alpha > 0$. Alors pour chaque $y \in H_2$, il existe $x_\alpha \in H_1$ telle que

$$\|Ax_\alpha - y\|^2 + \alpha \|x_\alpha\|^2 = \inf_{x \in H_1} \|Ax - y\|^2 + \alpha \|x\|^2. \quad (2.3)$$

Le minimiseur x_α est donné par l'unique solution de

$$\alpha x_\alpha + A^*Ax_\alpha = A^*y, \quad (2.4)$$

et dépend continuellement de y .
En outre, il en résulte que

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

et

$$R_\alpha : H_2 \rightarrow H_1$$

$$y \mapsto R_\alpha(y) = x_\alpha$$

où $R_\alpha = (\alpha I + A^*A)^{-1}A^*$.

Théorème 2.2.2 ([44]) Soit $A : H_1 \rightarrow H_2$ un opérateur linéaire compact. Alors pour tout $\alpha > 0$, l'opérateur $(\alpha I + A^*A) : H_1 \rightarrow H_1$ est bijectif et a un inverse borné. De plus, si A est injectif alors

$$R_\alpha = (\alpha I + A^*A)^{-1}A^*, \quad (2.5)$$

décrit un schéma de régularisation avec $\|R_\alpha\| \leq \frac{1}{2\sqrt{\alpha}}$.

Choix du paramètre de régularisation

(i) Méthode de Mozorov : ([44])

Dans la méthode de Mozorov, le paramètre α est choisi tel que :

$$\|A u_\alpha^\delta - y_{(\delta)}\| = \delta. \quad (2.6)$$

(ii) Méthode d'Arcangeli : ([44])

L'équation satisfaite est :

$$\|A u_\alpha^\delta - y_{(\delta)}\| = \frac{\delta}{\sqrt{\alpha}}. \quad (2.7)$$

2.3 La Méthode de Tykhonov Généralisé

La récupération d'une fonction ou d'un vecteur de paramètre de grande dimension ou image numérique à partir de mesures indirectes est une tâche centrale dans divers domaines scientifiques. Plusieurs méthodes pour résoudre de tels problèmes inverses sont bien développées et bien comprises. Récemment, de nouveaux algorithmes utilisant l'apprentissage en profondeur et les réseaux de neurones pour les problèmes inverses sont apparus. Cependant, il existe des résultats théoriques pour l'apprentissage en profondeur dans les problèmes inverses. Dans ([44]), ils établissent une analyse de convergence complète pour l'approche **NETT** (Network Tikhonov) proposée pour les problèmes inverses. **NETT** considère des solutions cohérentes avec les données ayant une petite valeur d'un régularisateur défini par un réseau de neurones formé. Ils dérivent des résultats bien posés et des estimations d'erreur quantitatives, et proposent une stratégie possible pour la formation du régularisateur. Dans ([39]), les résultats numériques sont présentés pour un problème de données éparées tomographiques, qui donnent de bonnes performances de **NETT** même pour des inconnues de type différent des données d'apprentissage. Pour dériver les résultats de convergence et le taux de convergence, ils introduisent un nouveau cadre basé sur la distance de Bregman absolue généralisant la distance de Bregman standard du convexe au non-convexe cas.

La régularisation de type Tikhonov généralisée pour des problèmes mal posés de grande échelle a été proposée par ([59]). Ils introduisent la méthode de projection de la bidiagonalisation itérative de Lanczos et la factorisation QR ([59]) qui est choisie pour résoudre le type général, et montrent que le paramètre de régularisation peut être choisi sans connaissance préalable de la variance du bruit en utilisant la méthode du principe d'équilibrage. La méthode itérative est plus facile pour calculer les paramètres de régularisation basés sur le type général. Il est très important de déterminer une règle d'arrêt fiable qui peut être partiellement choisie en combinant la méthode de projection du sous-espace de Krylov avec la convergence de l'algorithme de régularisation. Un algorithme implique la réalisation numérique efficace de la nouvelle règle de choix.

Un problème crucial concernant la régularisation de Tikhonov est le bon choix du paramètre de régularisation. Dans ([58]), ils ont présenté une généralisation d'une règle de choix de paramètre due à Reginska (1996), qui a abouti à un algorithme à virgule fixe pour la régularisation Tikhonov multiparamètre appelé **MFP**. Comme dans le cas à paramètre unique, l'algorithme ne nécessite aucune information sur le niveau de bruit. De plus, en combinant la projection sur le sous-espace de Krylov généré par l'algorithme de bidiagonalisation de Golub-Kahan (GKB) et la méthode **MFP** à chaque itération, ils dérivent un nouvel algorithme GKB-FP pour les problèmes de régularisation de Tikhonov multi-paramètres à grande échelle. La performance du **MFP** lorsqu'il est appliqué à des problèmes discrets mal posés bien connus sont évalués et comparés aux résultats obtenus par le principe de divergence. Les résultats indiquent que la MFP est efficace et compétitive. Les résultats numériques montrent que la régularisation multi-paramètres de Tikhonov peut améliorer significativement la qualité de la formulation standard de Tikhonov à un paramètre.

La régularisation de Tikhonov itérée généralisée avec des semi-normes opérateur-dépendantes a été proposée par ([8]). Ils introduisent et analysent les contreparties itératives stationnaires et non stationnaires de la méthode de Tikhonov généralisée avec des semi-normes générales

dépendantes de l'opérateur dans le terme de pénalité. Ils établissent leur taux de convergence dans des conditions affectant uniquement les paramètres d'itération, prouvant qu'ils surmontent le résultat de saturation.

Dans cette section, nous rappelons deux méthodes de régularisation de Tikhonov généralisées.

2.3.1 Deux Méthodes de Tikhonov Généralisées

Dans ([8]), on considère une équation de la forme

$$Ax = y, \quad (2.8)$$

où $A : X \rightarrow Y$ est un opérateur linéaire compact entre les espaces génériques de Hilbert X et Y sur \mathbb{R} ou \mathbb{C} . Soit $Rg(A)$ le domaine de A , on suppose $y \in Rg(A)$, c'est-à-dire que le problème $Ax = y$ a une solution $x^+ = A^+y$ de norme minimale. Ici, A^+ désigne l'opérateur inverse généralisé (Moore-Penrose) de A . Le problème $Ax = y$ est donc mal posé et doit être régularisé afin de calculer une solution numérique. Nous voulons approximer x^+ la solution de l'équation $Ax = y$, lorsque la solution y^δ de y est donnée par

$$y^\delta = y + \eta \text{ et } \|\eta\| \leq \delta$$

où η est appelé le vecteur de bruit et δ est appelé le niveau de bruit. Puisque A^+y^δ n'est pas une bonne approximation de x^+ , Nous approximations x^+ avec $x_\alpha^\delta := R_\alpha y^\delta$ où $\{R_\alpha\}$ est une famille d'approximation continues d'opérateurs dépendant d'un paramètre α . Une classe de tel régularisation sont les méthodes dites basées sur des filtres qui exploitent devalueent en valeur singulière (*sve*) de l'opérateur A , agissant sur son spectre afin de diminuer l'effet du bruit sur la solution reconstruite. En effet, si on désigne par $(\sigma_m; v_m, u_m)_{m \in \mathbb{N}}$ les (*sve*) de A , où $\{F_\alpha\}_{\alpha \in \mathbb{R}^+}$ est une famille de fonctions (filtres). On peut écrire :

$$x_\alpha^\delta = x^+ - e_a + e_n,$$

avec

$$e_a = \sum_{m:\sigma_m>0} (1 - F_\alpha(\sigma_m)) \sigma_m^{-1} \langle y, u_m \rangle v_m,$$

est l'erreur d'approximation et

$$e_n = \sum_{m:\sigma_m>0} F_\alpha(\sigma_m) \sigma_m^{-1} \langle \eta, u_m \rangle v_m$$

est l'erreur de bruit.

Le rôle de la fonction de filtre F_α est alors de faire la médiation entre l'erreur d'approximation et l'erreur de bruit.

Dans cette section, nous discutons deux types récents de méthodes de régularisation qui généralisent la méthode classique de Tikhonov. Nous utiliserons la notation $F_{\alpha,\cdot}$, pour indiquer les nouveaux filtres, où \cdot sera remplacé par le paramètre supplémentaire introduit par la méthode choisie.

Définition 2.3.1 ([8]), On appelle méthode Tikhonov pondérée-I, une méthode basée sur le filtre

$$R_{\alpha,r}y := \sum_{m:\sigma_m>0} F_{\alpha,r}(\sigma_m) \sigma_m^{-1} \langle y, u_m \rangle v_m,$$

où la fonction filtre est

$$F_{\alpha,r}(\sigma) = \frac{\sigma^{r+1}}{\sigma^{r+1} + \alpha} = \frac{\sigma^2}{\sigma^2 + \alpha\sigma^{1-r}}, \quad (2.9)$$

pour $\alpha > 0$ et $r \geq 0$. Pour $r = 1$ le filtre de Tikhonov classique est récupéré. On note :

$$\begin{aligned} x_{\alpha,r} & : = R_{\alpha,r}y, \quad y \in \text{Dom}(A^+), \\ x_{\alpha,r}^\delta & : = R_{\alpha,r}y^\delta, \quad y^\delta \in Y. \end{aligned}$$

Définition 2.3.2 ([8]), On appelle méthode de Tikhonov pondérée-II, une méthode basée sur le filtre

$$R_{\alpha,j}y := \sum_{m:\sigma_m>0} F_{\alpha,j}(\sigma_m) \sigma_m^{-1} \langle y, u_m \rangle v_m,$$

où la fonction filtre est

$$F_{\alpha,j}(\sigma) = \frac{\sigma^2}{\sigma^2 + \alpha \left[1 - \left(\frac{\sigma}{\sigma_1} \right)^2 \right]^j}, \quad (2.10)$$

pour $\alpha > 0$ et $j \in \mathbb{N}$. Pour $j = 0$ le filtre de Tikhonov classique est récupéré. On note :

$$\begin{aligned} x_{\alpha,j} & : = R_{\alpha,j}y, \quad y \in \text{Dom}(A^+), \\ x_{\alpha,j}^\delta & : = R_{\alpha,j}y^\delta, \quad y^\delta \in Y. \end{aligned}$$

Étant donné un opérateur W sur tout espace de Hilbert, si l'on considère la semi-norme $\|\cdot\|_W$ induite par $\|x\|_W^2 := \langle Wx, Wx \rangle$, alors la méthode de Tikhonov pondérée-I peut également être définie comme l'unique minimiseur du fonctionnelle,

$$R_{\alpha,r}y := \arg \min_{x \in X} \{ \|Ax - y\|^2 + \alpha \|x\|_W^2 \}, \quad (2.11)$$

où la semi-norme $\|\cdot\|_W$ est induite par l'opérateur $W := (A^*A)^{\frac{1-r}{4}} : X \rightarrow X$. Pour $r > 1$, W doit être entendu comme l'inverse de Moore-Penrose (pseudo). En recherchant un point stationnaire pour l'équation (2.11), on a

$$R_{\alpha,r}y = \left[(A^*A) + \alpha (A^*A)^{\frac{1-r}{2}} \right]^{-1} A^*y. \quad (2.12)$$

De la même manière, la méthode de Tikhonov pondérée-II peut être définie comme l'unique minimiseur de la fonctionnelle

$$R_{\alpha,j}y := \arg \min_{x \in X} \{ \|Ax - y\|^2 + \alpha \|x\|_B^2 \}, \quad (2.13)$$

où $B := \left(I - \frac{A^*A}{\|A^*A\|} \right)^{\frac{j}{2}} : X \rightarrow X$, en recherchant un point stationnaire pour l'équation (2.13), on a

$$R_{\alpha,j}y = \left[(A^*A) + \alpha \left(I - \frac{A^*A}{\|A^*A\|} \right)^j \right]^{-1} A^*y. \quad (2.14)$$

Les deux méthodes peuvent alors être classées dans le contexte plus général des méthodes inverses généralisées pondérées, à savoir

$$R_\alpha y := \arg \min_{x \in X} \{ \|Ax - y\|^2 + \alpha \|x\|_\Lambda^2 \}, \quad (2.15)$$

ou encore

$$R_\alpha y = [(A^*A) + \alpha(\Lambda^*\Lambda)]^{-1} A^*y. \quad (2.16)$$

où Λ est un opérateur convenable. Remarquons simplement que si $\Lambda^*\Lambda$ et A^*A commutent, alors en désignant par $(\lambda_n; v_n, u_n)_{n \in \mathbb{N}}$ les sve de Λ , l'opérateur (2.16) peut s'écrire

$$R_\alpha y := \sum_{m: \sigma_m > 0} F_\alpha(\sigma_m, \lambda_m) \sigma_m^{-1} \langle y, u_m \rangle v_m, \text{ avec } F_\alpha(\sigma, \lambda) = \frac{\sigma^2}{\sigma^2 + \alpha \lambda^2}. \quad (2.17)$$

Maintenant, soit $f : [0, \sigma_1] \rightarrow [0, \infty)$ une fonction Borel-mesurable et considérons l'opérateur $f(A^*A)$, qui commute avec A^*A . D'après les équations ((2.17), (2.9) et (2.10)), il est clair que les deux méthodes de filtrage pondéré- I et pondéré- II sont de la forme (2.16) avec $\Lambda = \sqrt{f(A^*A)}$ et où

$$f(\sigma^2) = \sigma^{1-r} \text{ et } f(\sigma^2) = \left[1 - \left(\frac{\sigma}{\sigma_1} \right)^2 \right]^j,$$

respectivement. Selon ([8]), ils introduisent une nouvelle méthode **mixte** qui utilise à la fois les méthodes pondérées I et pondérées II en combinant leurs fonctions filtre.

Définition 2.3.3 *Fixant $F_{\alpha,r,j}$ tel que*

$$F_{\alpha,r,j}(\sigma) := \frac{\sigma^2}{\sigma^2 + \alpha f(\sigma^2)} = \frac{\sigma^2}{\sigma^2 + \alpha f_1(\sigma^2) + \alpha f_2(\sigma^2)}, \quad (2.18)$$

avec

$$f_1(\sigma^2) = \sigma^{1-r} \text{ et } f_2(\sigma^2) = \left[1 - \left(\frac{\sigma}{\sigma_1} \right)^2 \right]^j,$$

nous définissons la méthode mixte comme une méthode basée sur le filtre

$$R_{\alpha,r,j} y := \sum_{m: \sigma_m > 0} F_{\alpha,r,j}(\sigma_m) \sigma_m^{-1} \langle y, u_m \rangle v_m. \quad (2.19)$$

On note

$$\begin{aligned} x_{\alpha,r,j} & : = R_{\alpha,r,j} y, \quad y \in \text{Dom}(A^+), \\ x_{\alpha,r,j}^\delta & : = R_{\alpha,r,j} y^\delta, \quad y^\delta \in Y. \end{aligned}$$

Remarquons que pour $r = 1$ et $j = 0$ on retrouve la méthode standard de Tykhonov.

Proposition 2.3.1 [8] *(L'ordre optimal de la régularisation de Tikhonov pondérée- I)*

Soit A un opérateur linéaire compact dont le domaine est infini. Pour tout $r \geq 0$ donné, la méthode de Tikhonov pondérée- I , $R_{\alpha,r}$ est une méthode de régularisation d'ordre optimal sous l'hypothèse à priori $x^+ \in X_{v,\rho}$ avec $0 < v \leq r + 1$. Le meilleur taux possible de convergence par rapport à δ est $\|x^+ - x_{\alpha,r}^\delta\| = O(\delta^{\frac{r+1}{r+2}})$, qui est obtenu pour $\alpha = \left(\frac{\delta}{\rho}\right)^{\frac{r+1}{r+2}}$. Par contre, si $\|x^+ - x_{\alpha,r}\| = O(\alpha)$ alors $x^+ \in X_{r+1}$.

Démonstration. La démonstration de cette proposition se trouve détaillée dans ([8]). \square

Proposition 2.3.2 [8] (*L'ordre optimal de la régularisation de Tikhonov pondérée-II*)

Soit A un opérateur linéaire compact dont le domaine est infini. Pour tout $j \geq 0$ donné, la méthode de Tikhonov pondérée-II, $R_{\alpha,j}$ est une méthode de régularisation d'ordre optimal sous l'hypothèse à priori $x^+ \in X_{v,\rho}$ avec $0 < v \leq 2$. Le meilleur taux possible de convergence par rapport à δ est $\|x^+ - x_{\alpha,j}^\delta\| = O(\delta^{\frac{2}{3}})$, qui est obtenu pour $\alpha = \left(\frac{\delta}{\rho}\right)^{\frac{2}{3}}$.

Démonstration. La démonstration de cette proposition se trouve détaillée dans ([8]). \square

Proposition 2.3.3 [8] (*L'ordre optimal de la méthode mixte*)

Soit A un opérateur linéaire compact dont le domaine est infini. Pour tout $r \geq 0$ et $j \in \mathbb{N}$ donné, la méthode **mixte**, $R_{\alpha,r,j}$ est une méthode de régularisation d'ordre optimal sous l'hypothèse à priori $x^+ \in X_{v,\rho}$ avec $0 < v \leq r + 1$. Le meilleur taux possible de convergence par rapport à δ est $\|x^+ - x_{\alpha,r,j}^\delta\| = O(\delta^{\frac{2}{3}})$, qui est obtenu pour $\alpha = \left(\frac{\delta}{\rho}\right)^{\frac{r+1}{r+2}}$. Par contre, si $\|x^+ - x_{\alpha,r,j}\| = O(\alpha)$ alors $x^+ \in X_{r+1}$.

Démonstration. La démonstration de cette proposition se trouve détaillée dans [8]. \square

2.4 La Méthode de Tykhonov en Imagerie

Le traitement d'images numériques est l'une des principales parties de l'apprentissage automatique ou de la vision par ordinateur. Pour un résultat précis, les images doivent être traitées avant analyse en plus d'être étudiées spécifiquement en machine learning. Pour une image initiale, on suppose qu'elle a été atténuée par un bruit additif v , et éventuellement par un opérateur R . Son inverse est souvent numériquement difficile. A partir de l'image observée $f = Ru + v$ (donc une version réduite de l'image originale u), on essaie de reconstruire u . Si nous supposons que le produit additive v est gaussien, alors la méthode du maximum de vraisemblance nous amène à trouver u comme solution au problème de minimisation :

$$\alpha := \inf_u \|f - Ru\|_2^2,$$

où $\|\cdot\|_2$ désigne la norme dans L^2 . Il s'agit d'un problème inverse mal posé. Pour le résoudre numériquement, nous introduisons un terme de régularisation et considérons le problème

$$\alpha_\lambda := \inf_u (\|f - Ru\|_2^2 + \lambda \|\nabla u\|^2). \quad (2.20)$$

Dans la suite, nous ne considérerons que le cas où R est l'opérateur d'identité ($Ru = u$) et nous avons utilisé le plus ancien modèle de récupération d'image que Tikhonov. Soient $V = H_0^1(\Omega)$ et $H = L^2(\Omega)$. On considère le problème de réduction initial (s'adapte aux données) :

$$(\mathcal{P}) \quad \min_{u \in V} \{J(u) = \|u - f\|_H^2\}.$$

Soient $\Omega \subset \mathbb{R}^N$ un sous-ensemble ouvert, borné et l'image observée (donnée) $f : \Omega \rightarrow \mathbb{R}$ telle que $f = u + v$ où u est l'image originale restaurée (niveau de gris au point $x = (x_1, x_2)$), v bruit gaussien de moyenne nulle et de variance σ . Le problème de régularisation minimise alors l'énergie de formation :

$$(\mathcal{P}_\alpha) \quad \min_{u \in V} \{ J_\alpha(u) = \|u - f\|_H^2 + \alpha \|\nabla u\|_H^2 \} ; \forall \alpha > 0,$$

c'est-à-dire

$$(\mathcal{P}_\alpha) \quad \min_{u \in V} \left\{ J_\alpha(u) = \int_{\Omega} [(u - f)(x)]^2 dx + \alpha \int_{\Omega} |\nabla u|^2 dx \right\} ; \forall \alpha > 0.$$

On ne veut ajuster u que pour la donnée f , mais on impose aussi que le gradient soit "suffisamment petit" (cela dépend du paramètre α). L'image avec de petites dégradations est "lisse". Les bords sont érodés et la restauration se traduira par une image floue.

Remarque 2.4.1 La fonctionnelle $J(u) = \|u - f\|_H^2$ n'est pas coercive sur V :
En effet ; pour $\Omega =]0, 1[$, $u_n(x) = x^n$, $u_d = 0$.

On voit que

$$\|u_n\|_2 = \frac{1}{\sqrt{2n}}, \quad \|u'_n\|_2 = \frac{n}{\sqrt{2n-1}}.$$

On a

$$\lim_{n \rightarrow +\infty} \|u_n\|_V = +\infty \text{ et } \lim_{n \rightarrow +\infty} J(u_n) = 0.$$

Il n'est pas clair que le problème (\mathcal{P}) ait une solution. Donc la fonctionnelle $J(u)$ n'est pas coercive, c'est-à-dire, la solution n'existe pas alors le problème original de minimisation est mal posé.

Proposition 2.4.1 ([6]) Supposon que (\mathcal{P}) admet au moins une solution \check{u} . Le problème (\mathcal{P}_α) admet une solution unique u_α . De la famille (u_α) on peut extraire une sous-suite qui converge (faiblement) dans V vers une solution u^* de (\mathcal{P}) lorsque $\alpha \rightarrow 0$.

Démonstration. La solution existe, unique et converge donc le problème régularisé de Tykhonov est bien posé. La démonstration de cette proposition se trouve détaillée dans ([6]). \square

Remarque 2.4.2 (Calcul de u_α solution du problème régularisé de Tykhonov) Si J_α est différentiable au sens de Gâteaux et si le problème (\mathcal{P}_α) a une solution u_α , alors une condition nécessaire et suffisante est que la solution vérifie

$$J'_\alpha(u_\alpha) = 0.$$

La réciproque est vraie si J_α est convexe. Un simple calcul montre que

$$\begin{aligned} \forall u \in V \quad J'_\alpha(u_\alpha) \cdot u &= \int_{\Omega} (u_\alpha - f)(x)u(x)dx + \int_{\Omega} \nabla u_\alpha(x) \nabla u(x)dx \\ &= \int_{\Omega} (u_\alpha - f - \Delta u_\alpha)(x)u(x)dx. \end{aligned}$$

Par conséquent l'équation d'Euler qui donne la solution u_α est la suivante :

$$u_\alpha - f - \Delta u_\alpha = 0, \quad u_\alpha \in H_0^1(\Omega).$$

Comme pour la méthode des contours actifs, on peut se contenter d'approcher la solution u_α en écrivant la formulation dynamique :

$$\frac{\partial u}{\partial t} - \Delta u + u = f.$$

2.4.1 Résultats Numériques pour le Modèle de Tykhonov

Nous avons utilisé trois exemples d'images, introduisant le bruit gaussien additif, et tenté de récupérer l'image débruitée à partir de l'image bruité pour tester la méthode de Régularisation de Tykhonov. Soit I les matrices qui représentent une image de taille $m \times n$. Nous avons ensuite utilisé la commande `noisyI = imnoise(I, 'gaussian', sigma)`; de Matlab pour définir notre image de bruit $noisyI$, où $sigma$ est une variante du niveau de bruit gaussien. On utilise la valeur $\lambda = 0.2$ pour l'étude.

Nous évaluons la qualité des images restaurées par le modèle de Tykhonov, nous utilisons l'erreur carrée (MSE), le taux de bruit du signal (SNR), le rapport signal/bruit de crête ($PSNR$), l'indice de qualité d'image (IQI), corrélation croisée normalisée (NK), différence moyenne (AD), contenu structurel (SC), différence maximale (MD) et erreur absolue normalisée (NAE).

Images	Taille $n \times m$	$\frac{\ Débruité I - Bruitée I\ ^2}{(MN * sigma^2) - 1}$	Débruité PSNR	Bruité PSNR
cameraman	398 × 398	-0.97352	23.9646	20.3181
barbara	510 × 510	-0.96845	23.6274	20.1075
camera	340 × 340	-0.97294	24.0817	20.4143
flower	256 × 256	-0.97523	24.1745	20.1339
girl	216 × 233	-0.97692	24.3646	20.2102
Iline	1961 × 3553	-0.97614	24.2821	19.9895
university	480 × 640	-0.97551	24.3021	20.554

Tableau 2.1 – Résultats obtenus par l'algorithme de régularisation de Tykhonov, $sigma = 0.01$.

2.5 Conclusions

La méthode de régularisation de Tikhonov est la plus ancienne méthode régulière utilisée pour résoudre des problèmes inverses mal posé en imagerie. Ce modèle vise à produire mieux la solution pour supprimer le bruit de l'image. On remarque que si on a utilisé le terme de régularisation classique $L(u) = \|\nabla u\|_2^2$, cette méthode n'est pas bien adapté au problème de restauration d'image car l'image restaurée u est alors beaucoup trop lisse (surtout, les bords sont érodés). Alors une approche plus efficace consiste à regarder la variation totale,



Figure 2.1 – Les images de textes ” cameraman, girl, university.”

Images	Taille $n \times m$	$\frac{\ Débruité I - Bruitée I\ ^2}{(MN * \sigma^2) - 1}$	Débruité PSNR	Bruitée PSNR
cameraman	398×398	-0.97254	19.8452	17.9764
barbara	510×510	-0.96842	19.7739	17.9621
camera	340×340	-0.97612	20.7023	18.8938
flower	256×256	-0.97468	19.9527	17.9206
girl	216×233	-0.9771	20.2027	18.1269
Iline	1961×3553	-0.97636	20.0426	17.9203
university	480×640	-0.97617	20.5036	18.5907

Tableau 2.2 – Résultats obtenus par l’algorithme de régularisation de Tychonov, $\sigma = 0.08$.

c’est-à-dire en prenant $L(u) = \int |Du|$ pour trouver des image plus claire n’est pas lisse. Cela conduit à changer l’espace de travail à un espace de Banach particulier pour la minimisation des fonctions, mais qui est bien adapté au problème, c’est-à-dire, des espaces de fonctions à

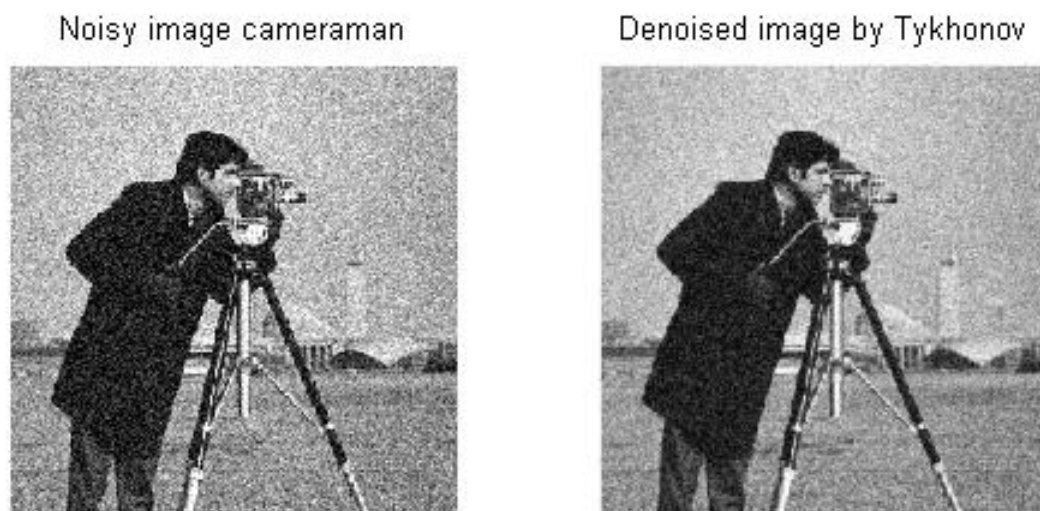


Figure 2.2 – Image bruitée de cameraman et image débruité par Tykhonov, sigma 0.08.

Images	Taille $n \times m$	$\frac{\ Débruité I - Bruitée I\ ^2}{(MN * \sigma^2) - 1}$	Débruité PSNR	Bruitée PSNR
cameraman	398×398	-0.97377	13.7374	13.2205
barbara	510×510	-0.97051	13.7499	13.2581
camera	340×340	-0.98386	15.4174	15.0525
flower	256×256	-0.9748	13.6479	13.078
girl	216×233	-0.97887	14.0148	13.4558
lline	1961×3553	-0.97781	13.769	13.2012
university	480×640	-0.97776	14.4049	13.8881

Tableau 2.3 – Résultats obtenus par l'algorithme de régularisation de Tychonov, sigma = 0.2.

variations bornées BV .



Figure 2.3 – Image bruitée de girl et image débruité par Tykhonov, sigma 0.08.

Images	MSE	SNR	PSNR	IQI	NK	AD	SC	MD	NAE
cameraman	95.4334	3.9647	28.3338	2.1712e-05	1.3361	-64.2595	0.4506	244	0.6637
barbara	96.1126	4.2188	28.3030	6.7731e-09	1.2060	-51.0406	0.4732	223	0.7462
camera	94.6071	4.3055	28.3716	3.0292e-05	1.1294	-37.6595	0.7005	241	0.3220
flower	95.6821	4.2095	28.3225	-3.6032e-06	1.5423	-88.3622	0.2636	201	1.1744
girl	95.0412	28.3517	4.1605	2.0528e-07	1.3220	-75.1643	0.4206	201	0.7728
l1ne	95.9190	4.2459	28.3118	0.9841	1.3469	-57.9367	0.4791	254	0.5287
university	88.4816	4.5937	28.6623	7.1609e-04	1.0743	-46.4538	0.5522	254	0.7034

Tableau 2.4 – Mesures de performance par l’algorithme Tykhonov, sigma=0.01.

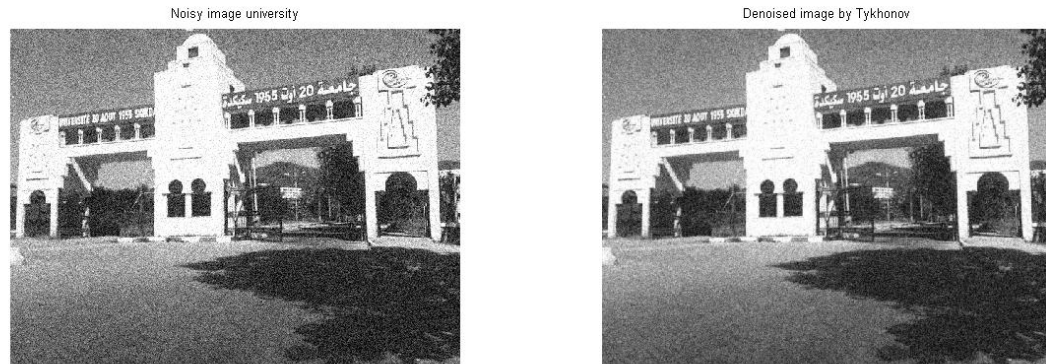


Figure 2.4 – Image bruitée de University et image débruité par Tykhonov, sigma 0.08.

Images	MSE	SNR	PSNR	IQI	NK	AD	SC	MD	NAE
cameraman	164.1012	1.6105	25.9797	-4.2489e-06	1.3902	-73.4494	0.4244	233	0.6934
barbara	165.0845	1.8696	25.9537	4.0697e-08	1.2888	-62.3501	0.4363	223	0.7574
camera	162.7122	1.9505	26.0166	3.0208e-05	1.1702	-46.9840	0.6554	235	0.3418
flower	1.8539	164.5883	25.9668	-3.0678e-06	1.6220	-96.3537	0.2529	197	1.1856
girl	163.1718	1.8132	26.0044	2.8824e-07	1.3688	-82.2831	0.4034	202	0.7837
liline	164.7801	1.8959	25.9618	0.9893	1.4084	-66.8355	0.4511	238	0.5417
university	151.8053	2.2494	26.3179	7.7624e-04	1.1404	-57.1934	0.5148	254	0.7062

Tableau 2.5 – Mesures de performance par l’algorithme Tykhonov, sigma=0.08.

Images	MSE	SNR	PSNR	IQI	NK	AD	SC	MD	NAE
cameraman	238.3766	-0.0110	24.3582	1.1434e-06	1.4810	-89.3073	0.3809	221	0.7922
barbara	239.9024	0.2463	24.3305	-4.0468e-08	1.4267	-81.3551	0.3786	223	0.8199
camera	236.3087	0.3299	24.3960	3.0276e-05	1.2233	-59.7684	0.6005	234	0.3932
flower	239.7857	0.2196	24.3326	-2.8803e-06	1.7591	-110.1366	0.2331	182	1.2489
girl	237.7749	0.1780	24.3691	-1.5265e-07	1.4463	-94.0664	0.3748	195	0.8222
liline	239.5990	0.2701	24.3360	0.9882	1.5113	-81.7794	0.4052	208	0.6085
university	219.8879	0.6402	24.7088	7.7368e-04	1.2425	-74.6313	0.4586	254	0.7748

Tableau 2.6 – Mesures de performance par l’algorithme Tykhonov, sigma=0.2.

Chapitre 3

Modèle de Rudin-Osher-Fatemi

3.1 Introduction Historique

La restauration ou la reconstruction d'une image réelle u à partir d'une observation f est une tâche essentielle dans le traitement d'image. Le problème est d'extraire u de f , étant donné qu'une image de fonction f définie sur Ω , avec $\Omega \subset \mathbb{R}^2$ un domaine ouvert et bornée. L'observation f est généralement une version d'image bruyante ou floue de l'image réelle. Il faut trouver une alternative à la régularisation trop violente de Tykhonov. La première idée est de remplacer le terme "régularisation $\|\nabla u\|_H^2$ quadratique" qui est en fait une pénalisation par une norme variationnelle de V moins contraignant.

Rudin-Osher-Fatemi ([47]) ont proposé le premier modèle de restauration d'image basé sur une image bruyante avec un bruit additif utilisant la régularisation (TV) définie par

$$TV(u(x, y)) = \int_{\Omega} |\nabla u(x, u)| \, dx dy \quad \text{avec} \quad |\nabla u| = \sqrt{u_x^2 + u_y^2}.$$

La méthode de régularisation de la variation totale (TV) du traitement d'image est utilisée pour éliminer le bruit des images numériques. C'est la seule régularisation qui est utilisée pour préserver les bords de l'image et éliminer les composants souvent bruyants de l'image. (TV) est une technique développée par Rudin-Osher-Fatemi pour le problème de débruitage d'image, depuis lors, il a été appliqué à une variété d'autres problèmes d'imagerie.

$$J(u) = \int_{\Omega} |D(u)| \, dx = \sup \left\{ \int_{\Omega} u(x) \operatorname{div}(\varphi(x)) \, dx : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_{\infty} \leq 1 \right\}.$$

L'expression de régularisation $\int |D(u)|$ est parfois appelée "régularisation à variation totale TV ". Pour ce qui est de la minimisation, TV élimine la possibilité d'avoir des oscillations, mais elle offre la possibilité d'avoir des discontinuités. Cette capacité d'inclure des "sauts" dans la solution l'a rendu très populaire pour la destruction d'image. Malheureusement, le problème de ROF est difficile à résoudre parce que le terme TV est non linéaire et non différentiable. Néanmoins, en raison de la propriété souhaitable de débruitage, ce problème a reçu beaucoup d'attention, et beaucoup d'algorithmes de solution ont été proposés. Le problème de débruitage d'image est un problème inverse mal posé de la suppression du bruit d'une image. Il couvre un large éventail de sujets et est actif dans le domaine de

recherche du traitement d'image. Il existe différents types de bruit, le cas le plus étudié avec un bruit additif gaussien : $f = u + \eta$, où f est l'image bruyante observée, u est l'image originale et η la variation aléatoire gaussienne à moyenne nulle.

3.2 Problème de ROF

Rudin-Osher-Fatemi ont proposé un modèle ([20]) dans lequel l'image est divisée en deux parties : $f = u + v$ où v est le bruit et u est la partie "régulière", u est une image inconnue, f est une mesure bruyante typique au début d'une image propre et $\lambda > 0$ est un paramètre. Ainsi, en utilisant la formule $u+v$ avec $u \in BV(\Omega)$ et $v \in L^2(\Omega)$, nous chercherons une solution au problème et nous n'appliquerons la régularisation qu'à la partie "bruit". Si $f \in L^2$ est vrai, alors, le minimiseur u existe, unique et stable dans L^2 donc le problème de ROF est bien posé. Le problème de Rudin-Osher-Fatemi est donné par

$$(\mathcal{P}_{ROF}) \quad \alpha_{ROF} := \inf_u \left\{ J(u) + \frac{1}{2\lambda} \|v\|_2^2 \mid u \in BV(\Omega), v \in L^2(\Omega), f = u + v \right\},$$

ce qui conduit à une décomposition $[BV(\Omega), L^2(\Omega)]$ de l'image f .

Ici $J(u)$ désigne la variation totale de u et $\lambda > 0$, où

$$J(u) = \sup \left\{ \int_{\Omega} u(x) \operatorname{div}(\varphi(x)) dx : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_{\infty} \leq 1 \right\},$$

et BV l'espace des fonctions à variations bornées, c'est l'espace de toutes les images à TV qui est défini par

$$BV(\Omega) := \{u \in L^1(\Omega), J(u) < +\infty\}.$$

En dimension 1, les fonctions C^1 , les fonctions k -Lipschitziennes et les fonctions $W^{1,1}$ sont à variations bornées. Par contre, toutes les fonctions continues ne sont pas à variations bornées. L'espace des fonctions de variations bornées est bien adapté au traitement d'image parce que l'intérêt de cet espace pour le traitement d'image repose sur le fait qu'il est bien adapté à la minimisation de la fonctionnelle.

Proposition 3.2.1 [6], L'application

$$\begin{aligned} BV(\Omega) &\longrightarrow \mathbb{R}^+ \\ u &\longmapsto \|u\|_{BV(\Omega)} = \|u\|_{L^1} + J(u), \end{aligned}$$

est une norme.

Théorème 3.2.1 [6], L'espace $BV(\Omega)$ muni de la norme

$$\|u\|_{BV(\Omega)} = \|u\|_{L^1} + J(u),$$

est un espace de Banach.

Remarque 3.2.1 On a

$$W^{1,1}(\Omega) \subset BV(\Omega).$$

En particulier, comme Ω est borné

$$\forall 1 \leq P \leq +\infty \quad W^{1,p}(\Omega) \subset BV(\Omega).$$

Toute fonction d'un espace de Sobolev est à variation bornée.

Du est une mesure finie qui est définie par

$$Du = \nabla u dx + (u^+ - u^-)n_u dh_{S_u}^{n-1} + C_u.$$

Selon la théorie de décomposition de Lebesgue, elle est divisée en deux parties : une qui est toujours continue par rapport à la mesure de Lebesgue, et une qui est singulière par rapport à la mesure de Lebesgue. Le composant doit être continu avec une dérivée de Radon-Nykodym, notée ∇u qui est dans L^1 . Ceci est vrai pour toutes les mesures (raisonnables). Maintenant, dans le cas d'un gradient de fonction BV , le composant unique se décompose dans le terme de saut $(u^+ - u^-)n_u d\mathcal{H}_{S_u}^1$ et la partie de Cantor C_u . Mais C_u est la différence entre Du et $\nabla u dx + (u^+ - u^-)n_u d\mathcal{H}_{S_u}^1$.

Décomposition dans BV
[Giusti 84, Evans & Gariepy 92, Ambrosio 98]

$$Du(\Omega) = \underbrace{\nabla u \cdot dx}_{\text{Partie absolument continue}} + \underbrace{(u^+ - u^-)n_u d\mathcal{H}_{S_u}^1}_{\text{Partie singulière (sauts)}} + \underbrace{C(u)}_{\text{Partie Cantor}}$$

Figure 3.1 – Décomposition dans BV.

Exemple 3.2.1 (Exemple de fonction BV) : Soit

$$u(x_1, x_2) = \begin{cases} 0 & \text{si } x_1 < 0, \\ 1 & \text{si } x_1 > 0. \end{cases}$$

Avec cette fonction, on a

$$\nabla u = 0, \quad u_-(x) = \begin{cases} 0 & \text{si } x_1 \leq 0, \\ 1 & \text{si } x_1 > 0, \end{cases} \quad \text{et } u_+(x) = \begin{cases} 0 & \text{si } x_1 < 0, \\ 1 & \text{si } x_1 \geq 0, \end{cases}$$

où

$$S_u = \{x_1 = 0\}, \quad n_u = (1, 0) \text{ et } Du = H^1_{\{x_1=0\}}.$$

Ici u_- est la limite à gauche de la discontinuité de u et u_+ la limite à droite de la discontinuité de u .

Théorème 3.2.2 ([6]) *Le problème $(\mathcal{P}_{\mathcal{ROF}})$ admet une solution unique.*

Démonstration. En effet, d'après ([6]), soit le problème de ROF suivant :

$$(\mathcal{P}_{\mathcal{ROF}}) \quad \inf \left\{ J(u) + \frac{1}{2\lambda} \|v\|_2^2 \mid u \in BV(\Omega), v \in L^2(\Omega), f = u + v \right\}.$$

1. Existence de la solution :

En effet, soit $u_n \in BV(\Omega)$, c'est-à-dire, $u_n \in L^1(\Omega)$ et $J(u_n) < +\infty$, $v_n \in L^2(\Omega)$ une suite minimisante. On a v_n est bornée dans $L^2(\Omega)$ donc peut extraire une sous-suite (notée v_n) faiblement convergente vers v^* dans $L^2(\Omega)$, comme la norme $L^2(\Omega)$ est convexe et semi-continue inférieurement, alors

$$\|v^*\|_2^2 \leq \liminf_{n \rightarrow +\infty} \|v_n\|_2^2.$$

Puisque Ω est borné, $u_n = f - v_n$ est aussi bornée dans $L^2(\Omega)$ et dans $L^1(\Omega)$. Comme $J(u_n)$ est borné, donc u_n est bornée dans $BV(\Omega)$. Cela a l'effet que u_n converge fortement dans $L^1(\Omega)$ vers $u^* \in BV(\Omega)$.

D'autre part, J est semi-continue inférieurement, alors $J(u^*) \leq \liminf_{n \rightarrow +} J(u_n)$. Puis

$$J(u^*) + \frac{1}{2\lambda} \|v^*\|_2^2 \leq \liminf_{n \rightarrow +} J(u_n) + \frac{1}{2\lambda} \|v_n\|_2^2 = \inf(\mathcal{P}_{\mathcal{ROF}}).$$

On a, pour tout n , $f = u_n + v_n$ et $f = u^* + v^*$. En conséquence u^* est une solution du problème $(\mathcal{P}_{\mathcal{ROF}})$. Le problème $(\mathcal{P}_{\mathcal{ROF}})$ peut s'écrire de manière équivalente suivante :

$$\inf_{u \in BV(\Omega)} F(u) := J(u) + \frac{1}{2\lambda} \|u - f\|_2^2.$$

Pour que la fonctionnelle F soit convexe, une condition nécessaire et suffisante d'optimalité du premier ordre est que

$$u^* \text{ solution de } (\mathcal{P}_{\mathcal{ROF}}) \iff 0 \in \partial F(u^*) \iff u^* \in \partial F^*(0).$$

2. L'unicité de la solution : la fonctionnelle J est strictement convexe par rapport (u, v) et la contrainte est affine donc la solution u^* est unique.

□

Théorème 3.2.3 ([6], [20]) *La solution du problème ROF est donnée par*

$$u = f - \lambda \Pi_{\lambda K}(f),$$

où $\Pi_{\lambda K}$ est le projecteur orthogonal sur λK , avec

$$K := \{ \operatorname{div}(\varphi) / \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_\infty \leq 1 \}.$$

Démonstration. En effet, d'après ([6], [20]), soit le problème de ROF suivant :

$$(\mathcal{P}_{\mathcal{ROF}}) \quad \min_u \left\{ J(u) + \frac{1}{2\lambda} \|u - f\|_2^2 / u \in BV(\Omega), v \in L^2(\Omega), f = u + v \right\}.$$

On obtient d'après l'équation d'Euler Lagrange

$$0 \in u - f + \lambda \partial J(u) \Leftrightarrow u = f + \frac{1}{2\lambda} \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right);$$

∂J est le sous-différentiel de J défini par

$$w \in \partial J(u) \iff \partial J(u) := \{ J(v) \geq J(u) + \langle w, v - u \rangle_X, \forall v \in X \}.$$

Si u est un minimiseur de $(\mathcal{P}_{\mathcal{ROF}})$, alors d'après la condition nécessaire et suffisante d'optimalité du premier ordre, on a

$$0 \in \partial \left(J(u) + \frac{1}{2\lambda} \|u - f\|_X^2 \right) = \frac{u - f}{\lambda} + \partial J(u).$$

Puisque J est une fonction convexe, semi-continue inférieurement et propre c'est-à-dire,

J est une fonction convexe

⇕

$$\forall x, y \in X, \lambda \in [0, 1] : J(\lambda x + (1 - \lambda)y) \leq \lambda J(x) + (1 - \lambda)J(y);$$

J est une fonction semi-continue inférieurement

⇕

$$\forall x_0 \in X, \liminf_{x \rightarrow x_0} J(x) \geq J(x_0);$$

J est une fonction propre sur $X \iff J(x) > -\infty \forall x \in X$ et $J \neq +\infty$.

Alors

$$\frac{f - u}{\lambda} \in \partial J(u) \iff u \in \partial^* J\left(\frac{f - u}{\lambda}\right) \iff 0 \in -u + \partial^* J\left(\frac{f - u}{\lambda}\right).$$

Ceci est aussi équivalent à

$$0 \in \frac{f - u}{\lambda} - \frac{f}{\lambda} + \frac{1}{\lambda} \partial^* J\left(\frac{f - u}{\lambda}\right).$$

On en déduit que $w = \frac{f-u}{\lambda}$ est un minimiseur de

$$\frac{\|w - \frac{f}{\lambda}\|^2}{2} + \frac{1}{\lambda} J^*(w).$$

Comme

$$J^*(v) = \chi_K(v) = \begin{cases} 0 & \text{si } v \in K \\ +\infty & \text{si non;} \end{cases}$$

J^* est l'indicatrice de K , cela implique que $\frac{f-u}{\lambda}$ est la projection orthogonale de $\frac{f}{\lambda}$ sur le convexe K . Donc la solution du ROF est donnée par

$$u = f - \lambda \Pi_{\lambda K}(f).$$

□

3.2.1 Discrétisation

Selon ([20]); en variables discrètes, la variation totale (TV) est l'ensemble des accroissements de f .

On note X l'espace Euclidien $\mathbb{R}^{N \times N}$, si $u \in X$ alors $\nabla u \in Y = X \times X$ et

$$\int_{\Omega} |D(u)| dx = \int_{\Omega} |\nabla u| dx \approx \sum_{i,j} |\nabla u|_{i,j} \quad \text{avec} \quad |\nabla u|_{i,j} = \sqrt{(u_{i+1,j} - u_{i,j})^2 + (u_{j+1,i} - u_{i,j})^2}.$$

On choisit $(\nabla u)_{i,j} = ((\nabla u)_{i,j}^1, (\nabla u)_{i,j}^2)$ avec

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{si } i < N, \\ 0 & \text{si } i = N, \end{cases}$$

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{si } j < N, \\ 0 & \text{si } j = N. \end{cases}$$

On introduit la divergence discrète, par analogie à la divergence continue, par

$$\text{div} = -\nabla^*.$$

$$\forall p \in \mathbb{R}^{N^2} \times \mathbb{R}^{N^2}, u \in \mathbb{R}^{N^2}, \langle -\text{div}(p), u \rangle_{\mathbb{R}^{N^2}} = \langle p, \nabla u \rangle_{\mathbb{R}^{N^2} \times \mathbb{R}^{N^2}}$$

On a

$$\text{div}(p)_{i,j} = \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{si } 1 < i < N \\ p_{i,j}^1 & \text{si } i = 1 \\ -p_{i-1,j}^1 & \text{si } i = N \end{cases} + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{si } 1 < i < N \\ p_{i,j}^2 & \text{si } j = 1 \\ p_{i,j-1}^2 & \text{si } j = N. \end{cases}$$

On utilise l'algorithme suivant de Chambolle pour calculer $\Pi_{\lambda K}(f)$ en dimension 2 :

Algorithme 3.2.1 ([20]) Donner un algorithme pour calculer $\Pi_{\lambda K}(f)$ équivalent à résoudre le problème suivant :

$$\Pi_{\lambda K}(f) = \arg \min \{ \|\lambda \operatorname{div}(p) - f\|_X^2, \quad |p_{i,j}| \leq 1, i, j = 1 \dots N \}.$$

La condition de Karush-Kuhn-Tucker donne l'existence des multiplicateurs lagrangiens $\alpha_{i,j} \geq 0$ associés à chaque problème avec contrainte, tel que l'on ait :

$$\forall i, j \quad -(\nabla(\lambda \operatorname{div}(p) - f))_{i,j} + \alpha_{i,j} p_{i,j} = 0.$$

Cas 1 : $\alpha_{i,j} > 0$ et $|p_{i,j}| = 1 \Rightarrow \alpha_{i,j} = |\nabla(\lambda \operatorname{div}(p) - f)|_{i,j}$.

Cas 2 : $\alpha_{i,j} = 0$ et $|p_{i,j}| < 1 \Rightarrow -\nabla(\lambda \operatorname{div}(p) - f)_{i,j} = 0$.

On propose un algorithme de gradient de decente semi-implicite où algorithme du point fixe :

Algorithme 3.2.2 On choisit $\tau > 0$, $p^0 = 0$ et pour tout $n \geq 0$:

$$p_{i,j}^{n+1} = p_{i,j}^n - \tau(-(\nabla(\lambda \operatorname{div}(p^n) - \frac{f}{\lambda}))_{i,j} + \alpha_{i,j} p_{i,j}^{n+1});$$

$$p_{i,j}^{n+1} = p_{i,j}^n + \tau((\nabla(\lambda \operatorname{div}(p^n) - \frac{f}{\lambda}))_{i,j} - |\nabla(\lambda \operatorname{div}(p^n) - f)|_{i,j} p_{i,j}^{n+1});$$

$$p_{i,j}^{n+1}(1 + \tau |\nabla(\lambda \operatorname{div}(p^n) - f)|_{i,j}) = p_{i,j}^n + \tau(\nabla(\lambda \operatorname{div}(p^n) - \frac{f}{\lambda}))_{i,j};$$

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau (\nabla(\operatorname{div}(p^n) - \frac{f}{\lambda}))_{i,j}}{1 + \tau |(\nabla(\operatorname{div}(p^n) - \frac{f}{\lambda}))|_{i,j}}. \quad (3.1)$$

La convergence de l'Algorithme est donnée par le Théorème suivant :

Théorème 3.2.4 ([20]) Si le paramètre τ dans l'expression (3.1), vérifie $\tau \leq 1/8$, alors $\lambda \operatorname{div} p^n$ converge vers $\Pi_{\lambda K}(f)$ quand $n \rightarrow +\infty$, c'est-à-dire,

$$\lim_{n \rightarrow +\infty} \lambda \operatorname{div}(p^n) - \Pi_{\lambda K}(f) = 0.$$

Alors, la solution du problème est donnée par

$$u = f - \lambda \operatorname{div}(p^\infty), \quad \text{où } p^\infty = \lim_{n \rightarrow \infty} p^n.$$

Démonstration. La démonstration de ce théorème se trouve détailler dans ([20]). \square
Théoriquement le taux de convergence $\tau \leq \frac{1}{8}$ mais en pratique en prend $\tau \leq \frac{1}{4}$.

3.2.2 Image Débruitée

Chambolle ([20]) et Rudin-Osher-Fatemi ([48]) proposaient l'idée de réduire la variation totale pour le débruitage d'image, ce qui implique que la variation observée est faible. Supposons que l'image observée $f = (f_{i,j})_{1 \leq i,j \leq N}$ est l'addition d'une image a priori lisse par morceaux (ou avec peu d'oscillation) $u = (u_{i,j})_{1 \leq i,j \leq N}$ et un bruit gaussien aléatoire, de variance σ^2 . En conséquence, il est recommandé qu'on essaie de récupérer l'image originale u en résolvant le problème

$$\min_u \{J(u) := \|u - f\|^2 = \|v\|^2 = \|\operatorname{div} p\|^2 = N^2 \sigma^2\}.$$

Le nombre total de pixels est N^2 . Il existe un multiplicateur de Lagrange $\lambda > 0$ tel que, si $\|f - \langle f \rangle\|^2 \geq N^2 \sigma^2$ (avec $\langle f \rangle$ la valeur moyenne des pixels $f_{i,j}$), ce problème a une solution unique qui est donnée par le problème suivant :

$$(\mathcal{P}_{\mathcal{R} \circ \mathcal{F}}) \quad \min_{u \in X} \left\{ J(u) + \frac{1}{2\lambda} \|u - f\|_2^2 \right\}.$$

Pour trouver la convergence rapide de limite u qui résoud ce problème, on remplace λ par $\frac{N\sigma}{\|\operatorname{div} p^n\|}$ après chaque itération de

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau \left(\nabla(\operatorname{div}(p^n) - \frac{f}{\lambda}) \right)_{i,j}}{1 + \tau \left| \left(\nabla(\operatorname{div}(p^n) - \frac{f}{\lambda}) \right) \right|_{i,j}}.$$

3.3 Résultats Numériques

Nous avons utilisé trois exemples d'images. Nous trouvons les images débruitées par la méthode de *ROF* à partir de l'image bruitée par le bruit gaussien additif. Soit I les matrices qui représentent une image de taille $m \times n$. Nous avons ensuite utilisé la commande `noisyI = imnoise(I, 'gaussian', sigma)`; de Matlab pour définir notre image de bruit *noisyI*, où *sigma* est une variante du niveau de bruit gaussien. Nous utilisons les valeurs $\varepsilon = 0.0001$; $\lambda = 0.2$ et la tolérance $Tol = 10^{-3}$ dans nos études et le nombre d'itérations pour la boucle de 0 jusqu'à 100.

Nous évaluons la qualité des images restaurées par les modèles de restauration d'image, nous utilisons l'erreur carrée (*MSE*), le taux de bruit du signal (*SNR*), le rapport signal/bruit de crête (*PSNR*), l'indice de qualité d'image (*IQI*), corrélation croisée normalisée (*NK*), différence moyenne (*AD*), contenu structurel (*SC*), différence maximale (*MD*) et erreur absolue normalisée (*NAE*).

Il existe plusieurs méthodes numériques pour les solutions du modèle de *ROF* pour plus de détails voir par exemples :

1. **la méthode implicite (M1)** : qui utilise la méthode des différences finies avec le schéma implicite pour résoudre l'équation Euler Lagrange de *ROF* ;

2. la méthode de Lagrangien Augmenté (M2) :

$$\min_{u,p} \max_X L_{ROF}(u,p,X) = \int_{\Omega} |p| + \frac{\lambda}{2} \|u - f\|_2^2 + \int_{\Omega} X \cdot (p - \nabla u) + \frac{r}{2} \int_{\Omega} |p - \nabla u|^2,$$

avec $L_{ROF}(u,p,X)$ la fonction lagrangienne augmentée, X le multiplicateur de Lagrange et r constante positive.



Figure 3.2 – Les images de textes ” animal, fleur, camera.”

3.4 Conclusions

Le modèle de ROF améliore le modèle de Tychonov et donne de meilleure qualité d'image car il remplace le terme de régularisation $\|\nabla u\|_H^2$ quadratique par la norme V qui est une norme moins contraignante régularisante, et utilise l'espace BV pour assurer la continuité. BV est le bon espace de fonctions pour les images, car c'est un espace de fonctions qui peuvent

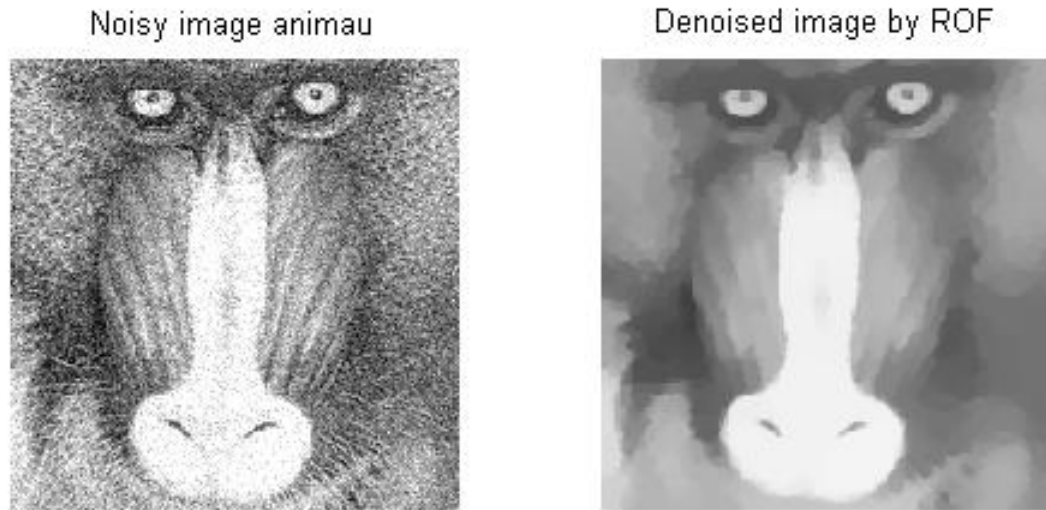


Figure 3.3 – Image bruitée de animal et image débruité par ROF, sigma 0.08.

Images	Taille $n \times m$	Nombre d'itérations	$\frac{\ Débruité I - Bruitée I\ ^2}{(MN * \sigma^2) - 1}$	Débruité PSNR	Bruitée PSNR
cameraman	398×398	25	0.1349	24.6132	20.3181
barbara	510×510	25	0.41724	22.3979	20.1075
camera	340×340	26	0.1179	24.9959	20.4143
flower	256×256	22	0.099939	25.9711	20.1339
girl	216×233	21	0.018344	26.1987	20.2102
lilne	1961×3553	24	34.5489	34.5489	19.9895
university	480×640	25	0.050125	25.0007	20.554

Tableau 3.1 – Résultats obtenus par l'algorithme de régularisation de ROF, sigma = 0.01.

être discontinues sur des courbes, donc qui peuvent représenter des images avec des contours. Cette méthode est excellente si l'image reconstruite est presque lisse par morceaux, Elle nécessite un grand nombre d'itérations, ce qui est peut être parfois lent. Le modèle de *ROF* permet d'obtenir un effet d'escalier en restauration des images lisses dans les applications

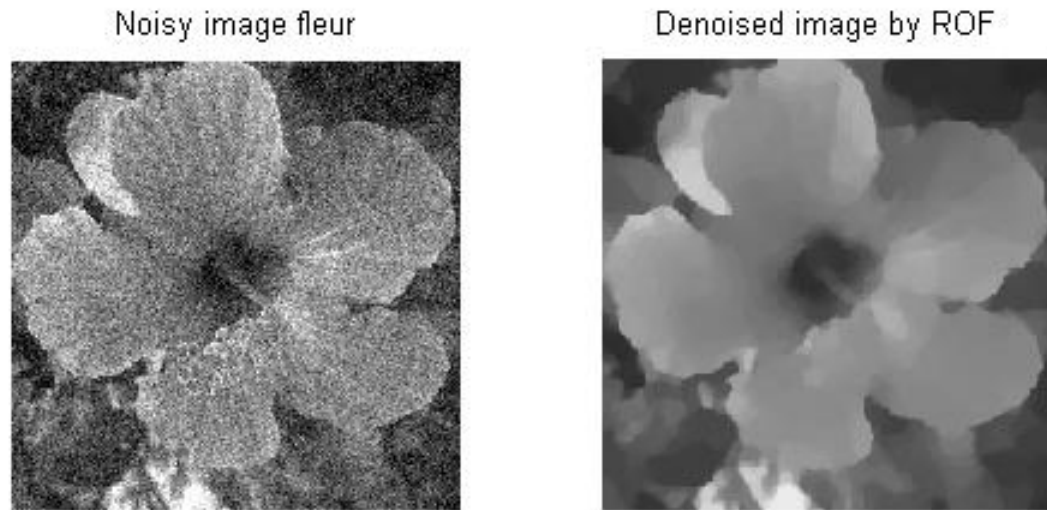


Figure 3.4 – Image bruitée de fleur et image débruité par ROF, sigma 0.08.

Images	Taille $n \times m$	Nombre d'itérations	$\frac{\ Débruité I - Bruitée I\ ^2}{(MN * \sigma^2) - 1}$	Débruité PSNR	Bruitée PSNR
cameraman	398×398	25	0.1349	24.6132	20.3181
barbara	510×510	25	0.41724	22.3979	20.1075
camera	340×340	29	-0.87675	20.7695	18.8938
flower	256×256	22	0.099939	25.9711	20.1339
girl	216×233	20	-0.87384	20.7782	18.1269
lilne	1961×3553	29	-0.87735	21.8224	17.9203
university	480×640	26	-0.8717	20.6516	18.5907

Tableau 3.2 – Résultats obtenus par l'algorithme de régularisation de ROF, sigma = 0.08.

où les bords ne sont pas les principales caractéristiques. Il génère, également, la perte de contraste d'image pendant le processus de restauration et il contient également la difficulté dans la norme de variation totale TV .

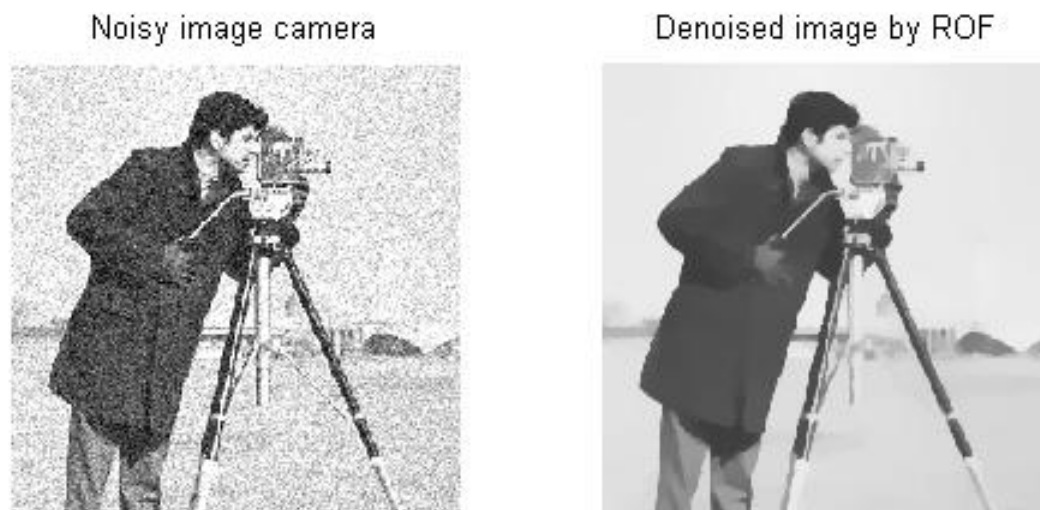


Figure 3.5 – Image bruitée de camera et image débruité par ROF, sigma 0.08.

Images	Taille $n \times m$	Nombre d'itérations	$\frac{\ Débruité I - Bruitée I\ ^2}{(MN * \sigma^2) - 1}$	Débruité PSNR	Bruitée PSNR
cameraman	398×398	38	-0.94385	13.7602	13.2205
barbara	510×510	30	-0.93372	13.5776	13.2581
camera	340×340	38	-0.9668	15.234	15.0525
flower	256×256	34	-0.94424	13.7758	13.078
girl	216×233	25	-0.95348	14.0803	13.4558
lline	1961×3553	32	-0.95397	14.0965	13.2012
university	480×640	35	-0.95166	14.3564	13.8881

Tableau 3.3 – Résultats obtenus par l'algorithme de régularisation de ROF, sigma = 0.2.

Images	MSE	SNR	PSNR	IQI	NK	AD	SC	MD	NAE
cameraman	95.4334	3.9647	28.3338	-2.4160e-06	1.3366	-64.3582	0.4536	223	0.6382
barbara	96.1126	4.2188	28.3030	4.8606e-08	1.2066	-51.1091	0.4790	223	0.7199
camera	94.6071	4.3055	28.3716	3.1438e-05	1.1302	-37.7750	0.7046	234	0.3002
flower	95.6821	4.2095	28.3225	-7.1457e-07	1.5430	-88.4721	0.2649	182	1.1511
girl	95.0412	4.1605	28.3517	-1.5265e-07	1.3229	-75.2978	0.4226	196	0.7586
liline	95.9190	4.2459	28.3118	0.9893	1.3479	-58.0632	0.4805	206	0.5112
university	88.4816	4.5937	28.6623	7.7624e-04	1.0743	-46.5255	0.5581	250	0.6777

Tableau 3.4 – Mesures de performance par l’algorithme ROF, sigma=0.01.

Images	MSE	SNR	PSNR	IQI	NK	AD	SC	MD	NAE
cameraman	95.4334	3.9647	28.3338	-2.4160e-06	1.3366	-64.3582	0.4536	223	0.6382
barbara	96.1126	4.2188	28.3030	4.8606e-08	1.2895	-62.4257	0.4411	223	0.7338
camera	162.7122	1.9505	26.0166	3.1438e-05	1.1711	-47.1331	0.6584	234	0.3313
flower	95.6821	4.2095	28.3225	-7.1457e-07	1.5430	-88.4721	0.2649	182	1.1511
girl	163.1718	1.8132	26.0044	-1.5265e-07	1.3696	-82.4141	0.4053	196	0.7715
liline	164.7801	1.8959	25.9618	0.9893	1.4095	-66.9820	0.4521	188	0.5319
university	151.8053	2.2494	26.3179	7.7624e-04	1.1407	-57.3012	0.5196	250	0.6852

Tableau 3.5 – Mesures de performance par l’algorithme ROF, sigma=0.08.

Images	MSE	SNR	PSNR	IQI	NK	AD	SC	MD	NAE
cameraman	238.3766	-0.0110	24.3582	-1.5908e-07	1.4816	-89.4316	0.3828	190	0.7872
barbara	239.9024	0.2463	24.3305	4.8606e-08	1.4276	-81.4507	0.3819	223	0.8033
camera	236.3087	0.3299	24.3960	3.1438e-05	1.2246	-59.9995	0.6018	234	0.3897
flower	239.7857	0.2196	24.3326	-7.1457e-07	1.7601	-110.2763	0.2341	182	1.2374
girl	237.7749	0.1780	24.3691	-1.5265e-07	1.4474	-94.2270	0.3762	196	0.8140
liline	239.5990	0.2701	24.3360	0.9893	1.5124	-81.9270	0.4059	160	0.6046
university	219.8879	0.6402	24.7088	7.7624e-04	1.2430	-74.7657	0.4619	250	0.7667

Tableau 3.6 – Mesures de performance par l’algorithme ROF, sigma=0.2.

Chapitre 4

Modèle de Yves Meyer et Vese Osher

4.1 Introduction

Dans de nombreux problèmes d'analyse d'image, nous pouvons prendre une image f et trouver une autre image u qui est "proche" de f , faisant u est une version simplifiée de f . Typiquement, cela est écrit comme un problème inverse. La plupart des modèles supposent que $f = u + v$, où v représente le bruit ou (texture). En supposant que cela modélise le bruit, la composante v n'est généralement pas maintenu. La composante v peut être importante dans certaines circonstances, surtout si elle représente la texture. La texture est caractérisée comme un motif récurrent d'éléments minuscules. Le bruit est un motif de minuscules caractéristiques, pourtant il est composé de valeurs aléatoires, sans rapport. Les fonctions oscillatoires avec des valeurs positives et négatives et une moyenne nulle peuvent être utilisées pour décrire le bruit additif ou la texture ([41]).

En utilisant les concepts d'Yves Meyer de ([41]), nous montrons comment extraire de f les deux composantes u et v , en utilisant un cadre simple pour la minimisation de la variation totale développé par Rudin, Osher et Fatemi ([48]). Seules deux fonctions (g_1, g_2) sont nécessaires pour représenter la composante texturée v dans son intégralité. Nous fournissons un modèle qui combine le modèle de conservation de texture de Y. Meyer avec le modèle de préservation de bord de *ROF*. La minimisation de l'énergie et les équations aux dérivées partielles sont les méthodes utilisées. Pour simuler le bruit ou la texture, l'espace des fonctions oscillantes a été présenté pour la première fois par Yves Meyer ([41], [56]).

Afin de séparer la composante u dans BV et la composante v en tant que fonction oscillante (texture ou bruit) de f , Yves Meyer conseille d'utiliser un espace de fonctions appelé $G(\mathbb{R}^2)$, également connu sous le nom d'espace des fonctions oscillantes, qui est en quelque sorte le dual de l'espace BV . Meyer désigne l'espace G comme le dual de l'espace $W^{1,1}(\Omega)$ (l'ensemble des fonctions f telles que $\nabla f \in L^1(\Omega)^2$). Il a également ajouté deux espaces supplémentaires, indiqués par les lettres F et E . L'espace F est défini comme G , mais maintenant g_1, g_2 appartiennent à l'espace de John et Nirenberg $BMO(\Omega)$ au lieu de l'espace $L^\infty(\Omega)$. Enfin, Meyer définit le troisième espace E qu'il considère comme G , mais g_1, g_2 appartiennent à l'espace de Besov $B_\infty^{-1,\infty}(\Omega)$. E coïncide avec $B_\infty^{-1,\infty}(\Omega)$. Meyer montre que si la composante $v = f - u$ représente une texture ou un bruit, l'un des trois grands espaces G , F ou E doit être utilisé pour le modéliser.

4.2 Problème de Meyer

Yves Meyer ([56]) a examiné divers modèles de décomposition $f = u + v$. Il est possible d'étudier théoriquement le modèle *ROF* en divisant l'image en deux parties : les composantes géométriques u et v oscillantes. Y. Meyer propose un modèle dit "Décomposition en texture v et géométrie u ", la solution dans ce modèle est obtenue en minimisant :

$$(\mathcal{P}_{Meyer}) \quad \alpha_{Meyer} := \min_u \{ J(u) + \alpha \|v\|_G / u \in BV(\Omega), v \in G(\Omega), f = u + v \}. \quad (4.1)$$

Ici $\alpha > 0$ et $J(u)$ désigne la variation totale de u

$$J(u) = \sup \left\{ \int_{\Omega} u(x) \operatorname{div}(\varphi(x)) dx : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_{\infty} \leq 1 \right\}.$$

Y. Meyer donne la définition suivante :

Définition 4.2.1 ([41]) $G(\mathbb{R}^2)$ est un espace Banach construit de distributions v qui peut être écrit sous la forme

$$G(\mathbb{R}^2) := \{ v(x, y) = \partial_x g_1(x, y) + \partial_y g_2(x, y) / g_1, g_2 \in L^{\infty}(\mathbb{R}^2) \}, \quad (4.2)$$

avec g_1 et g_2 dans $L^{\infty}(\mathbb{R}^2)$. Nous verrons que l'espace G construit des fonctions oscillantes v , comme le justifie Y. Meyer, et que les oscillations sont bien mesurées par la norme

$$\|v\|_G = \inf \left\{ \begin{array}{l} \|g\|_{L^{\infty}(\mathbb{R}^2)} = \operatorname{ess\,sup}_{x \in \mathbb{R}^2} |g(x)| / v = \operatorname{div} \vec{g}, \\ \vec{g} = (g_1, g_2) \in L^{\infty}(\mathbb{R}^2) \times L^{\infty}(\mathbb{R}^2), |\vec{g}| = \sqrt{g_1^2 + g_2^2} \end{array} \right\}. \quad (4.3)$$

L'espace adapté au traitement d'image selon le modèle Meyer est l'espace des fonctions oscillantes $G(\mathbb{R}^2)$, qui modélise les signaux oscillants (l'espace de Banach contient des signaux oscillants), en particulier le bruit et les textures. La norme L^2 n'est pas la meilleure option pour capturer la partie oscillante d'une image. C'est pourquoi il a introduit l'espace G des fonctions oscillantes, un espace qui était déjà plus approprié.

Un problème de régularisation *TV* est une perte de contraste. Supposons f égale à la valeur de h dans un disque ou si le rayonnement externe est zéro,

$$f(x) = \begin{cases} h & \text{si } |x| \leq r, \\ 0 & \text{sinon.} \end{cases}$$

Si $\Omega = \mathbb{R}^2$, Meyer a démontré que le débruitage gaussien *TV*-régularisé diminue la valeur du disque en $\frac{2}{\lambda r}$, où

$$u(x) = \begin{cases} \max\{0, h - \frac{2}{\lambda r}\} & \text{si } |x| \leq r, \\ 0 & \text{sinon.} \end{cases}$$

On peut noter que si le contraste a diminué, la forme exacte du bord du cercle n'a pas changé.

4.2.1 La relation entre ROF et Meyer

On remarque que dans un problème Meyer utilise la même forme de problème que *ROF*. On utilise le paramètre $\alpha = \frac{1}{2\lambda}$ tel que $\lambda > 0$, et on modifie l'espace des fonctions v pour obtenir les meilleurs résultats. Meyer démontre les limites du modèle *ROF*, qui reconstruit u à partir de f . Ce problème de minimisation se produit dans l'espace BV .

$$\inf_{u \in BV} \{F(u) = \int_{\Omega} |\nabla u| + \lambda \int_{\Omega} |f - u|, f = u + v\}. \quad (4.4)$$

L'équation d'Euler-Lagrange associée est

$$u = f + \frac{1}{2\lambda} \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right).$$

Dans le modèle de Rudin-Osher-Fatemi, la fonction qui représente le bruit est définie comme suit :

$$\begin{cases} v = f - u = -\frac{1}{2\lambda} \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) = \operatorname{div} \vec{g} \quad \text{tel que} \quad \vec{g} = (g_1, g_2); \\ \text{avec} \\ g_1 = -\frac{1}{2\lambda} \frac{u_x}{|\nabla u|}; \quad g_2 = -\frac{1}{2\lambda} \frac{u_y}{|\nabla u|}; \quad \left\| \sqrt{g_1^2 + g_2^2} \right\|_{L^\infty}^2 = \frac{1}{2\lambda}. \end{cases}$$

Ce modèle *ROF* offre d'excellentes performances pour le débruitage des images tout en maintenant les frontières. Yves Meyer démontre en citant ([41], [56]) que le modèle *ROF* supprime la texture si λ est assez petit. Cependant, si le paramètre λ est trop petit, les détails les plus fins, comme la texture, sont perdus. Meyer a proposé un nouveau problème de minimisation pour régler ce problème. Il a étudié théoriquement le modèle *ROF* et a démontré que la texture sera éliminée par le modèle s'il est suffisamment petit. Meyer suggère également d'utiliser l'espace de fonctions oscillantes, qui est essentiellement le dual de l'espace BV , pour modéliser la texture ou le bruit. En remplaçant $L^2(\Omega)$ de $(f - u)$ (voir [4.4]), par un classique restreint qui est mieux adapté à la représentation de motifs texturés ou oscillatoires.

Lemme 4.2.1 *Soit f_n , $n \geq 1$ une suite de fonctions dans $L^2(D)$ avec les trois propriétés suivantes (où D est un disque centré en 0 et de rayon R) :*

1. *il existe un ensemble compact K tel que les supports des f_n , $n \geq 1$ sont inclus dans l'intérieur de K ;*
2. *il existe $q > 2$ et $C > 0$ tel que $\|f_n\|_{L^q(\mathbb{R}^2)} \leq C$;*
3. *la suite f_n converge vers 0 au sens distributionnel. Alors $\|f_n\|_G$ converge vers 0 lorsque n tend vers l'infini.*

Y. Meyer donne également (dans [41]), les trois résultats suivants :

Lemme 4.2.2 *Si $v \in L^2(\mathbb{R}^2)$, alors $\left| \int f(x, y)v(x, y)dx dy \right| \leq \|f\|_{BV} \|v\|_G$.*

Lemme 4.2.3 *La décomposition idéale de Rudin, Osher et Fatemi de f est donnée par $u = 0$ et $v = f$ si la norme de f en G ne dépasse pas $\frac{1}{2\lambda}$.*

Théorème 4.2.1 Soient f , u et v trois fonctions dans $L^2(\mathbb{R}^2)$. Si $\|f\|_G > \frac{1}{2\lambda}$, alors la décomposition de Rudin Osher Fatemi $f = u + v$ est caractérisée par les deux conditions :

$$\|v\|_G = \frac{1}{2\lambda} \quad \text{et} \quad \int u(x, y)v(x, y)dxdy = \frac{1}{2\lambda}\|u\|_{BV}.$$

4.2.2 Discrétisation du Modèle de Meyer

L'intérêt pour cet espace de fonctions oscillantes provient du fait de l'existence d'images fortement oscillantes avec de petites normes moyennes en $G(\mathbb{R}^2)$, c'est-à-dire que les fonctions G qui ont de grandes oscillations mais de petites normes moyennes et le fait que la norme L^2 n'est pas la meilleure option pour capturer des parties oscillantes d'images. Pour cette raison, il a développé un tout nouvel espace appelé l'espace G des fonctions oscillantes, qui est mieux adapté tout de suite. Nous avons une convergence pour la norme $\|\cdot\|_G$ pour les fonctions oscillantes avec une moyenne nulle et grand pour les autres fonctions.

Sur la base d'une approximation étroite de la norme L^∞ (cité dans ([56])), nous sommes arrivés à la conclusion suivante :

$$\left\| \sqrt{g_1^2 + g_2^2} \right\|_{L^\infty} = \lim_{p \rightarrow \infty} \left\| \sqrt{g_1^2 + g_2^2} \right\|_{L^p}, \quad g_1, g_2 \in L^\infty(\mathbb{R}^2).$$

Si $\lambda, \mu > 0$ sont des paramètres de réglage et $\lambda \rightarrow \infty$, $p \rightarrow \infty$, alors l'approximation du modèle Y. Meyer est donnée par

$$\inf_{u, g_1, g_2} \left\{ G_p(u, g_1, g_2) = \int |\nabla u| + \lambda \int |f - u - \partial_x g_1 - \partial_y g_2|^2 dxdy + \mu \left[\left(\sqrt{g_1^2 + g_2^2} \right)^p \right]^{\frac{1}{p}} \right\}, \quad (4.5)$$

avec

$$\int |\nabla u| \quad \text{assure que } u \in BV(\mathbb{R}^2),$$

$$\int |f - u - \partial_x g_1 - \partial_y g_2|^2 dxdy \quad \text{assure que } f \approx u + \text{div}(\vec{g}),$$

$$\mu \left[\left(\sqrt{g_1^2 + g_2^2} \right)^p \right]^{\frac{1}{p}} \quad \text{est une pénalité sur la norme } v = \text{div}(\vec{g}) \text{ dans } G.$$

En conséquence, la forme des équations d'Euler-Lagrange est fournie ici.

$$u = f - \partial_x g_1 - \partial_y g_2 + \frac{1}{2\lambda} \text{div} \left(\frac{\nabla u}{|\nabla u|} \right). \quad (4.6)$$

$$\mu \left(\left\| \sqrt{g_1^2 + g_2^2} \right\|_p \right)^{1-p} \left(\sqrt{g_1^2 + g_2^2} \right)^{p-2} g_1 = 2\lambda \left(\frac{\partial}{\partial x} (u - f) + \partial_{xx}^2 g_1 + \partial_{xy}^2 g_2 \right). \quad (4.7)$$

$$\mu \left(\left\| \sqrt{g_1^2 + g_2^2} \right\|_p \right)^{1-p} \left(\sqrt{g_1^2 + g_2^2} \right)^{p-2} g_2 = 2\lambda \left(\frac{\partial}{\partial y} (u - f) + \partial_{xy}^2 g_1 + \partial_{yy}^2 g_2 \right). \quad (4.8)$$

Les conditions aux limites pour $p = 1$ sont

$$\begin{cases} \frac{\nabla u}{|\nabla u|}(\eta_x, \eta_y) = 0, \\ (f - u - \partial_x g_1 - \partial_y g_2) \eta_x = 0, \\ (f - u - \partial_x g_1 - \partial_y g_2) \eta_y = 0. \end{cases}$$

4.3 Problème de Vese Osher

Pour réaliser le programme, Vese et Osher, utilisent une approximation de modèle de Y. Meyer, ce qui fait d'eux les premiers qui ont suggéré une méthode pour résoudre le problème de Meyer numériquement. La méthode est basée sur une version discrète de l'espace $G(\mathbb{R}^2)$. Vese et Osher suggèrent le modèle suivant :

$$(\mathcal{P}_{Vese-Osher}) \quad \alpha_{Vese-Osher} := \inf_{(u, v) \in BV \times G(\Omega)} \left\{ J(u) + \lambda \|f - u - v\|_2^2 + \mu \|v\|_G / \right\}, \quad (4.9)$$

où Ω est un ouvert borné, u la composante structure et v la composante oscillante.

On remarque que dans son problème Vese Osher utilise la même méthode que Meyer pour les premier et deuxième termes, en ajoutant le troisième terme $\lambda \|f - u - v\|_2^2$ et en conservant le même espace $G(\mathbb{R}^2)$ pour les fonctions oscillantes. Le but de ce triple termes est d'identifier une méthode pour résoudre le problème de Meyer numérique.

Les étapes que nous utilisons dans nos calculs numériques pour arriver à la réponse à ce problème sont :

1. remplacer l'expression $\|v\|_G$ par $\left\| \sqrt{g_1^2 + g_2^2} \right\|_p$, où $v = \text{div}(g_1, g_2)$;
2. pour les résonances numériques, $p = 1$ est utilisé car il permet des calculs plus rapides après chaque itération ;
3. fournit l'équation d'Euler Lagrange ;
4. avec une approche des différences finies semi-implicites, nous utilisons une technique itérative à point fixe.

4.3.1 Discrétisation Numérique du Modèle de Meyer

D'après ([56]), qui utilise la méthode des différences semi-implicite et la procédure itérative basée sur le point fixe, les équations (4.6), (4.7) et (4.8) sont critiquées numériquement. Les valeurs initiales de l'algorithme itératif sont les suivantes :

$$h = 1, \quad p = 1 \text{ et } n = 100.$$

$$\begin{cases} u_0 = f; \\ g_1^0 = -\frac{1}{2\lambda} \frac{f_x}{|\nabla f|}; \\ g_2^0 = -\frac{1}{2\lambda} \frac{f_y}{|\nabla f|}. \end{cases}$$

Les notions employées sont les suivantes : $u_{i,j} = u(ih, jh)$, $f_{i,j} = f(ih, jh)$, $g_{1,i,j} = g_1(ih, jh)$, avec l'étape $h > 0$ et le point (ih, jh) pour tous les $0 \leq i, j \leq M$, et le changement de variable est pris égale à

$$H(g_1, g_2) = \left(\left\| \sqrt{g_1^2 + g_2^2} \right\|_p \right)^{1-p} \left(\sqrt{g_1^2 + g_2^2} \right)^{p-2}.$$

La méthode des différences finies utilise les dérivés mixtes ainsi que les dérivés de premier et deuxième ordre :

$$\partial_x g = \frac{\partial g}{\partial x} = \frac{g_{i+1,j} - g_{i-1,j}}{2h};$$

$$\partial_y g = \frac{\partial g}{\partial y} = \frac{g_{i,j+1} - g_{i,j-1}}{2h};$$

$$\partial_{xx} g = \frac{\partial^2 g}{\partial^2 x} = \frac{g_{i+1,j} - 2g_{i,j} + g_{i-1,j}}{h^2};$$

$$\partial_{yy} g = \frac{\partial^2 g}{\partial^2 y} = \frac{g_{i,j+1} - 2g_{i,j} + g_{i,j-1}}{h^2};$$

$$\partial_{xy} g = \frac{\partial^2 g}{\partial x \partial y} = \frac{1}{4h^2} (g_{i+1,j+1} + g_{i-1,j-1} - g_{i+1,j-1} - g_{i-1,j+1});$$

$$\nabla g = \left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \right) = \left(\frac{g_{i+1,j} - g_{i-1,j}}{2h}, \frac{g_{i,j+1} - g_{i,j-1}}{2h} \right);$$

$$\text{div} g = \frac{\partial^2 g}{\partial^2 x} + \frac{\partial^2 g}{\partial^2 y} = \frac{1}{h^2} (g_{i+1,j} - 4g_{i,j} + g_{i-1,j} + g_{i,j+1} + g_{i,j-1}).$$

Par conséquent, voici la discrétisation des équations (4.6), (4.7) et (4.8).

$$u_{i,j}^{n+1} = \frac{1}{1 + \frac{1}{2\lambda h^2} (c_1 + c_2 + c_3 + c_4)} \left(\begin{aligned} & f_{i,j} - \frac{g_{1,i+1,j}^n - g_{1,i-1,j}^n}{2h} - \frac{g_{2,i,j+1}^n - g_{2,i,j-1}^n}{2h} \\ & + \frac{1}{2\lambda h^2} (c_1 u_{i+1,j}^n + c_2 u_{i-1,j}^n + c_3 u_{i,j+1}^n + c_4 u_{i,j-1}^n) \end{aligned} \right); \quad (4.10)$$

$$g_{1,i,j}^{n+1} = \frac{2\lambda}{\mu H(g_{1,i,j}, g_{2,i,j})} \left(\begin{aligned} & \frac{u_{i+1,j}^n - u_{i-1,j}^n}{2h} - \frac{f_{i+1,j} - f_{i-1,j}}{2h} + \frac{g_{1,i+1,j}^n - 2g_{1,i,j}^{n+1} + g_{1,i-1,j}^n}{h^2} \\ & + \frac{1}{4h^2} (g_{2,i+1,j+1}^n + g_{2,i-1,j-1}^n - g_{2,i+1,j-1}^n - g_{2,i-1,j+1}^n) \end{aligned} \right); \quad (4.11)$$

$$g_{2,i,j}^{n+1} = \frac{2\lambda}{\mu H(g_{1,i,j}, g_{2,i,j})g_{2,i,j}} \left(\begin{aligned} & \frac{u_{i,j+1}^n - u_{i,j-1}^n}{2h} - \frac{f_{i,j+1} - f_{i,j-1}}{2h} + \frac{g_{2,i,j+1}^n - 2g_{2,i,j}^n + g_{2,i,j-1}^n}{h^2} \\ & + \frac{1}{4h^2} (g_{1,i+1,j+1} + g_{1,i-1,j-1} - g_{1,i+1,j-1} - g_{1,i-1,j+1}) \end{aligned} \right). \quad (4.12)$$

On a les notations suivantes :

$$\begin{aligned} c_1 &= \frac{1}{\sqrt{\left(\frac{u_{i+1,j}^n - u_{i,j}^n}{h}\right)^2 + \left(\frac{u_{i,j+1}^n - u_{i,j-1}^n}{2h}\right)^2}}; \\ c_2 &= \frac{1}{\sqrt{\left(\frac{u_{i,j}^n - u_{i-1,j}^n}{h}\right)^2 + \left(\frac{u_{i-1,j+1}^n - u_{i-1,j-1}^n}{2h}\right)^2}}; \\ c_3 &= \frac{1}{\sqrt{\left(\frac{u_{i+1,j}^n - u_{i-1,j}^n}{2h}\right)^2 + \left(\frac{u_{i,j+1}^n - u_{i,j}^n}{h}\right)^2}}; \\ c_4 &= \frac{1}{\sqrt{\left(\frac{u_{i+1,j-1}^n - u_{i-1,j-1}^n}{2h}\right)^2 + \left(\frac{u_{i,j}^n - u_{i,j-1}^n}{h}\right)^2}}. \end{aligned} \quad (4.13)$$

4.3.2 Résolution du Problème de Vese Osher

Afin de résoudre le problème de Vese Osher, nous allons explorer ce dernier problème dans le cas discriminant, lorsque l'image est un vecteur avec deux dimensions de taille $N \times N$, l'espace Eulidien $X = \mathbb{R}^{N \times N}$ et $Y = X \times X$.

Si $u \in X$, alors $\nabla u \in Y$ est définie par $(\nabla u)_{i,j} = \left((\nabla u)_{i,j}^1, (\nabla u)_{i,j}^2 \right)$, avec

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{si } i < N, \\ 0 & \text{si } i = N. \end{cases} \quad \text{et } (\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{si } j < N, \\ 0 & \text{si } j = N. \end{cases} \quad (4.14)$$

Dans le cas de discrétisation, la variance totale (TV) de u est définie comme :

$$J_d(u) = \sum_{1 \leq i,j \leq N} |(\nabla u)_{i,j}|.$$

L'opérateur de divergence est $div = -\nabla^*$ (l'adjoint de ∇) avec

$$(div(p))_{i,j} = \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{si } 1 < i < N, \\ p_{i,j}^1 & \text{si } i = 1, \\ -p_{i-1,j}^1 & \text{si } i = N. \end{cases} + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{si } 1 < j < N, \\ p_{i,j}^2 & \text{si } j = 1, \\ -p_{i,j-1}^2 & \text{si } j = N. \end{cases} \quad (4.15)$$

De plus, l'espace G^d est défini par

$$G^d := \{v \in X / \exists g \in Y \text{ tel que } v = \text{div}(g)\}.$$

Sachant que

$$\begin{aligned} G_\mu(\Omega) &= \{v \in G(\Omega) / \|v\|_G \leq \mu\}. \\ G_\mu^d(\Omega) &= \{v \in G^d(\Omega) / \|v\|_{G^d} \leq \mu\}. \end{aligned}$$

J_d^* est la fonction indicateur de $G_1^d(\Omega)$, qui est définie par

$$J_d^*(v) = \chi_{G_1^d(\Omega)}(v) = \begin{cases} 0 & \text{si } v \in G_1^d, \\ +\infty & \text{sinon.} \end{cases}$$

Par conséquent, nous proposons l'algorithme suivant pour résoudre le problème de Vese Osher :

Algorithme 4.3.1 *Algorithme pour résoudre le problème de Vese Osher*

$$\inf_{(u,v) \in BV(\Omega) \times G_\mu(\Omega)} F_{\lambda,\mu}(u,v) = \begin{cases} J(u) + \frac{1}{2\lambda} \|f - u - v\|_2^2 & \text{si } v \in G_\mu(\Omega) \\ +\infty & \text{si } v \in G(\Omega) \setminus G_\mu(\Omega). \end{cases}$$

Plus précisément ;

$$\inf_{(u,v) \in X \times X} F_{\lambda,\mu}(u,v) = \begin{cases} J_d(u) + \frac{1}{2\lambda} \|f - u - v\|_X^2 & \text{si } v \in G_\mu^d(\Omega) \\ +\infty & \text{si } v \in X \setminus G_\mu^d(\Omega). \end{cases}$$

$$\inf_{(u,v) \in X \times X} F(u,v) = \inf_{(u,v) \in X \times X} \left\{ J_d(u) + \frac{1}{2\lambda} \|f - u - v\|_X^2 + J_d^*\left(\frac{v}{\mu}\right) \right\}.$$

Nous divisons le problème en deux sous-problèmes :

$$(Pbm1) \quad u \text{ solution, } v \text{ fixe : } \quad \inf_{v \in X} \left\{ J_d(u) + \frac{1}{2\lambda} \|f - u - v\|_X^2 \right\}.$$

\curvearrowright La solution est
selon ROF
changement de variable
 $f = f - v$

$$u^\wedge = f - v - P_{G_\mu^d}(f - v).$$

$$(Pbm2) \quad v \text{ solution, } u \text{ fixe : } \quad \inf_{v \in G_\mu^d(\Omega)} \left\{ \|f - u - v\|_X^2 \right\}.$$

\curvearrowright la solution est
selon ROF
changement de variable
 $f = f - u$

$$v^\wedge = P_{G_\mu^d}(f - u).$$

Lemme 4.3.1 *Il existe une solution unique $(\hat{u}, \hat{v}) \in X \times G_\lambda^d$ qui minimise $F_{\lambda,\mu}(u, v)$ dans $X \times G_\lambda^d$.*

Algorithme 4.3.2 *Algorithme de Vese Osher*

1- Initialisation : $u_0 = v_0 = 0$.

2- Itération : $v_{n+1} = P_{G_\mu}(f - u_n)$; $u_{n+1} = f - v_{n+1} - P_{G_\lambda}(f - v_{n+1})$.

3- Test d'arrêt : $(|u_{n+1} - u_n|, |v_{n+1} - v_n|) \leq \varepsilon$.

Chapitre 5

Théorèmes de Projection de Bregman et Projection Oblique

5.1 Introduction Historique

Aujourd'hui, le traitement d'images est une branche du traitement du signal dédiée au traitement des images et des vidéos. Le traitement d'images est un ensemble d'opérations effectuées sur des images, destinées à améliorer la lisibilité et à faciliter l'interprétation des images. La récupération d'image est un problème très intéressant dans l'industrie. C'est un problème fondamental dans divers domaines des sciences appliquées comme l'imagerie médicale (Ngwa et Agyingi, [45]; Sidky et al., [51]) la microscopie, l'astronomie (Brune, Sawatzky et Burger, [15]), la récupération de film (Chan, Golub et Mulet, [21]) et l'encodage d'images et de vidéos (Chan, Golub et Mulet, [21]; Chen, Wu et Yu, [22]). La récupération d'image est un problème intéressant en traitement d'image car il se situe au début de la chaîne de collecte, qui consiste à restaurer une image originale propre à partir d'une image dégradée ([15], [51]). Ce type de mesure de distance en imagerie a été utilisé pour la première fois par Bregman en 1967 (Bregman ([15])). Ce concept a été inventé et développé par Censor et Lent (Censor et Lent ([27])). En 1976, Bregman a découvert une technique élégante et efficace pour l'utilisation de la fonction D_g dans le processus de conception et d'analyse d'algorithmes de faisabilité et d'optimisation. Cela a ouvert un domaine de recherche croissant dans lequel la technique de Bregman est appliquée de diverses manières à la conception et à l'analyse d'algorithmes itératifs non seulement pour résoudre des problèmes de faisabilité et d'optimisation, mais également pour résoudre des inégalités et calculer des points fixes de cartes non linéaires. Pour plus de détail on peut voir les références (Bauschke, Borwein and Combettes ([6]); Butnariu, Censor and Reich ([13]); Butnariu and Iusem ([14])).

La projection de Bregman est une technique utilisée dans le domaine du traitement d'image et de la vision par ordinateur pour résoudre des problèmes d'optimisation avec des contraintes de régularisation. Elle est basée sur la distance de Bregman (Bregman ([15])), pour mesurer la dissimilarité entre une solution proposée et une référence. La distance de Bregman est une mesure asymétrique qui tient compte de la structure spécifique des données et peut capturer des informations de corrélation entre les pixels. La projection de Bregman est souvent utilisée dans des problèmes d'imagerie où une certaine régularisation est nécessaire pour obtenir une

solution assez régulière et cohérente. Elle permet de combiner des contraintes sur la fidélité des données et des contraintes sur la régularité spatiale de l'image.

L'idée principale de la projection de Bregman est d'exprimer le problème d'optimisation non convexe sous la forme d'une série de problèmes d'optimisation convexe plus simples. Ces problèmes sont résolus de manière itérative en utilisant des techniques d'optimisation convexe telle que la descente de gradient. En utilisant la projection de Bregman, il est possible de trouver une solution qui minimise la fonction objectif tout en respectant les contraintes imposées. Cette technique a été appliquée avec succès dans divers domaines de l'imagerie, tels que la déconvolution d'image, la reconstruction d'image à partir de données incomplètes ou bruitées, la segmentation d'image, etc.

5.2 Théorème de Projection de Bregman

Définition 5.2.1 Soient V un espace de Banach réflexif, X un sous-ensemble fermé et convexe non vide de V , et l'espace dual de X noté X^* . Soit $g : X \rightarrow \mathbb{R}$ une fonction convexe, $u \in X$, $p \in X^*$. On appelle p le sous-gradient de g en u si

$$\forall v \in X : g(v) - g(u) - \langle p, v - u \rangle \geq 0.$$

L'ensemble de tous les sous-gradients de g en u , est appelé le sous-différentiel de g et noté $\partial g(u)$.

Définition 5.2.2 Soit $g : X \rightarrow \mathbb{R}$. On appelle dérivée directionnelle de g en u dans la direction v la quantité

$$g'(u, v) = \lim_{\lambda \rightarrow 0^+} \frac{g(u + \lambda v) - g(u)}{\lambda}.$$

Si $\exists p \in X^*$ telle que $g'(u, v) = \langle p, v \rangle \forall v \in X$ alors on dit que g est Gâteaux différentiable en u et p est appelé le Gâteaux-différentiel de g en u noté $g'(u)$.

Proposition 5.2.1 Si $g : X \rightarrow \mathbb{R}$ est continue en $u \in X$ et g admet un seul sous-gradient alors g est Gâteaux-différentiable en $u \in X$ et on a $\partial g(u) = \{g'(u)\}$.

En outre, g admet un minimum en $u \in X$ si et seulement si, $0 \in \partial g(u)$, dans ce cas on a

$$g(v) - g(u) \geq \langle 0, v - u \rangle = 0.$$

Définition 5.2.3 ([38], [4]) (Fonction de Legendre) Supposons que g soit une fonction convexe propre sur X ($X \subset \mathbb{R}^d$ espace Euclidien) avec $\text{int}(\text{dom}(g)) \neq \emptyset$. Alors g est une fonction de type Legendre ou convexe de type Legendre, si elle satisfait les conditions suivantes :

- g est différentiable sur $\text{int}(\text{dom}(g))$,
- $\lim_{\lambda \rightarrow 0^+} \langle \nabla g(x + \lambda(y - x)), y - x \rangle = -\infty, \forall x \in \text{FV}(\text{dom}(g)), y \in \text{int}(\text{dom}(g))$,
- g est strictement convexe sur $\text{int}(\text{dom}(g))$.

Remarque 5.2.1 ([3]) On note Γ l'ensemble des fonctions convexes, semi-continues inférieurement et propres sur \mathbb{R}^d .

1. On dit que $g \in \Gamma$ est une fonction convexe de type Legendre, c'est-à-dire, g est essentiellement lisse et essentiellement strictement convexe.
2. g est 1-coercive, c'est-à-dire, $\lim_{\|x\| \rightarrow +\infty} \frac{g(x)}{\|x\|} = +\infty$. Une condition équivalente est $\text{dom } g^* = \mathbb{R}^d$.
3. l'ensemble C est un sous-ensemble fermé non vide de $U = \text{int}(\text{dom}(g))$.

5.2.1 Distance de Bregman

La distance de Bregman est une mesure de dissimilarité entre deux points dans un espace vectoriel. Elle a de nombreuses applications, notamment en optimisation non convexe, en traitement du signal et en traitement d'image. Elle est utilisée pour formuler des problèmes d'optimisation et de régularisation, pour construire des algorithmes itératifs et pour mesurer la convergence et la précision des résultats. Il existe plusieurs exemples de distances de Bregman, tels que la distance euclidienne, la divergence de Kullback-Leibler, la divergence de Itakura-Saito, etc.

Définition 5.2.4 ([38] (*Distance de Bregman 1*)) Soit X un espace de Banach et $g : X \rightarrow]-\infty, +\infty[$ une fonction convexe, semi-continue inférieurement et propre, alors g est de type Legendre et $C \subset \text{int}(\text{dom}g)$ un ensemble non vide, g différentiable sur $\text{int}(\text{dom}g) \neq \emptyset$. Alor la distance de Bregman D_g est définie par

$$\begin{cases} D_g : X \times \text{int}(\text{dom}g) \rightarrow [0, +\infty] \\ (x, y) \longrightarrow D_g(x, y) = g(x) - g(y) - \langle \nabla g(y), x - y \rangle. \end{cases} \quad (5.1)$$

Remarque 5.2.2 ([38] Soit $g : X \rightarrow]-\infty, +\infty[$ la fonction de Legendre avec la dérivée à droite g'_+ dans la direction h ,

$$g'_+(x, h) = \lim_{t \rightarrow 0^+} \frac{g(x + th) - g(x)}{t}.$$

Il existe une autre façon de définir la distance de Bregman par rapport à g entre les points $x, y \in \text{dom}(g)$:

$$D_g(y, x) = g(y) - g(x) - g'_+(x, y - x) \quad \forall x, y \in \text{dom}(g);$$

$$D_g(y, x) = g(y) - g(x) + g'_+(x, x - y) \quad \forall x, y \in \text{dom}(g);$$

$$D_g(y, x) = g(x) + g^*(\nabla_g(y)) - \langle \nabla_g(y), x \rangle \quad \forall x \in X, y \in \text{int}(\text{dom}g);$$

$$D_g(y, x) = D_{g^*}(\nabla_g(y), \nabla_g(x)) \quad \forall x, y \in \text{int}(\text{dom}g).$$

Définition 5.2.5 (*Distance de Bregman 2*) Supposons $g : X \rightarrow \mathbb{R}$ une fonction convexe, $u, v \in X$, et $p \in \partial g(u)$ un sous-gradient du sous-différentiel de g au point u , alors la distance de Bregman entre les points u, v est définie par

$$D_g^p(u, v) = g(u) - g(v) - \langle p, u - v \rangle.$$

Définition 5.2.6 ([38]) *La fonction distance de Bregman sur C est définie par*

$$D_C^g(x) := \inf_{y \in C} D_g(y, x), \quad \forall x \in \text{int}(\text{dom}(g)). \quad (5.2)$$

Définition 5.2.7 ([38]) (**Projection de Bregman**) *Soit $g : X \rightarrow \mathbb{R}$ une fonction de Legendre. Soit C un ensemble convexe fermé dans X avec $C \cap \text{int}(\text{dom}(g)) \neq \emptyset$. Alors le problème d'approximation $\inf_{y \in C} D_g(y, x)$ admet une solution unique $\text{proj}_C^g(x)$ appelé la projection de Bregman de x sur C qui est définie par*

$$\begin{cases} \text{proj}_C^g(x) := \arg \min_{y \in C} D_g(y, x) \in \text{int}(\text{dom}(g)), \\ \text{proj}_C^g(x) := \{y \in C : D_g(y, x) = D_C^g(x)\} \quad \forall x \in \text{int}(\text{dom}(g)). \end{cases} \quad (5.3)$$

La projection de Bregman $\text{proj}_C^g(x)$ sur l'ensemble C par rapport à g est définie comme l'ensemble des solutions du problème d'optimisation $\inf_{y \in C} D_g(y, x)$:

$\inf_{y \in C} D_g(y, x) \rightarrow$ le problème d'optimisation de Bregman est le minimum de distance de Bregman,

$\text{proj}_C^g(x) = \arg \min_{y \in C} D_g(y, x) \rightarrow$ la solution du problème est la projection de Bregman.

Lemme 5.2.1 *Le problème de Bregman (\mathcal{P}_α) admet une solution unique, où*

$$(\mathcal{P}_\alpha) \quad \alpha := \inf_{y \in C} D_g(y, x). \quad (5.4)$$

Démonstration. *En effet, on a g est une fonction convexe de type de Legendre semi-continue inférieurement, propre, c'est-à-dire,*

i) g est une fonction strictement convexe si pour tout $x, y \in X$ avec $x \neq y$ et tout $\lambda \in [0, 1]$, on a :

$$g(\lambda x + (1 - \lambda)y) < \lambda g(x) + (1 - \lambda)g(y);$$

ii) g est une fonction semi-continue inférieurement si, et seulement si,

$$\forall x_0 \in X, \quad \liminf_{x \rightarrow x_0} g(x) \geq g(x_0);$$

iii) g est une fonction propre sur X si, et seulement si,

$$g(x) > -\infty \quad \forall x \in X \quad \text{et} \quad g \neq +\infty;$$

iv) g est Gâteaux-différentiable si, et seulement si,

$$g'_+(x, h) = \lim_{t \rightarrow 0^+} \frac{g(x + th) - g(x)}{t}$$

la dérivée à droite g'_+ dans la direction h .

Comme g est une fonction de Legendre alors il existe un point unique

$$\text{proj}_C^{g(x)} = \arg \min_{y \in C} D_g(y, x) \in \text{int}(\text{dom}(g)),$$

solution du problème de Bregman (\mathcal{P}_α)

$$\alpha := \inf_{y \in C} D_g(y, x).$$

□

Exemple 5.2.1 (*Distance de Bregman*) Soit

$$C_0 = \mathbb{R}^n, \quad g_0(x) = \frac{1}{2} \|x\|^2,$$

alors la distance Euclidienne

$$D_{g_0}(x, y) = \frac{1}{2} \|x - y\|^2.$$

est une distance de Bregman.

Exemple 5.2.2 (*Distance de Bregman*) Soit

$$C_1 = \mathbb{R}_{++}^n := \{x \in \mathbb{R}^n, x_i \geq 0, i = 1 \dots n\}, \quad g_1(x) = x \log(x) - x + 1,$$

alors la K-L divergence (*Kullback-Leibler divergence*) $D_{g_1} : \mathbb{R}_+^n \times \mathbb{R}_{++}^n \rightarrow \mathbb{R}$ définit par :

$$D_{g_1}(x, y) = \sum_{i=1}^n \left(x_i \log\left(\frac{x_i}{y_i}\right) + y_i - x_i \right),$$

est une distance de Bregman.

Proposition 5.2.2 Soit $g_1 : \mathbb{R}_+^n \rightarrow \mathbb{R}$ deux fois continue différentiable et strictement convexe dans \mathbb{R}_+^n . Soit $g_1(1) = g_1'(1) = 0$, $g_1''(1) > 0$, et $\lim_{t \rightarrow 0^+} g_1'(t) = -\infty$. Pour la distance de Bregman D_g , on a

- (i) $D_g(x, y) \geq 0$, pour tout $(x, y) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$.
- (ii) $D_g(x, y) = 0$ si, et seulement si, $x = y$.
- (iii) Les ensembles de niveaux de $D_g(\cdot, y)$ sont bornés, pour tout $y \in \mathbb{R}_+^n$.
- (iv) Les ensembles de niveaux de $D_g(x, \cdot)$ sont bornés, pour tout $x \in \mathbb{R}_+^n$.
- (v) $D_g(\cdot, y)$ strictement convexe en x .
- (vi) $\lim_{v \rightarrow \infty} D_g(y^*, y^v) = 0$ si, et seulement si $\lim_{v \rightarrow \infty} y^v = y^*$.

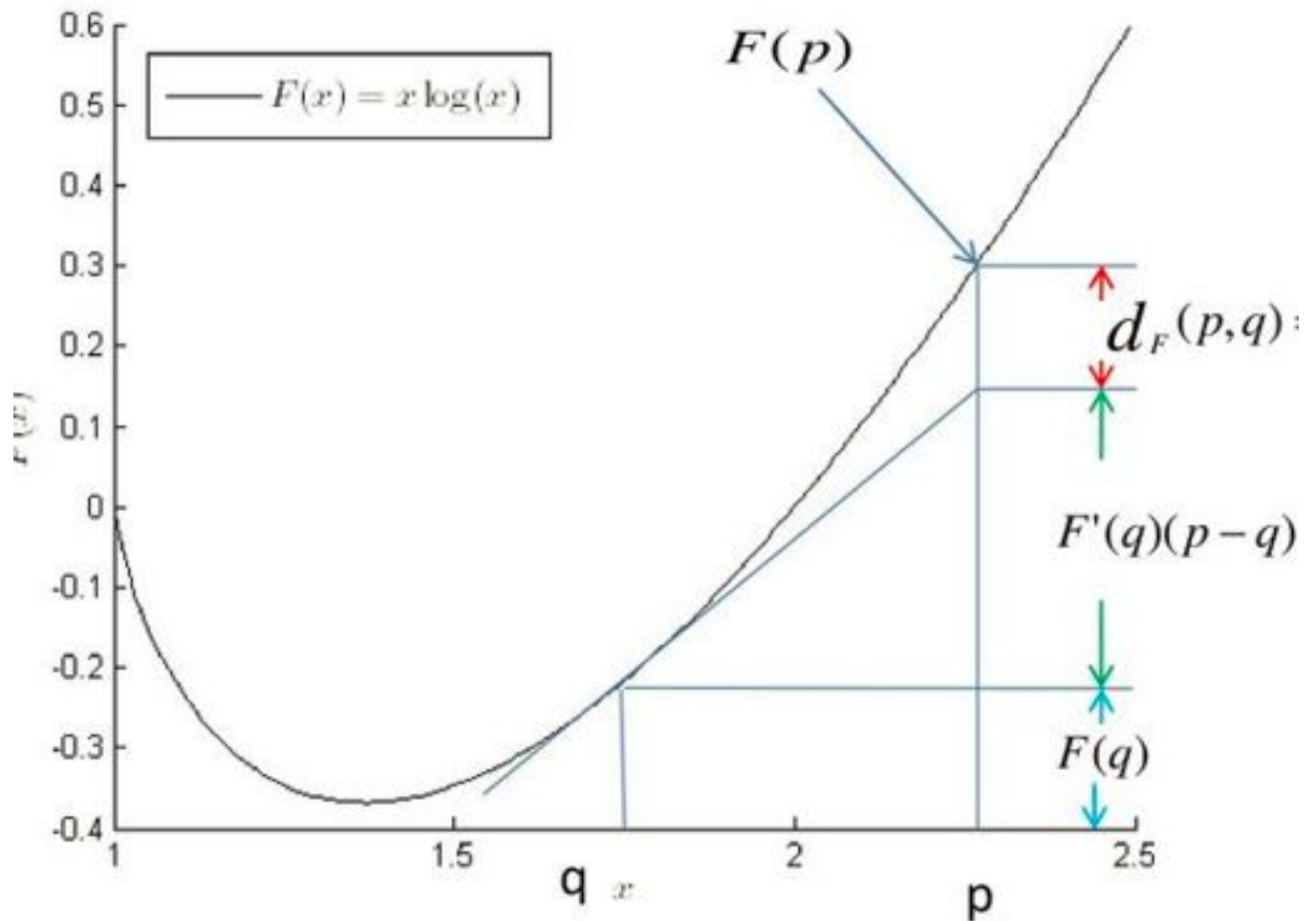


Figure 5.1 – Distance de Bregman

5.3 Théorème de Projection Oblique

5.3.1 Introduction au projection oblique

Le terme “projection oblique” fait typiquement référence à un procédé de représentation d’un objet tridimensionnel dans un espace bidimensionnel. Dans ce type de projection, l’objet est représenté avec une ou plusieurs de ses faces formant un angle avec le plan de projection ; cet angle n’est généralement pas l’un des angles de projection standard. Les projections obliques sont couramment utilisées dans les dessins techniques, l’architecture et l’ingénierie

pour fournir une représentation plus réaliste de la forme tridimensionnelle d'un objet tout en conservant la simplicité d'un dessin bidimensionnel. Dans une projection oblique, la profondeur de l'objet est représentée en mettant à l'échelle les dimensions de l'objet le long d'un axe tout en laissant les deux autres dimensions inchangées. Cela crée une illusion de profondeur, l'objet apparaissant comme s'il avait été incliné ou de travers.

La projection oblique est une technique de correction spectrale chimiométrique qui permet d'éliminer les effets de diffusion multiple dans les photographies hyperspectrales des paramètres de végétation. La méthode de projection oblique présuppose que les spectres des images hyperspectrales sont divisés en deux parties : une partie utilisable liée à la propriété d'intérêt et un composant non utile en raison des effets de diffusion et d'éclairage. Les procédures de projection orthogonale sont un moyen simple de supprimer la composante non utile des données. Toutefois, comme la partie non utile a une intersection non vide avec des renseignements utiles, son retrait complet peut entraîner une perte d'information.

La méthode de projection oblique utilise une projection spectrale non orthogonale pour minimiser la perte d'information tout en supprimant les effets des fluctuations d'éclairage dans la mesure du possible. Avant d'effectuer une projection, la projection oblique nécessite la détermination des sous-espaces utilisables et non utiles. Le sous-espace non utile, c'est-à-dire les effets d'éclairage, est défini en estimant des termes polynomiaux jusqu'à un degré maximal à l'aide de spectres. Le degré de polynôme peut être choisi en fonction de la complexité des effets d'éclairage.

La projection oblique, également connue sous le nom de projection orthogonale oblique, est une technique utilisée pour projeter une image ou un objet tridimensionnel sur un plan en utilisant une transformation linéaire. Elle est souvent utilisée pour générer des vues $2D$ à partir d'objets $3D$ dans le domaine de la vision par ordinateur et de la modélisation $3D$. Bien que la projection oblique puisse être utilisée dans des problèmes d'optimisation ou d'analyse d'image, elle n'est pas directement liée à la distance de Bregman. Les deux concepts sont utilisés dans des contextes différents et ont des applications distinctes.

Cependant, il est possible d'appliquer des techniques d'optimisation utilisant la distance de Bregman pour résoudre des problèmes liés à la projection oblique, par exemple en utilisant des méthodes de régularisation pour estimer les paramètres de la transformation de projection oblique. Dans ce cas, la distance de Bregman serait utilisée pour formuler la fonction objectif et guider la recherche des paramètres optimaux.

En résumé, la projection oblique et la distance de Bregman sont deux concepts indépendants dans le domaine du traitement d'image et de l'optimisation. elles sont utilisées dans des contextes différents et ont des applications distinctes, bien qu'il soit possible de combiner les techniques d'optimisation utilisant la distance de Bregman dans des problèmes liés à la projection oblique.

5.3.2 Projection Oblique

En mathématiques, le terme "projection oblique" peut avoir une signification légèrement différente par rapport à son utilisation en graphisme ou en ingénierie. En mathématiques, une projection oblique fait référence à un type de transformation qui transforme un vecteur ou un point sur un sous-espace de manière non orthogonale. Considérons un espace vectoriel

V , et soit W un sous-espace de V . Une projection oblique sur W est une transformation linéaire $P : V \mapsto W$ qui applique les vecteurs de V sur W , telle que pour tout vecteur v de V , la projection $P(v)$ est le vecteur le plus proche de v dans W .

La principale différence entre une projection oblique et une projection orthogonale est qu'une projection orthogonale transforme des vecteurs sur un sous-espace de manière perpendiculaire (orthogonale). Dans une projection oblique, les vecteurs cartographiés n'ont pas besoin d'être perpendiculaires au sous-espace. Il convient de noter que les projections obliques sont plus générales que les projections orthogonales et sont utilisées dans divers contextes mathématiques, tels que l'algèbre linéaire, l'optimisation et l'analyse fonctionnelle.

Pour définir une projection oblique, nous devons spécifier à la fois le sous-espace W et la direction de projection. La direction de projection détermine la manière dont les vecteurs sont transformés sur le sous-espace. La projection oblique P peut être représentée par une matrice, dite matrice de projection. Soit A une matrice dont les colonnes forment une base pour le sous-espace W . Alors la matrice de projection P est donnée par

$$P = A(A^T A)^{-1} A^T; \quad (5.5)$$

ici, A^T représente la transposée de la matrice A et $(A^T A)^{-1}$ désigne l'inverse du produit de la transposée de A et A . Multiplier un vecteur v par la matrice de projection P donne la projection oblique de v sur le sous-espace W .

Une projection parallèle avec des lignes de vision qui ne sont pas perpendiculaires au plan de projection est appelée projection oblique; (Tom McReynolds, David Blythe, in *Advanced Graphics Programming Using OpenGL*, 2005). Les projections obliques sont couramment utilisées lorsque le plan de projection est perpendiculaire à un axe de coordonnées et que les lignes de vision sont déplacées pour croiser deux autres côtés de l'objet. Les longueurs et les angles des faces d'objets parallèles au plan sont préservés à la suite de la projection. Si les courbes sur les faces de l'objet sont parallèles au plan de projection, des projections obliques peuvent être utiles.

Considérons le point (x_0, y_0, z_0) projeté à la position (x_p, y_p) pour obtenir une projection oblique. Les projecteurs sont spécifiés par deux angles : θ et ϕ , où θ est l'angle entre le plan de projection et la ligne $L = [(x_0, y_0), (x_p, y_p)]$, et ϕ est l'angle entre la ligne L et l'axe des x . On pose $l = \|L\|/z_0 = 1/\tan \theta$, la forme générale de la projection oblique est donnée par

$$P = \begin{pmatrix} 1 & 0 & l \cos \phi & 0 \\ 0 & 1 & l \sin \phi & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Le projecteur est perpendiculaire dans une projection orthographique, et la longueur de la ligne L est zéro, réduisant la matrice de projection à l'identité.

Les projections cavalières et de cabinet sont deux projections obliques ordinaires. Avec des lignes de vision à 45 degrés, la projection cavalière préserve les longueurs de lignes qui sont perpendiculaires ou parallèles au plan de projection. Cependant, comme l'œil humain est habitué à compenser le raccourcissement de la perspective, le fait que la longueur est conservée pour les lignes perpendiculaires crée une illusion d'optique dans laquelle les lignes perpendiculaires apparaissent plus longues qu'elles ne le sont. Pour compenser cela, la projection

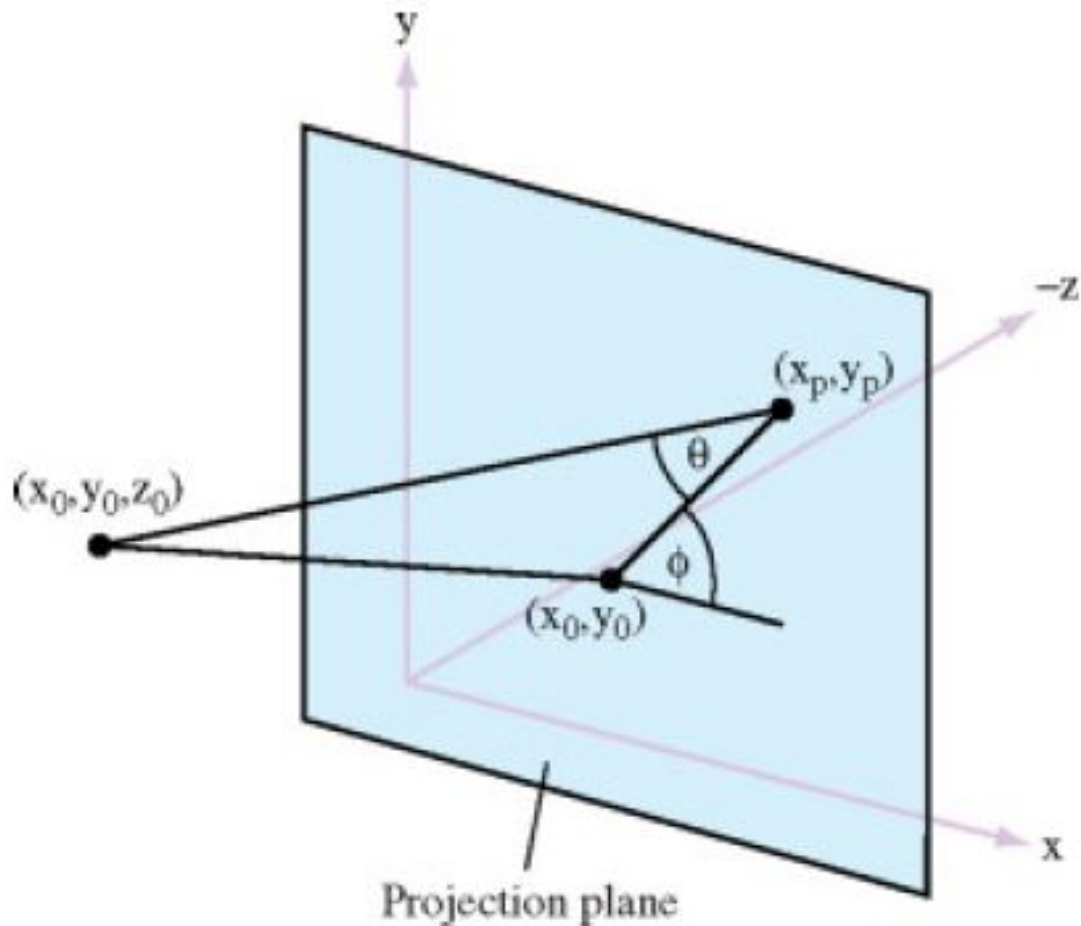


Figure 5.2 – Projection oblique.

de cabinet réduit les lignes perpendiculaires à la moitié de la longueur des lignes parallèles et modifie l'angle θ à $\tan(2) = 63.43$ degrés. La transformation de projection est calculée à l'aide de la matrice P , tandis que la matrice de projection ortho-graphique donne le reste de la description du volume de la vue. Le plan de projection est supposé être perpendiculaire à l'axe z dans la matrice P . Avant d'appliquer la projection, une transformation supplémentaire peut être utilisée pour changer la direction de visualisation.

5.3.3 Projection oblique itérative sur des ensembles convex et problèmes de faisabilité divisée (SFP)

5.3.4 Problèmes (SFP) et l'algorithme CQ

Soit A une matrice réelle de taille $M \times N$ et C, Q des ensembles convexes fermés non vides de \mathbb{R}^N et \mathbb{R}^M , respectivement. D'après ([16]), soient

$$A(C) = \{AC \mid c \in C\}; A^{-1}(Q) = \{x \mid Ax \in Q\}; F = C \cap A^{-1}(Q).$$

Le problème de faisabilité divisée (**SFP**) introduit par Censor et Elfving ([19]), est le suivant :

$$(\mathbf{SFP}) := \begin{cases} \text{Trouver } x \in C \text{ avec } Ax \in Q; \\ \text{si } x \text{ existe.} \end{cases} \quad (5.6)$$

Trouver un membre de l'intersection des deux ensembles Q et $A(C)$, ou de l'intersection des deux ensembles $A^{-1}(Q)$ et C , est l'équivalent de résoudre (**SFP**). Pour résoudre **SFP** on utilise une méthode itérative connue sous le nom l'algorithme CQ qui n'implique pas l'inverse de la matrice A .

Les méthodes de Landweber et de Landweber projetées pour obtenir des solutions exactes ou approximatives des équations linéaire $Ax = b$ sont des exemples notables de l'algorithme CQ . La technique de reconstruction algébrique (ART) de Gordon et al est un cas particulier de la $BICQ$ qui est un algorithme CQ itératif par bloc, peut être ajoutée pour accélérer davantage l'algorithme.

Les principales caractéristiques de l'algorithme CQ ne nécessitent pas d'inversions de matrice et utilisent simplement des projections orthogonales simples. La technique CQ est simplifiée lorsque la matrice A est pleine puisque la limite supérieure L peut être facilement prédite à partir du nombre d'entrées non nulles de A . Cela accélère également la convergence en permettant de prendre des valeurs plus grandes.

De nombreux problèmes de reconstruction d'image peuvent être exprimés comme des problèmes de faisabilité divisées (**SFP**) ([16]). Chaque niveau d'intensité de pixel est représenté par une entrée dans le vecteur x , qui représente une image vectorisée. On peut choisir l'ensemble C pour inclure des caractéristiques comme la nonnégativité. Les mesures linéaires fonctionnelles ou de projection que nous avons effectuées, ainsi que les combinaisons linéaires supplémentaires d'éléments de x sur lesquels nous souhaitons placer des contraintes peuvent être décrites par la matrice A . Le vecteur de données mesurées peut alors être combiné avec des ensembles convexes supplémentaires, comme les cônes non négatifs, afin de créer l'ensemble Q , qui sert de description des restrictions imposées.

La reconstruction d'images tomographiques par émission dynamique, dans laquelle le vecteur x représente la concaténation de nombreuses images correspondant à des temps discrets séquentiels, est une de ces applications qui fait l'objet de recherches actives. Afin de limiter le comportement des intensités aux pixel fixes dans le temps et d'assurer la cohérence (ou presque) avec les données mesurées, la matrice A et l'ensemble Q peuvent alors être choisis. Ce qui suit est l'algorithme CQ pour résoudre le problème (**SFP**).

Algorithme 5.3.1 ([16]) Algorithme CQ

Soit x_0 arbitraire. Pour $k = 0, 1, \dots$

L'étape itérative de la méthode CQ est la suivante :

$$x^{k+1} = P_C(x^k + \gamma A^T(P_Q - I)A x^k), \quad (5.7)$$

où $\gamma \in (0, 2/L)$ est la plus grande valeur propre de la matrice $A^T A$ et P_C, P_Q sont des projections orthogonales, aussi appelée carte de proximité sur C et Q respectivement ; c'est-à-dire

$$P_C(x) := \min \{\|c - x\|, \text{ pour tout } c \in C\}.$$

Il est courant de décrire un problème de reconstruction d'image comme un problème **(CFP)**, dans lequel nous cherchons une image vectorisée x à l'intersection d'un grand nombre d'ensembles convexes. Le **(SFP)** peut être considéré comme un cas particulier du **(CFP)**.

Soit K_j , $j=1, \dots, J$, des sous-ensembles convexes fermés non vides de \mathbb{R}^M avec intersection non vide K . Le problème de faisabilité convexe **(CFP)** est de trouver un élément de K .

Définition 5.3.1 ([16]) *La projection oblique : Soit A une matrice réelle de taille $M \times N$ et soient C , Q des ensembles convexes fermés non vides dans \mathbb{R}^N et \mathbb{R}^M , respectivement. Lorsque $M = N$ et A est inversible, l'application*

$$P_{A^{-1}(Q)}^{A^T A} = A^{-1} P_Q A. \quad (5.8)$$

est une projection oblique sur l'ensemble $A^{-1}(Q)$; c'est-à-dire

$$A^{-1} P_Q (Ax) := \min \{ f(z) = (z - x)^T A^T A (z - x) \}; \forall z \in A^{-1}(Q).$$

Le contre-exemple suivant nous montre que la suite itérative

$$x^{k+1} = P_C P_{A^{-1}(Q)}^{A^T A} (x^k), \quad (5.9)$$

ne doit pas converger vers une solution.

Soit $M = N = 2$, C l'axe horizontal, Q l'axe vertical et A est la matrice suivantes :

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 0 \end{pmatrix} \quad (5.10)$$

Soit $x^0 = (0, 1)^t$ un point de départ. Alors $Ax^0 = (1, 1)^t$; $P_Q(Ax^0) = (0, -1)^t$; $A^{-1}P_Q(Ax^0) = (1, 1)^t$ et $P_C A^{-1}P_Q(Ax^0) = (1, 0)^t = x^0$. Dans cet exemple, la solution au **(SFP)** est $(0, 0)$, ce qui montre que (5.9), est une suite itérative qui ne parvient pas à converger.

Cependant, les projections peuvent être utilisées pour créer des algorithmes itératifs convergents à condition qu'un type de relaxation généralisée soit inclus, basé sur des distances généralisées distinctes à chaque étape.

5.3.5 Convergence de l'algorithme CQ

Soit F l'ensemble des $c \in C$ (qui peut ou non être vide) auquel la fonction $\|P_Q A c - A c\|$ atteint sa valeur minimale sur C . En ce qui concerne l'algorithme CQ, nous avons le théorème suivant.

Théorème 5.3.1 [16] $F \neq \emptyset$. Pour chaque vecteur initial x^0 , la suite x^k décrite par l'équation (5.7), converge vers un membre de F .

Corollaire 5.3.1 [16] Chaque fois qu'il y a des solutions **(SFP)**, la suite x^k décrite par l'équation (5.7), converge vers l'une de ces solutions.

5.3.6 Les méthodes Landweber

Nous considérons la méthode projetée de Landweber et l'approche de Landweber [35] comme exemples spéciaux de l'algorithme CQ. Le livre de Bertero et Boccacci [7] contient une discussion détaillée de ces méthodes, principalement dans le contexte de la restauration de l'image dans les espaces de dimensions infinies ces fonctions (voir aussi [36]). L'algorithme CQ se transforme en approche itérative de Landweber pour résoudre les équations linéaires $Ax = b$ lorsque $C = \mathbb{R}^N$ et $Q = b$.

Algorithme 5.3.2 ([16]) *L'algorithme de Landweber*

Pour x^0 arbitraire et $k = 0, 1, \dots$, soit

$$x^{k+1} = x^k + \gamma A^T(b - Ax^k) \quad (5.11)$$

Pour C convexe fermé non vide, nous obtenons la méthode Landweber projetée pour trouver une solution de $Ax = b$ sur C .

Algorithme 5.3.3 ([16]) *L'algorithme Landweber projeté*

Pour x^0 arbitraire et $k = 0, 1, \dots$, soit

$$x^{k+1} = PC(x^k + \gamma A^T(b - Ax^k)). \quad (5.12)$$

À partir de la preuve de la convergence de l'algorithme CQ, il s'ensuit que l'algorithme de Landweber converge vers une solution des moindres carrés de $Ax = b$ et l'algorithme Landweber projeté converge vers un minimiseur $\|Ac - b\|$ pour tout $c \in C$, chaque fois que de telles solutions existent.

5.3.7 Le CFP et l'algorithme MSFP

Soit $T = P_J P_{J-1} \dots P_1$, la projection orthogonale sur l'ensemble convexe fermé non vide $K_j \subseteq \mathbb{R}^N$ pour $j = 1, \dots, J$, P_j indiquent la projection sur cet ensemble. La suite $\{T^k x^0\}$ converge vers un membre de K pour tout vecteur initial x^0 si l'intersection K des K_j est non vide. Ceci est le résultat de l'approche des projections orthogonales successives (SOP).

Bregman introduit la méthode des projections généralisées successives (SGP) dans [11]. Il démontre que la convergence est maintenue si les projections orthogonales P_j sont remplacées par des projections définies par rapport à une seule distance généralisée de Bregman fixe. La convergence peut être perdue si T est constitué de projections liées à diverses distances de Bregman, comme le montrent les contre-exemples. Au-delà du théorème de Cheney-Goldstein pour le cas de projections orthogonales et $J = 2$, peu est connu sur le cas de $K = \emptyset$.

La technique itérative SGP de Bregman est étendue par l'algorithme MSGP pour prendre en compte les projections de Bregman correspondant à diverses distances de Bregman.

Soient S_h et S_f domaines respectives pour les fonctions Bregman h et f . Supposons que D_h et D_f satisfont $D_h(x, z) \geq D_f(x, z)$ pour tous les $x \in \overline{S_h} \cap \overline{S_f}$ et $z \in S_h \cap S_f$ pour leurs distances Bregman associées. Pour $x \in \overline{S_h} \cap \overline{S_f}$, $y \in S_f$ et $z \in S_h \cap S_f$, soit la fonction $G(x, y, y, z, f, h)$ de x définie comme :

$$G(x, y, y, z, f, h) = D_f(x, y) + D_h(x, z) - D_f(x, z)$$

Soit $f_j, j = 1, \dots, J$ une famille de fonctions de Bregman, et h une fonction de Bregman qui "domine" la famille, c'est à dire, $D_h(x, z) \geq D_{f_j}(x, z)$ pour tous j et tous $x \in \overline{S_h} \cap \overline{S_f}$ et $z \in S_h \cap S_f$. Soit $S = S_h \cap (\bigcap_{i=1}^J S_i)$; $P_{K_j}^{f_j} x$ représente le minimiseur de $D_{f_j}(z, x)$ pour $z \in K_j$.

L'algorithme *MSGP* est le suivant :

Algorithme 5.3.4 ([16]) *L'algorithme MSGP*

Pour $k = 0, 1, \dots$

Après avoir calculer x^k nous pouvons alors calculer x^{k+1} comme suit :

Soit $G^k(x) := G(x, P_{K_j}^{f_j}(x^k), x^k, f_j, h); j = k(\text{mod}J) + 1$.

En supposant que $G^k(x)$ possède un unique minimiseur, nous choisissons x^{k+1} . nous supposons également que $x^{k+1} \in S_h$, pour que

$$\nabla h(x^{k+1}) = \nabla h(x^k) - \nabla f_j(x^k) + \nabla f_j(P_{K_j}^{f_j}(x^k)). \quad (5.13)$$

Enfin, nous supposons que nous avons une cohérence de zone cyclique; c'est à dire pour chaque k , le vecteur x^{k+1} décrit par (5.13) est en S_{f_m} , $m = (k+1)(\text{mod}J) + 1$.

On a le théorème de convergence suivant :

Théorème 5.3.2 Soit $K \cap S \neq \emptyset$. Toute suite x^k produite par la méthode itérative décrite par l'algorithme *MSGP* converge vers l'un des membres de $K \cap S$.

Comme nous venons de le mentionner, si P_Q est déterminé, alors la projection oblique $P_{A^{-1}(Q)}^{A^T A}$ est $A^{-1}P_Q A$ lorsque A est inversible. Avec $K_1 = C$ et $K_2 = A^{-1}(Q)$, l'algorithme *MSGP*, et les projections et distances définies par les fonctions de Bregman $h(x) = f_1(x) = x^T x$ et $f_2(x) = x^T A^T A x$, respectivement, nous pouvons décrire l'algorithme *CQ*, à condition que $\gamma \in (0, \frac{1}{L})$. Comme indiqué précédemment, la technique *CQ* est toujours valable pour $\gamma \in (0, \frac{1}{L})$ et pour toute matrice A .

5.4 Algorithmes Itérative de Bregman

La régularisation itérative de Bregman a été initialement introduite et étudiée en traitement d'images par Osher et al en [46]. Dans [26] Osher, Burger, Goldfarb, Xu et Yin ont proposé l'algorithme itératif de Bregman comme algorithme efficace pour résoudre des problèmes d'optimisation, des problèmes de minimisation convexe sous contraintes, leurs idées principales étaient d'abord de transformer un problème d'optimisation contraint en un problème sans contrainte, puis, au moyen de la distance de Bregman. Cet algorithme appliqué au problème :

$$\min_u \{J(u) + H(u, f)\} \quad (5.14)$$

tel que $J : X \rightarrow \mathbb{R}$, $H : X \rightarrow \mathbb{R}$ sont des fonction convexe non négative de $u \in X$ pour un f fixé, avec X un ensemble convexe fermé. De plus le terme de fidélité $H(u, f)$ est supposée différentiable. Rappelons que f est un vecteur ou une matrice.

Comme cas partuculier on prend le problème de débruitage de la variation totale $H(u) = \frac{1}{2\lambda} \|u - b\|_2^2$, c'est un problème du ROF . Au lieu de résoudre (\mathcal{P}_{ROF}) une fois, la procédure de régularisation itérative de Bregman de Osher et Al résout une suite de problèmes convexe :

$$u^{k+1} := \begin{cases} \min_u J(u) + \frac{1}{2} \|u - b^{k+1}\|_2^2, \\ \text{avec } b^{k+1} = b + (b^k - u^k). \end{cases} \quad (5.15)$$

La différence entre (\mathcal{P}_{ROF}) et (5.15), réside dans l'utilisation de la régularisation, alors que (\mathcal{P}_{ROF}) régularise u en minimisant directement sa variation totale, (5.15), fait de même en minimisant la distance de Bregman basée sur la variation totale de u a une solution précédente u^k . Alors pour trouver le résultat on applique l'algorithme itératif de Bregman qui est un algorithme efficace et plus rapide pour les problèmes d'optimisation convexe et sans contrainte. On définit le sous-gradient par

$$p \in \partial J(u) \iff J(u) \geq J(v) + \langle p, u - v \rangle \forall u.$$

Algorithme 5.4.1 Algorithme Itératif de Bregman 1

Initialisation : u_0 donnée, $\gamma > 0$.

$u^{k+1} \leftarrow \arg \min D_J^{P^k}(u, u^k) + \gamma H(u); p^k \in \partial J(u^k)$.

$0 \in \partial J(u^{k+1}) - p^k + \gamma H(u^{k+1})$ alors $p^k + \gamma H(u^{k+1}) \in \partial J(u^{k+1})$.

Si H différentiable $\nabla H = \partial H$, on a $p^k \in \partial J(u^k)$.

Alors $p^{k+1} \in \partial J(u^{k+1})$, on prend $p^{k+1} = p^k + \gamma H(u^{k+1})$.

Algorithme 5.4.2 Algorithme Itératif de Bregman 2

Initialisation : $p^0 \in \partial J(u^0)$.

Pour $k = 0, 1, \dots$ faire

$u^{k+1} \leftarrow \arg \min D_J^{P^k}(u, u^k) + \gamma H(u); p^k \in \partial J(u^k)$.

$p^{k+1} = p^k + \gamma H(u^{k+1})$.

Cas partique d'application de débruitage; si $u \in \mathbb{R}^n$ avec des contraintes d'égalité linéaire, A est une matrice (si on prend $A = I$ c'est le modèle de ROF) et l'ensemble

$$H(u) = \frac{1}{2} \|Au - f\|_2^2 \implies \nabla H = Au - f.$$

Alors l'algorithme itératif de Bregman simplifié est défini par

Algorithme 5.4.3 Algorithme Itératif de Bregman Simplifié 3

Initialisation : $u^0 \in \mathbb{R}^n, b^0 = 0$.

Pour $k = 0, 1, \dots$ faire

$u^{k+1} \leftarrow \arg \min J(u) + \frac{\gamma}{2} \|Au - f - b^k\|_2^2$.

$b^{k+1} = b^k + Au^{k+1} - f$.

Lorsque les contraintes sont linéaire, l'itération de Bregman est équivalente à la méthode lagrangienne augmentée (également connue sous le nom de méthode des multiplicateurs). Les sous-gradients p^k sont représentés par les variables auxiliaires b^k , qui sont ajoutés dans le terme de pénalité quadratique.

Osher, Burger, Goldfarb, Xu, and Yin définit l'algorithme régularisation itérative de Bregman comme suit :

Algorithme 5.4.4 *Algorithme Régularisation Itérative de Bregman*

Soit : $J(\cdot)$, $H(\cdot)$.

Initialisation : $k = 0$, $u_0 = 0$, $p_0 = 0$.

Tan que “ u^k ne converge pas ” faire

$$u^{k+1} \leftarrow \arg \min D_J^{p^k}(u, u^k) + \frac{1}{2\lambda} \|u - b\|_2^2.$$

$$p^{k+1} \leftarrow p^k - \nabla H(u^{k+1}) \in \partial J(u^{k+1}).$$

$k \leftarrow k + 1$.

Fin tant que.

Le but de l’algorithme itératif de Bregman est :

- appliquer à de nombreux problème de débruitage d’image comme modèle ROF ;
- donner des bonnes propriétés de convergence ;
- on a la convergence vers l’image originale ;
- la convergence en terme de distance de Bregman ;
- algorithme de split Bregman pour résoudre le problème ROF.

5.4.1 Théorème de Convergence

L’itération de Bregman a été proposée pour la première fois dans [46] pour le débruitage d’images basé sur la TV . D’autres propriétés et d’analyse de convergence de ce schéma itératif de Bregman a été prouvée en détail dans ([46], [15]). La première itération de cet algorithme $u^1 = \min_u (J(u) + H(u))$. Pour résoudre le problème initial nous avons besoin que le terme résiduel soit minimal. Quand le terme résiduel converge, alors l’algorithme itératif de Bregman continue. L’algorithme itératif de Bregman a été appliquée à de nombreux problèmes. Par exemple les problèmes mal posé et le débruitage d’image car il présente de très bonne propriétés de convergence. Parmi ces propriétés : la convergence vers l’image originale que nous essayons de récupérer avec des données exactes, et la convergence en terme de distance de Bregman vers l’image original avec des donnée bruités, la diminution monotone du terme résiduel. Lors de l’étude du débruitage d’images, nous avons $J(u) = \|u\|_{BV}$ où $\|\nabla u\|_1$ et $H(u) = \frac{1}{2} \|u - f\|_2^2$.

Nous avons une suite générée par l’algorithme qui décroît de façon monotone vers H .

Proposition 5.4.1 ([12]) *On a décroissance monotone de H*

$$H(u^{k+1}) \leq H(u^{k+1}) + D_J^{p^k}(u^{k+1}, u^k) \leq H(u^k).$$

Proposition 5.4.2 ([12]) *La suite de termes résiduels $\{H(u^k)\}$ converge vers la valeur minimale de H*

Si \tilde{u} minimise $H : X \rightarrow \mathbb{R}$ et $J(\tilde{u}) < \infty$, alors

$$H(u^k) \leq H(\tilde{u}) + \frac{J(\tilde{u})}{k}.$$

Chapitre 6

Analyses Comparatives des Algorithmes de Proximale Pénalité et de Bregman pour le Débruitage D'image

Introduction

La restauration d'images est un problème intéressant et mal posé. Elle joue un rôle essentiel dans la conception de traitement d'image. Nous recherchons une image qui est aussi proche de l'originale que possible parmi les images qui ont été biaisées par le bruit gaussien et additive. La déconstruction d'image est une technique permettant de restaurer une image bruyante après sa capture. Les résultats numériques sont obtenus par la méthode proximal-pénalité, l'algorithme de split Bregman pour la TV anisotrope et isotrope décrivant des problèmes en termes de qualité d'image, de convergence et de taux de bruit du signal (SNR) sont comparés dans cette thèse. Remarquons que la TV isotrope décrite est plus rapide que l'anisotrope. Les résultats expérimentaux indiquent que l'algorithme prox donne la meilleure production de haute qualité (propre, non lisse et les textures sont préservées). En particulier, nous avons obtenu (21.4, 21) le SNR de l'image décrite par Prox pour sigma égale à 0.08 et 0.501, de même nous avons obtenu (10.0884, 10.1155) le SNR de l'image décrite par la TV anisotrope et la TV isotrope pour sigma égale à 0.08 et (-1.4635, -1.4733) pour sigma 0.501.

6.1 Méthodes Proximale-Pénalité

En appliquant un algorithme proximal pour résoudre un problème de minimisation, on propose une technique unique pour la restauration d'image. On prévoit qu'une image originale a été dégradée par un bruit additif. Nous essayons de reconstruire u de l'image que nous avons vu Im (qui est donc une version dégradée de l'image originale). La technique du maximum de vraisemblance nous amène à rechercher u comme solution aux problèmes d'optimisation

suivants, en supposant que le bruit additif est gaussien :

$$(\mathcal{P}) \quad \alpha := \arg \min_{u \in U^{ad}} \left\{ \frac{1}{2} \|Im - u\|_2^2 \right\},$$

où U^{ad} est l'ensemble admissible déterminé par

$$U^{ad} := \{u \in BV(\Omega), J(u) \leq 0\},$$

et $J(u)$ représente la variation totale en u telle que spécifiée par

$$J(u) := \sup \left\{ \int_{\Omega} u(x) \operatorname{div}(\varphi(x)) dx : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_{\infty} \leq 1 \right\}.$$

La régularisation de la variation totale (TV) est un terme de régularisation $J(u) = |D(u)|$ qui permet des solutions discontinues. En raison de sa capacité à inclure des "sauts" dans la solution, il est devenu particulièrement populaire pour l'image. L'espace des fonctions à variations bornées est connu comme BV . Il est caractérisé par

$$BV(\Omega) := \{u \in L^1(\Omega), J(u) < +\infty\}.$$

Pour résoudre ce problème, nous appliquerons l'approche de pénalité proximale (Aujol ([1]); Michelli, Shen, and Xux ([43])). Pour ce faire, nous énonçons l'idée de la méthode. Le fait que le problème (\mathcal{P}) est équivalent au problème (\mathcal{P}_{α_e}) est la principale idée.

$$(\mathcal{P}) \quad \alpha := \arg \min_{u \in U^{ad}} \left\{ \frac{1}{2} \|Im - u\|_2^2 \right\} \iff (\mathcal{P}_{\alpha_e}) \quad \alpha_e := \arg \min_{u \in BV} \left\{ \frac{1}{2} \|Im - u\|_2^2 + \Psi_{U^{ad}}(u) \right\},$$

où $\Psi_{U^{ad}}$ indique la fonction indicateur de U^{ad} définie par

$$\Psi_{U^{ad}}(u) := \begin{cases} 0 & \text{si } u \in U^{ad}, \\ +\infty & \text{sinon.} \end{cases}$$

Selon ([23, 24, 25, 27, 28]), le principe fondamental des méthodes de proximale pénalité est le suivant :

1) remplacer le problème (\mathcal{P}) par le problème (\mathcal{P}_r) sans contraintes

$$(\mathcal{P}_r) \quad \alpha_r := \arg \min_{u \in BV} \left\{ \varphi(u, r) = \frac{1}{2} \|Im - u\|_2^2 + r \cdot \max(0, \Pi_K(u)) \right\}.$$

Le coefficient de pénalité est r avec $r > 0$. La fonction de pénalité extérieur est désignée par h , où $h(u) = \max(0, \Pi_K(u))$, et $\Pi_K(u)$ est la projection de u sur K où

$$K := \{ \operatorname{div}(\varphi(x)) : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_{\infty} \leq 1 \}.$$

Lorsque ($r \rightarrow +\infty$) la solution obtenue $u(r)$ est une solution de (\mathcal{P}).

- 2) On commence par sélectionner un coefficient de pénalité r_1 puis on résout le problème sans contraintes

$$(\mathcal{P}_{r_1}) \quad \alpha_{r_1} := \arg \min_{u \in BV} \left\{ \varphi(u, r_1) = \frac{1}{2} \|Im - u\|_2^2 + r_1 \cdot \max(0, \Pi_K(u)) \right\},$$

soit $u(r_1)$ le point obtenu.

- 3) Si la quantité $r_1 h(u(r_1))$ est suffisamment petite, le test d'arrêt : $u(r_1)$ est une bonne estimation de la valeur optimale ; sinon, un coefficient de pénalité $r_2 > r_1$ sera calculé et le prochain nouveau problème sera résolu sans contraintes (\mathcal{P}_{r_2}).

Le problème est lié à

$$(\mathcal{P}_r)_w \quad \alpha_r(w) := \arg \min_{(u,w) \in BV^2} \left\{ \frac{1}{2} \|Im - u\|_2^2 + r \cdot \max(0, \Pi_K(u)) + \frac{1}{2} \|u - w\|_2^2 \right\}.$$

L'algorithme de relaxation transforme et génère une suite $\{u^k, w^k\}_k$ de sorte que u^{k+1} soit une solution de problème lorsque l'on applique à ce problème.

$$(\mathcal{P}_r)_{w^k} \quad \alpha_r(w^k) := \arg \min_{u \in BV} \left\{ \frac{1}{2} \|Im - u\|_2^2 + r \cdot \max(0, \Pi_K(u)) + \frac{1}{2} \|u - w^k\|_2^2 \right\},$$

et la solution du problème est w^{k+1} telle que

$$(\mathcal{P}_r)_{u^{k+1}} \quad \alpha_r(w) := \arg \min_{u \in BV} \left\{ \frac{1}{2} \|Im - u\|_2^2 + r \cdot \max(0, \Pi_K(u^{k+1})) + \frac{1}{2} \|u^{k+1} - w\|_2^2 \right\}.$$

En conséquence, une simple itération u^{k+1} résout le problème suivant :

$$(\mathcal{P}_r)_{u^k} \quad \alpha_r := \arg \min_{u \in BV} \left\{ \frac{1}{2} \|Im - u\|_2^2 + r \cdot \max(0, \Pi_K(u)) + \frac{1}{2} \|u - u^k\|_2^2 \right\}.$$

Algorithme 6.1.1 *Algorithme Proximale-Pénalité*

Étape 0 : ($k = 0$) soit $u^0 \in \mathbb{R}^n$, $\varepsilon > 0$ une précision.

Étape 1 : Nous utilisons l'approche de minimisation pour trouver une solution u^1 au problème suivant en utilisant un coefficient de pénalité r^0 et une précision $\delta > 0$.

$$\arg \min_{u \in \mathbb{R}^n} \left\{ \frac{1}{2} \|Im - u\|_2^2 + r_0 \cdot \max(0, \Pi_K(u)) + \frac{1}{2} \|u - u^0\|_2^2 \right\}.$$

Étape 2 : La solution obtenue est $u^1(r_0) = u^1$.

Si $\|u^1 - u^0\|_2^2 < \varepsilon$ et si $r_0 h(u^1(r_0)) < \delta$, alors u^1 est une excellente approximation de la solution optimale, et le calcul s'arrête à l'itération $k + 1$.

Sinon, choisissons $r_1 > r_0$ comme coefficient de pénalité. Posons $r_0 = r_1$, $u^0 = u^1$, $k = k + 1$ et revenons à l'étape 1.

6.2 Algorithmes de Bregman et Imagerie

L'algorithme itératif de Bregman a été appliquée à de nombreux problèmes, par exemple le problème mal posé, le débruitage d'image, la recherche de base ..., car il présente de très bonnes propriétés de convergence. Parmi ces propriétés : la convergence vers l'image originale que nous essayons de récupérer avec des données exactes, et la convergence en terme de distance de Bregman à l'image originale avec des données bruitées.

Dans le modèle de régularisation itératif utilisant la distance de Bregman basée sur la fonctionnelle de variation totale

$$J(u) = \lambda TV(u) = \lambda \int |\nabla u|.$$

Plus précisément, la distance de Bregman basée sur une fonction convexe $J(\cdot)$ entre les points u et v est définie comme suit :

$$D_J^p(u, v) = J(u) - J(v) - \langle p, u - v \rangle,$$

où $p \in \partial J(v)$ est un certain sous-gradient du sous-différentiel de J au point v . Comme $D_J^p(u, v) \neq D_J^p(v, u)$ en général c'est-à-dire n'est pas symétrique, $D_J^p(u, v)$ n'est pas une distance au sens habituel.

6.2.1 Algorithme de Split Bregman

L'algorithme de split Bregman a été initialement introduite par Goldstein et Osher en ([29]) pour résoudre les problèmes d'optimisation plus généraux de la forme :

$$\varpi := \min_{u \in X} \{H(u) + \|\Phi(u)\|_1\}, \quad (6.1)$$

où X est un ensemble convexe et fermé, $\Phi : X \rightarrow \mathbb{R}$ et $H : X \rightarrow \mathbb{R}$ sont des fonctions convexes. Ce problème est équivalent au problème de minimisation avec contraintes suivant :

$$\begin{cases} \overline{\varpi} := \min_{u \in X, d \in \mathbb{R}} \{H(u) + \|d\|_1\}, \\ \text{sujet à } d = \Phi(u). \end{cases} \quad (6.2)$$

L'algorithme de split Bregman introduit par Goldstein et Osher a été écrit sous la forme suivante :

Algorithme 6.2.1 *Algorithme de split Bregman*

Initialisation : $k = 0$, $u^0 = 0$, $b^0 = 0$;

Tant que $\|u^k - u^{k-1}\| > \text{tol}$ *faire* ;

$$u^{k+1} = \min_u H(u) + \frac{\lambda}{2} \|d^k - \Phi(u) - b^k\|_2^2;$$

$$d^{k+1} = \min_d |d| + \frac{\lambda}{2} \|d - \Phi(u^{k+1}) - b^k\|_2^2;$$

$$b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1});$$

$$k = k + 1.$$

Fin tant que

L'algorithme de split Bregman appliqué à plusieurs problèmes d'optimisation de la forme :

$$\widehat{w} := \min_{u \in X} \left\{ z(u) + \frac{1}{2} \|u - f\|_2^2 \right\}. \quad (6.3)$$

On applique l'algorithme de split Bregman aux problèmes de débruitage TV anisotrope et débruitage TV isotrope.

6.2.2 Problème de Débruitage TV Anisotrope

Nous considérons le problème de débruitage TV anisotrope ([12]) suivant :

$$(\mathcal{P}_1) \quad \tau_1 := \min_u \left\{ \left\| \frac{\partial u}{\partial x} \right\|_1 + \left\| \frac{\partial u}{\partial y} \right\|_1 + \frac{\mu}{2} \|u - f\|_2^2 \right\}, \quad (6.4)$$

où f est l'image bruitée. Nous noterons $\frac{\partial u}{\partial x}$ par u_x et $\frac{\partial u}{\partial y}$ par u_y . On considère le problème avec contraintes équivalent au problème (\mathcal{P}_1) :

$$(\mathcal{P}_2) \quad \begin{cases} \tau_2 := \min_u (\|d_x\|_1 + \|d_y\|_1 + \frac{\mu}{2} \|u - f\|_2^2), \\ \text{sujet à } d_x = u_x, d_y = u_y. \end{cases}$$

Pour résoudre le problème (\mathcal{P}_2) , nous résolvons le problème sans contraintes suivant :

$$(\mathcal{P}_3) \quad \tau_3 := \min_{u, d_x, d_y} \left\{ \|d_x\|_1 + \|d_y\|_1 + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - u_x\|_2^2 + \frac{\lambda}{2} \|d_y - u_y\|_2^2 \right\}. \quad (6.5)$$

On peut ainsi résoudre ce dernier problème en utilisant l'algorithme de split Bregman.

On définit

$$\text{shrink}(x, a) = \begin{cases} x - a & \text{si } x > a, \\ x + a & \text{si } x < -a, \\ 0 & \text{ailleurs,} \end{cases} \quad (6.6)$$

et la fonction de Gauss-Seidel

$$G_{i,j}^k = \frac{\lambda}{\mu + 4\lambda} (u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k + d_{x,i-1,j}^k + d_{x,i,j}^k + d_{y,i,j-1}^k + d_{y,i,j}^k + b_{x,i-1,j}^k + b_{x,i,j}^k + b_{y,i,j-1}^k + b_{y,i,j}^k) + \frac{\mu}{\mu + 4\lambda} f_{i,j}. \quad (6.7)$$

Algorithme 6.2.2 *Algorithme de split Bregman de problème de débruitage TV anisotrope :*

Initialisation : $k = 0, u^0 = 0, b^0 = 0$.

Tant que $\|u^k - u^{k-1}\| > \text{tol}$ *faire,*

$u^{k+1} = G^k$ *où* G *est la fonction de Gauss-Seidel ;*

$d_x^{k+1} = \text{shrink}(\nabla_x u^{k+1} + b_x^k, \frac{1}{\lambda})$.

$d_y^{k+1} = \text{shrink}(\nabla_y u^{k+1} + b_y^k, \frac{1}{\lambda})$.

$$\begin{aligned} b_x^{k+1} &= b_x^k (\nabla_x u^{k+1} - d_x^{k+1}). \\ b_y^{k+1} &= b_y^k (\nabla_y u^{k+1} - d_y^{k+1}). \\ k &= k + 1. \end{aligned}$$

Fin tant que.

6.2.3 Problème de Débruitage TV Isotrope

Nous considérons d'abord le problème de débruitage *TV* Isotrope ([12]) suivant :

$$(\mathcal{P}'_1) \quad I_{S_1} := \min_u \left\{ \|\nabla u\|_1 + \frac{\mu}{2} \|u - f\|_2^2 \right\}. \quad (6.8)$$

On considère le problème avec contraintes équivalent au problème (\mathcal{P}'_1) .

$$(\mathcal{P}'_2) \quad \begin{cases} \tilde{I}_{S_1} := \min_u \|(d_x, d_y)\|_1 + \frac{\mu}{2} \|u - f\|_2^2, \\ \text{sujet à } d_x = u_x, d_y = u_y. \end{cases} \quad (6.9)$$

Pour résoudre le problème (\mathcal{P}'_2) , nous résolvons le problème sans contraintes suivant :

$$(\mathcal{P}'_3) \quad \tilde{I}_{S_3} := \min_{u, d_x, d_y} \left\{ \|(d_x, d_y)\|_1 + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - u_x\|_2^2 + \frac{\lambda}{2} \|d_y - u_y\|_2^2 \right\}. \quad (6.10)$$

On peut ainsi résoudre ce dernier problème en utilisant l'algorithme de split Bregman. On définit :

$$s^k = \sqrt{|u_x^k - b_x^k|^2 + |u_y^k - b_y^k|^2}. \quad (6.11)$$

Algorithme 6.2.3 *Algorithme de split Bregman de problème débruitage TV isotrope :*

Initialisation : $k = 0, u^0 = 0, b^0 = 0$.

Tant que $\|u^k - u^{k+1}\| > \text{tol}$ *faire,*

$u^{k+1} = G^k$ *où* G *est la fonction de GaussSeidel;*

$$d_x^{k+1} = \frac{s^k \lambda (u_x^k + b_x^k)}{s^k \lambda + 1}.$$

$$d_y^{k+1} = \frac{s^k \lambda (u_y^k + b_y^k)}{s^k \lambda + 1}.$$

$$b_x^{k+1} = b_x^k + (u_x^{k+1} - d_x^{k+1}).$$

$$b_y^{k+1} = b_y^k + (u_y^{k+1} - d_y^{k+1}).$$

$k = k + 1$.

Fin tant que.

6.2.4 Combinaison des Problèmes Débruitage TV Anisotrope et Isotrope

Nous proposons une méthode de débruitage d'image en combinant des problèmes de débruitage *TV* anisotrope et isotrope.

$$(\mathcal{P}_{A+I}) \quad \min_u \left\{ \left\| \frac{\partial u}{\partial x} \right\|_1 + \left\| \frac{\partial u}{\partial y} \right\|_1 + \|\nabla u\|_1 + \frac{\mu}{2} \|u - f\|_2^2 \right\}.$$

Le problème (\mathcal{P}_{A+I}) est résolu en utilisant un problème (\mathcal{P}'_{A+I}) équivalent :

$$(\mathcal{P}'_{A+I}) \quad \begin{cases} \min_u \|d_x\|_1 + \|d_y\|_1 + \|(d_x, d_y)\|_1 + \frac{\mu}{2} \|u - f\|_2^2, \\ \text{sujet à } d_x = u_x, d_y = u_y. \end{cases}$$

Pour résoudre le problème (\mathcal{P}'_{A+I}) , nous résolvons le problème suivant sans contraintes :

$$(\mathcal{P}''_{A+I}) \quad \min_{u, d_x, d_y} \left\{ \|d_x\|_1 + \|d_y\|_1 + \|(d_x, d_y)\|_1 + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - u_x\|_2^2 + \frac{\lambda}{2} \|d_y - u_y\|_2^2 \right\}.$$

Dans nos prochaines études, nous programmons et comparons les méthodes combinant des problèmes débruitage TV anisotrope et isotrope.

6.2.5 Théorème de Convergence

Problème de débruitage TV anisotrope

D'après les différents algorithmes de split Bregman relatif au problème de débruitage TV anisotrope ([12] ; [31]), on définit les itérations suivantes :

$$\begin{aligned} d^{k+1} &= \text{prox}_{\frac{\mu}{\lambda} \|\cdot\|_1}(Bx^{k+1} + b^k) = \begin{cases} d_x^{k+1} = \text{shrink}(u_x^{k+1} + b_x^k, \frac{1}{\lambda}) \\ d_y^{k+1} = \text{shrink}(u_y^{k+1} + b_y^k, \frac{1}{\lambda}), \end{cases} \\ b^{k+1} &= b^k + (Bx^{k+1} - d^{k+1}) = \begin{cases} b_x^{k+1} = b_x^k + u_x^{k+1} - d_x^{k+1} \\ b_y^{k+1} = b_y^k + u_y^{k+1} - d_y^{k+1}, \end{cases} \\ Bx^{k+1} &= \nabla u^{k+1}, \end{aligned}$$

où B est une matrice de taille $2N^2 \times N$ définie par

$$B := \begin{bmatrix} I_N \otimes D \\ D \otimes I_N \end{bmatrix},$$

I_N est la matrice identité $N \times N$, et $P \otimes Q$ le produit de Kronecker des matrices P, Q . D est une matrice $N \times N$ définie par

$$D := \begin{bmatrix} 0 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & & -1 & 1 \end{bmatrix}.$$

Proposition 6.2.1 *Si $\lambda, \mu > 0$ et $c, x^0 \in \mathbb{R}^m$ alors, le schéma d'itération*

$$x^{k+1} = (I - \text{prox}_{\frac{\mu}{\lambda} \|\cdot\|_1})(x^k + c), \quad k = 0, 1, \dots$$

converge vers sa limite en un nombre finie d'étapes pour $i = 1, 2, \dots, m$,

$$\lim_{k \rightarrow \infty} (x^k)_i = \begin{cases} \frac{\mu \text{sign}(c_i)}{\lambda}, & c_i \neq 0, \\ (I - \text{prox}_{\frac{\mu}{\lambda} |\cdot|})(x^0)_i, & c_i = 0. \end{cases}$$

Démonstration. Pour $m = 1$. Nous divisons la démonstration en trois cas : $c > 0$, $c < 0$ et $c = 0$.

Cas $c > 0$: Dans ce cas, nous avons

$$\frac{\mu}{\lambda} = (I - \text{prox}_{\frac{\mu}{\lambda}|\cdot|})(\frac{\mu}{\lambda} + c).$$

Autrement dit, le nombre $\frac{\mu}{\lambda}$ est un point fixe du schéma itératif. Si $x^0 + c > \frac{\mu}{\lambda}$ est vrai, alors $x^1 = \frac{\mu}{\lambda}$ pour tout $k \geq 1$. En d'autres termes, une itération suffit pour atteindre la limite du schéma itératif.

Si $|x^0 + c| \leq \frac{\mu}{\lambda}$, nous avons

$$x^k = \begin{cases} x^0 + kc & \text{si } 0 \leq k < \left\lceil \frac{\mu}{\lambda c} - \frac{x^0}{c} \right\rceil, \\ \frac{\mu}{\lambda} & \text{si } k \geq \left\lceil \frac{\mu}{\lambda c} - \frac{x^0}{c} \right\rceil. \end{cases}$$

Ici, $[v]$ représente le plus petit entier qui dépasse v . Le schéma itératif atteint sa limite dans l'étape $\left\lceil \frac{\mu}{\lambda c} - \frac{x^0}{c} \right\rceil$.

Si $x^0 + c < -\frac{\mu}{\lambda}$ est vrai, alors $x^1 = -\frac{\mu}{\lambda}$. Dans l'étape $\left\lceil \frac{2\mu}{\lambda c} \right\rceil + 1$, le schéma atteint sa limite.

Cas $c < 0$: La démonstration pour le cas $c < 0$ est similaire à celle pour le cas $c > 0$. En particulier, le schéma itératif converge vers la limite $-\frac{\mu}{\lambda}$ en un nombre fini de pas.

Cas $c = 0$:

Enfin, nous considérons le cas $c = 0$. Dans ce cas, le schéma itératif devient

$$x^{k+1} = (I - \text{prox}_{\frac{\mu}{\lambda}|\cdot|})(x^k),$$

dans cet exemple.

Pour tout $k \geq 1$, on peut voir $x^k = \frac{\mu}{\lambda}$ si $x^0 \geq \frac{\mu}{\lambda}$; $x^0 \leq -\frac{\mu}{\lambda}$ si $x^0 = \frac{\mu}{\lambda}$; $x^k = x^0$ si $|x^0| \leq \frac{\mu}{\lambda}$. Par conséquent, l'itération atteint sa limite en une seule étape. \square

La convergence de l'algorithmes de split Bregman au problème débruitage TV anisotrope a été démontré dans ([49]). On peut voir qu'à chaque itération on doit résoudre un système linéaire

$$(I + \lambda B^t B)x^{k+1} = u - \lambda B^t (b^k - d^k).$$

Il a été suggéré dans [29] qu'une itération de Gauss-Seidel en un seul pas donne une approximation satisfaisante de x^{k+1} .

D'après la relation $b^{k+1} = Bx^{k+1} + b^k - d^{k+1}$ dans l'algorithme de débruitage de split Bregman par Goldstein-Osher, nous savons que $B^t Bx^{k+1} \approx B^t d^k$, quand n est suffisamment grand. Dans cette optique, si on néglige simplement le terme $\lambda B^t Bx^{k+1}$ et le terme $\lambda B^t x^k$ des côtés gauche et droit de la fonction linéaire du système linéaire

$$(I + \lambda B^t B)x^{k+1} = u - \lambda B^t (b^k - d^k),$$

ce système devient $x^{k+1} = x - \lambda B^t b^k$. Avec cette modification, l'algorithme de débruitage de split Bregman se réduit à l'algorithme de débruitage de Jia-Zhao ([51]) qui peut être considéré comme l'algorithme $FP2O-ATV$ pour le cas $k = 0$.

En outre, nous tenons à souligner que les démonstrations de convergence de l'algorithme de split Bregman de Goldstein-Osher et de l'algorithme de débruitage de Jia-Zhao utilisent largement les propriétés de la distance de Bregman.

En particulier, le paramètre λ qui garantit la convergence de l'algorithme de débruitage de Jia-Zhao doit être inférieur à $\frac{1}{8}$, alors que ce paramètre pour la convergence de l'algorithme *FP2O-ATV* est relâché à un nombre inférieur à $\frac{1}{4} \sin^{-2} \frac{(N-1)\pi}{2N}$ qui est légèrement supérieur à $\frac{1}{4}$.

Problème de débruitage TV isotrope

D'après les différents algorithmes de split Bregman au problème de débruitage *TV* isotrope ([12] ; [31]), on définit les relations suivantes :

$$d^{k+1} = \text{prox}_{\frac{1}{\lambda}\varphi}(Bx^{k+1} + b^k) = \begin{cases} d_x^{k+1} = \frac{s^k \lambda (u_x^k + b_x^k)}{s^k \lambda + 1}, \\ d_y^{k+1} = \frac{s^k \lambda (u_y^k + b_y^k)}{s^k \lambda + 1}. \end{cases}$$

$$b^{k+1} = b^k + (Bx^{k+1} - d^{k+1}) = \begin{cases} b_x^{k+1} = b_x^k + u_x^{k+1} - d_x^{k+1}, \\ b_y^{k+1} = b_y^k + u_y^{k+1} - d_y^{k+1}. \end{cases}$$

Proposition 6.2.2 *Si $\lambda, \mu > 0$ et $c, x^0 \in \mathbb{R}^m$ alors le schéma itératif*

$$x^{k+1} = (I - \text{prox}_{\frac{\mu}{\lambda}\|\cdot\|_2})(x^k + c), \quad k = 0, 1, \dots$$

converge vers sa limite en un nombre fini d'étapes pour $i = 1, 2, \dots, m$,

$$\lim_{k \rightarrow \infty} x^k = \begin{cases} x^0 - \max\left(\|x^0\|_2 - \frac{\mu}{\lambda}, 0\right) \frac{x^0}{\|x^0\|_2}, & \text{si } c = 0, \\ \frac{\mu c}{\lambda \|c\|_2}, & \text{si } c \neq 0. \end{cases}$$

Démonstration. Nous considérons deux cas :

Cas $c = 0$: Dans ce cas, l'équation se réduit à

$$x^{k+1} = (I - \text{prox}_{\frac{\mu}{\lambda}\|\cdot\|_2})(x^k).$$

Si $\|x^0\|_2 \geq \frac{\mu}{\lambda}$, nous avons $x^k = \frac{\mu x^0}{\lambda \|x^0\|_2}$ pour tout $k \geq 1$ et $x^k = x^0$ pour tout $k \geq 0$ si $\|x^0\|_2 < \frac{\mu}{\lambda}$, alors

$$\lim_{k \rightarrow \infty} x^k = x^0 - \max\left(\|x^0\|_2 - \frac{\mu}{\lambda}, 0\right) \frac{x^0}{\|x^0\|_2}.$$

Cas $c \neq 0$: Pour le cas $c \neq 0$, on a

$$x^\infty = (I - \text{prox}_{\frac{\mu}{\lambda}\|\cdot\|_2})(x^\infty + c),$$

c'est-à-dire, $c = \text{prox}_{\frac{\mu}{\lambda}\|\cdot\|_2}(x^\infty + c)$, avec x^∞ la limite des itérations de Picard. Si $\Psi = \frac{\mu}{\lambda}\|\cdot\|_2$ est une fonction convexe dans \mathbb{R}^d et $x \in \mathbb{R}^d$ alors,

$$c \in \partial_\Psi(x) \iff x = \text{prox}_\Psi(x + c);$$

donc $x^\infty = \frac{\mu c}{\lambda\|c\|_2}$. □

La convergence de l'algorithme de split Bregman au problème débruitage TV isotrope a été démontré dans ([53]). Comme dans l'algorithme de split Bregman relatif au problème de débruitage TV anisotrope, à chaque itération de l'algorithme de split Bregman relatif au problème de débruitage TV isotrope, il faut résoudre le système linéaire

$$(I + \lambda B^t B)x^{k+1} = x - \lambda B^t(b^k - d^k).$$

Une fois de plus, une étape de l'itération de Gauss-Seidel a été utilisée pour obtenir une approximation satisfaisante de x^{k+1} , (d'après [29]). Si le système linéaire est résolu à l'aide d'une étape de l'itération de Gauss-Seidel, l'analyse de convergence du schéma itératif résultant ne semble pas disponible dans la littérature.

6.3 Résultats Numériques

Nous utilisons plusieurs images pour les résultats numériques. Le bruit additif introduit est gaussien, et a tenté de récupérer l'image originale pour tester les méthodes de split Bregman dans le problème de la débruitage anisotrope et isotrope TV . Soit X les matrices qui représentent une image de taille $m \times n$. Nous avons ensuite utilisé Matlab $f = \text{imnoise}(X, 'gaussian', \text{sigma})$ commande pour définir notre image de bruit f , où sigma est une version du niveau sonore gaussien. Nous avons utilisé les valeurs $\mu = 0.1$, $\lambda = 0.2$ et la tolérance $Tol = 10^{-5}$ dans nos études.

Les résultats des algorithmes de débruitage TV anisotropiques et isotropes pour diverses images sont présentés dans les tableaux 6.1 et 6.2. La fonction $\|u - X\|_2^2$ est utilisée pour calculer les erreurs relatives. Les tableaux 6.3 et 6.4 présentent les paramètres de performance des algorithmes de débruitage TV anisotropiques et isotropes, avec $\text{sigma} = 0.08$. Les tableaux 6.5 et 6.6 donnent le SNR pour les algorithmes de débruitage TV anisotrope et isotrope pour une seule image "flower" ainsi que différentes valeurs de sigma . Le tableau 6.7 présente les valeurs du SNR pour l'image "flower" qui a été détachée à l'aide des algorithmes de Proximale-Pénalité, de débruitage TV anisotrope et débruitage TV isotrope.

Commentaires sur les résultats expérimentaux

- Comme le montre les tableaux 6.1 et 6.2, le débruitage isotrope TV est plus rapide et plus précis que la version anisotrope de la méthode de débruitage. Dans nos expériences, nous avons découvert que la suite des résidus de l'image "Iline" convergeait monotonement vers une valeur inférieure à celle de l'image bruyante originale. Cependant, en raison de la grande taille de cette image. Cela prend beaucoup de temps. Les erreurs dans les images du caméraman et de l'université ont augmenté de façon monotone et ont convergé vers un point

Images	Taille $n \times m$	Nombre d'itérations	Erreur relative	Temps (s)
cameraman	398×398	43	0.104779	45.680272
barbara	510×510	41	0.173054	85.609672
camera	340×340	7	0.165105	5.001468
flower	256×256	51	0.211968	24.340052
girl	216×233	47	0.149544	17.045064
liline	1961×3553	164	0.143347	10438.736396
university	480×640	141	0.143296	300.869216

Tableau 6.1 – Résultats obtenus par l'algorithme de débruitage TV anisotrope, sigma = 0.08.

Images	Taille $n \times m$	Nombre d'itérations	Erreur relative	Temps (s)
cameraman	398×398	23	0.104677	57.237110
barbara	510×510	137	0.17464	182.507988
camera	340×340	143	0.172107	86.802806
flower	256×256	22	0.210941	41.570893
girl	216×233	24	0.151122	18.938243
liline	1961×3553	15	0.143349	771.958677
university	480×640	24	0.143312	120.298105

Tableau 6.2 – Résultats obtenus par l'algorithme de débruitage TV isotrope, sigma = 0.08.

Images	MSE	SNR	PSNR	IQI	NK	AD	SC	MD	NAE
cameraman	7.6326e+03	15.9295	9.3041	2.8414e-05	1.1423	-44.5176	0.6582	255	0.3806
barbara	1.4142e+04	11.8896	6.6257	-3.5635e-06	1.3828	-77.6333	0.3766	234	0.8772
camera	1.1758e+04	12.4348	7.4276	-7.3644e-05	1.3689	-73.9475	0.4184	228	0.7373
flower	1.6863e+04	10.0637	5.8614	5.5853e-06	1.5642	-88.7960	0.2694	255	1.1090
girl	1.3374e+04	12.9140	6.8682	8.4450e-08	1.3312	-75.0687	0.4254	221	0.7501
liline	232.5184	12.9318	24.4662	0.0052	1.0129	-5.3346	0.9706	255	0.0268
university	1.4088e+04	13.1774	6.6423	2.3093e-04	1.1405	-61.5217	0.4907	255	0.7909

Tableau 6.3 – Mesures de performance par l'algorithme de débruitage TV anisotrope, sigma = 0.08.

Images	MSE	SNR	PSNR	IQI	NK	AD	SC	MD	NAE
cameraman	7.5933e+03	15.8979	9.3265	3.1550e-05	1.1435	-44.1626	0.6577	255	0.3825
barbara	1.3937e+04	11.8882	6.6890	-3.3960e-06	1.3831	-77.6695	0.3784	242	0.8666
camera	1.1626e+04	12.4178	7.4764	-7.3644e-05	1.3804	-76.0561	0.4157	228	0.7384
flower	1.6711e+04	10.1155	5.9007	5.1306e-06	1.5626	-88.6839	0.2706	255	1.0941
girl	1.3120e+04	12.9056	6.9516	4.4023e-07	1.3339	-75.3454	0.4267	221	0.7382
liline	210.5850	12.9318	24.8965	0.0052	1.0122	-5.0294	0.9724	255	0.0250
university	1.3858e+04	13.1735	6.7138	2.3140e-04	1.1553	-63.5094	0.4866	255	0.7864

Tableau 6.4 – Mesures de performance par l'algorithme de débruitage TV isotrope, sigma = 0.08.

d'erreur relatif supérieur à l'erreur relative de l'image bruyante initiale. Cela peut être vu dans les images et les visuels qui ont été présentés.

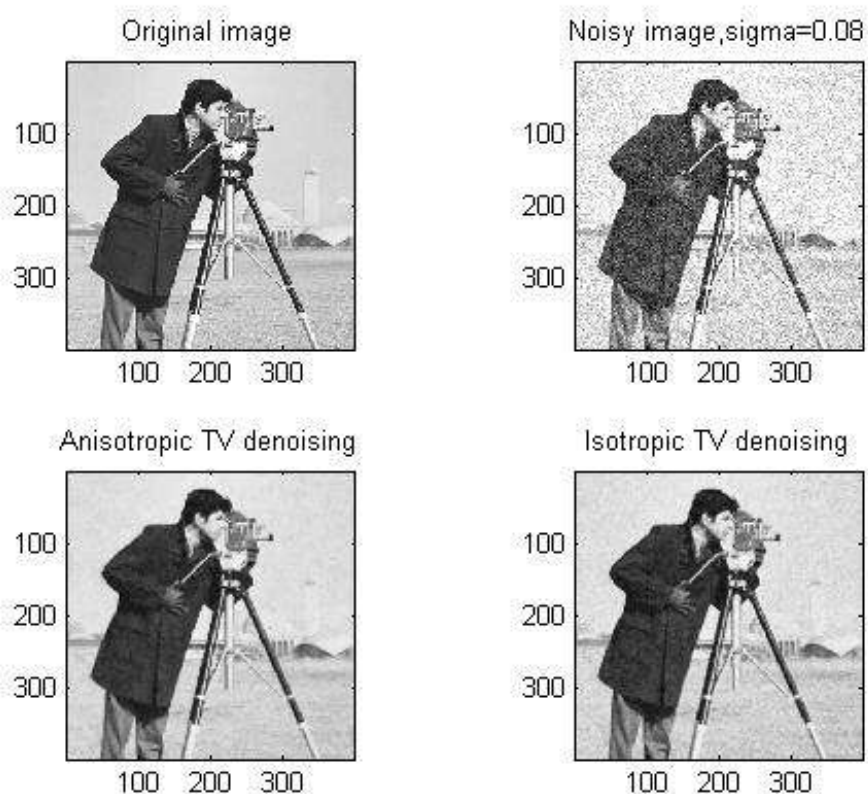


Figure 6.1 – Résultats obtenus par Split Bregman en utilisant l'image de cameraman.

- Dans les tableaux 6.3 et 6.4, nous évaluons la qualité des images restaurées par les modèles

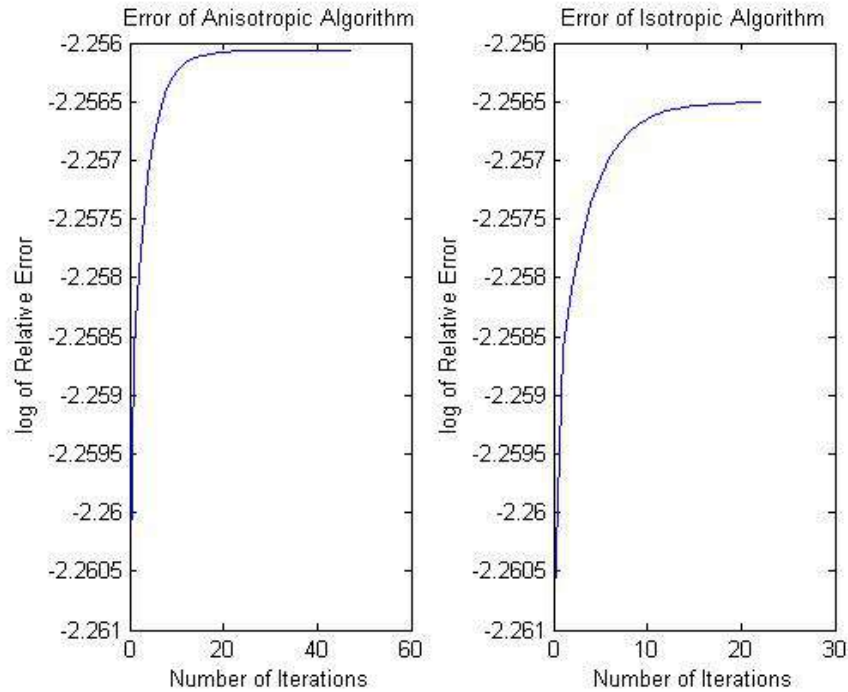


Figure 6.2 – Résultats obtenus de l’erreur de Split Bregman en utilisant l’image de cameraman.

de restauration d’image, nous utilisons l’erreur carrée (MSE), taux de bruit du signal (SNR), rapport signal de crête au bruit ($PSNR$), indice de qualité d’image (IQI), corrélation croisée normalisée (NK), différence moyenne (AD), contenu structurel (SC), différence maximale (MD), et erreur absolue normalisée (NAE).

- Pour l’image “flower” des algorithmes de débruitage TV isotrope et anisotrope et la même

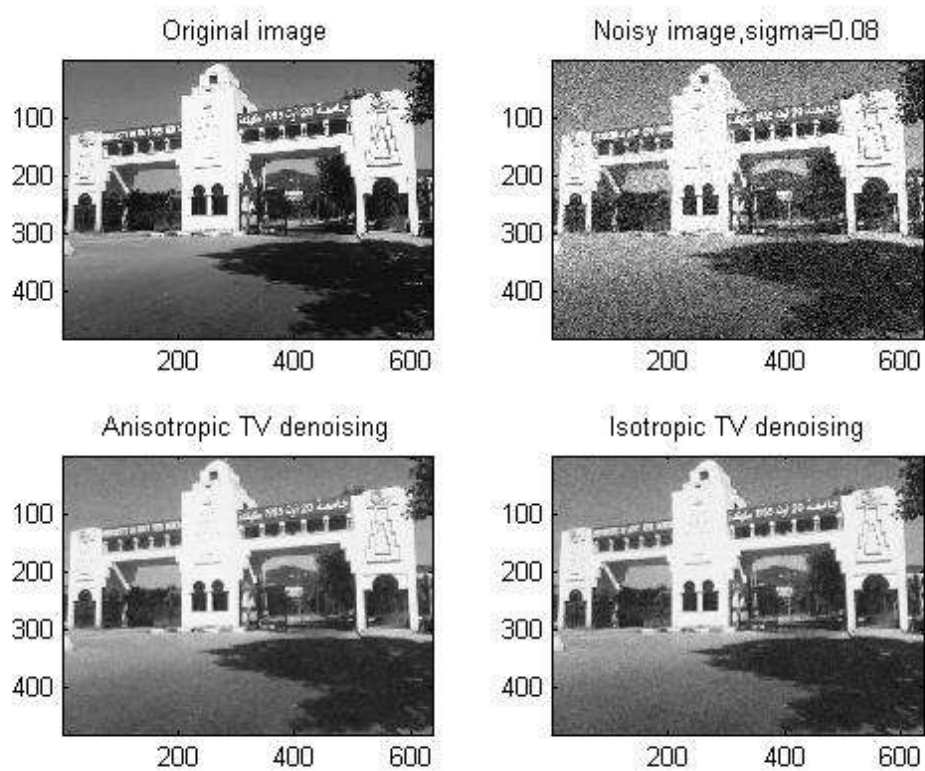


Figure 6.3 – Résultats obtenus par Split Bregman en utilisant l'image de university.

valeur de ségrégation de 0.08. Les tableaux 6.5 et 6.6 illustrent les divers résultats du SNR ,

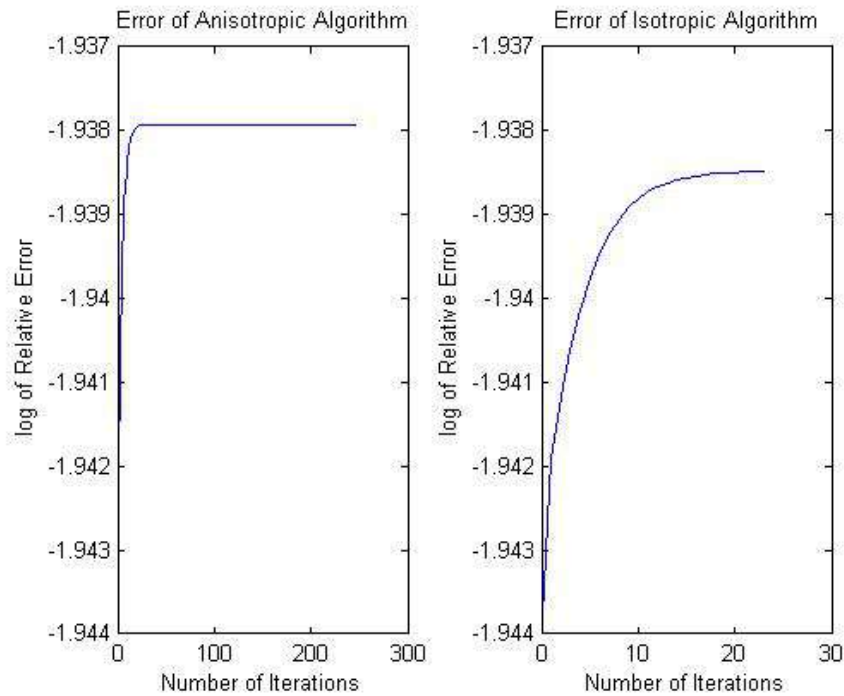


Figure 6.4 – Résultats obtenus de l’erreur de Split Bregman en utilisant l’image de university.

le nombre d’itérations, l’erreur relative et le temps.

- L’erreur relative de split Bregman, une technique itérative avec des filtres anisotropiques et isotropes, ne converge pas nécessairement de façon monotone, comme nous l’avons montré.
- La qualité des images produites avec la méthode Proximale, d’autre part, reste constante, comme la valeur du Lagrangien Augmenté-Proximal.
- Les figures (6.1 ; 6.3 ; 6.5) illustrent les résultats des méthodes de restauration telles que les

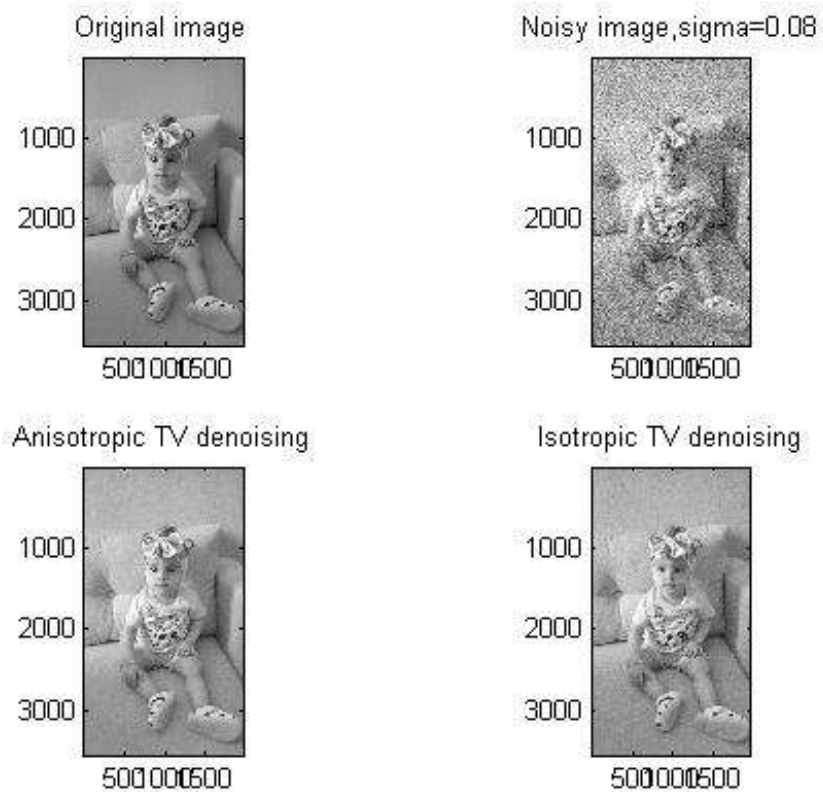


Figure 6.5 – Résultats obtenus par Split Bregman en utilisant l'image de Iline.

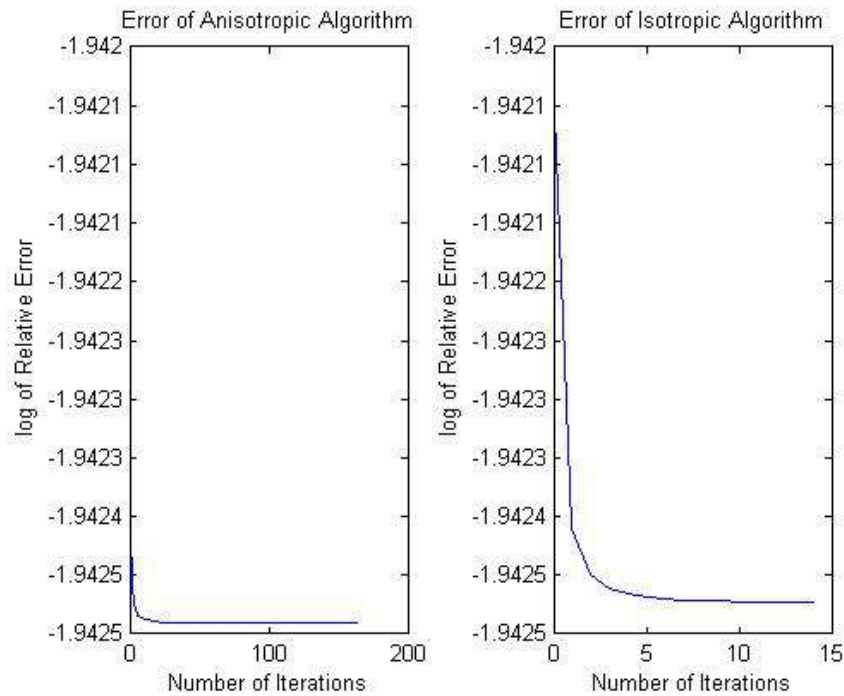


Figure 6.6 – Résultats obtenus de l’erreur de Split Bregman en utilisant l’image de Iline.

0.08.

- Les figures (6.7; 6.9) illustrent l’image originale et l’image bruyante “flower” pour un sigma égal à 0.08 et 0.501.
- Les figures (6.8; 6.10) montrent également que les algorithmes de restauration, tels que la débruitage isotrope TV et anisotrope TV , ne sont pas fiables et ont des problèmes lors du traitement de restauration. En d’autres termes, ils divergent à mesure que l’invariance

Sigma	SNR_TV_AS	Nombre d'itérations	Erreur relative	Temps (s)
0.08	10.0884	51	0.210941	73.156270
0.15	7.1100	47	0.393911	57.259335
0.25	3.6790	46	0.651452	78.888763
0.35	1.1272	68	0.905142	73.037620
0.5	-1.4576	182	1.24135	166.640882

Tableau 6.5 – Résultats obtenus par l'algorithme de débruitage TV anisotrope pour l'image flower, sigma = 0.08.

Sigma	SNR_TV_IS	Nombre d'itérations	Erreur relative	Temps (s)
0.08	10.1155	22	0.210019	41.570893
0.15	7.0932	21	0.394318	22.865811
0.25	3.6670	20	0.652357	37.383759
0.35	1.1227	19	0.905396	15.270150
0.5	-1.4507	18	1.24031	14.801319

Tableau 6.6 – Résultats obtenus par l'algorithme de débruitage TV isotrope pour l'image flower, sigma = 0.08.

Sigma	SNR Prox	SNR TV AS	SNR TV IS
0.08	21.4	10.0884	10.1155
0.15	25.1	7.1100	7.0932
0.25	20.4	3.6790	3.6670
0.35	21.9	1.1272	1.1227
0.501	21	-1.4635	-1.4733

Tableau 6.7 – Les différentes valeurs de SNR pour image débruitée par : Prox ; TV anisotrop ; TV isotrop.

du bruit blanc (sigma) augmente. Cependant, ce n'est pas le cas lors de l'utilisation des algorithmes proximaux. Cette dernière semble très ancienne et fiable. Elle fournit une variance sigma élevée pour l'image débruité dans le tableau 6.7.

- Le *SNR* des images restaurées par l'algorithme proximal est presque constant.

6.4 Conclusion

Dans ce travail, nous avons comparé les algorithmes de proximale-pénalité pour résoudre une classe de problèmes d'optimisation non différentiable avec des algorithmes *TV* anisotrope et *TV* isotrope pour résoudre des problèmes d'optimisation. Basé sur la comparaison des résultats de la restauration de différents modèles connexes, nous pouvons confirmer que l'algorithme proximal approprié pour la restauration d'image produit les meilleurs résultats de haute qualité (clair, pas lisse, et les textures sont conservées), et l'algorithme proximal convergence est garantie indépendamment des valeurs du *RSB* si nous les comparons avec

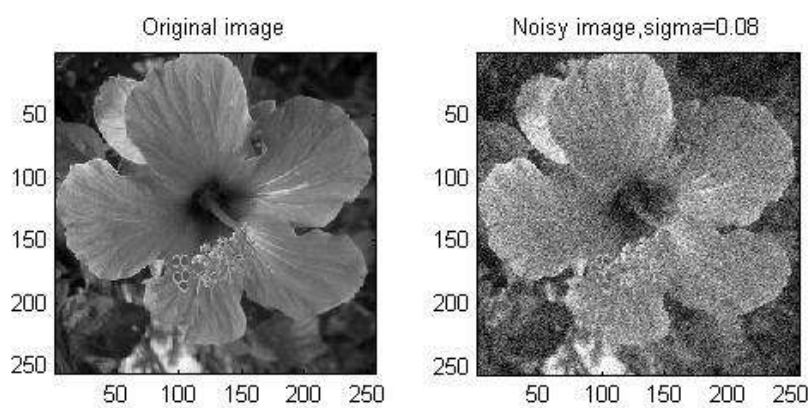


Figure 6.7 – Image originale et image débruitée pour $\sigma = 0.08$.

d'autres méthodes. Sur la base des résultats antérieurs, nous pouvons conclure que les algorithmes TV débruitage l'anisotrope et isotrope fonctionnent dans une relation de corrélation directe. En d'autres termes, indépendamment de la petite valeur de σ et que nous obtenons de meilleurs résultats de qualité d'image plus ancienne. L'approche converge monotonement avec une tolérance égale à 10^{-5} malgré la grande taille de l'image, il faut beaucoup de temps pour les calculer, et les algorithmes de débruitage TV isotrope sont plus rapide que

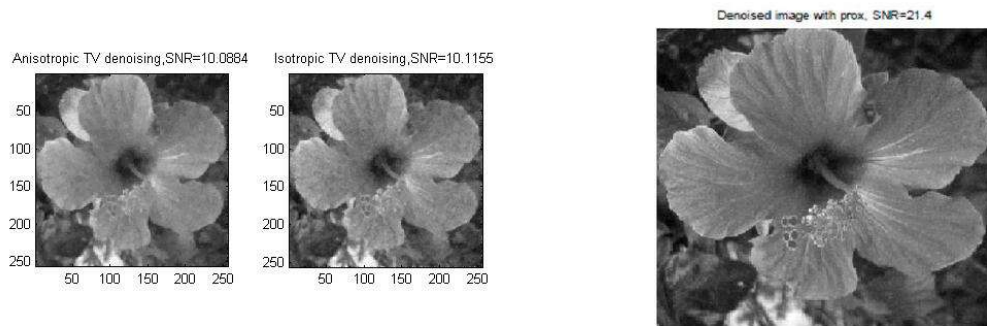


Figure 6.8 – Le SNR pour image débruitée par Prox ; TV anisotrop ; TV isotrop pour $\sigma = 0.08$.

les algorithmes l'anisotrope. Dans nos tests, nous avons découvert que l'image restaurée est plus nette et plus précise. L'algorithme Proximal donne, également, de meilleurs résultats de déconvolution que TV qui est anisotrope ou isotrope, les deux en termes de résultats visuels et les paramètres d'évaluation de la qualité de la restauration comme le $PSNR$ et le SNR .

Dans nos prochaines études, nous souhaitons combiner anisotrope et isotrope TV pour la débruitage d'image, pour programmer la méthode dans MATLAB et nous voulons comparé les méthodes de débruitage.

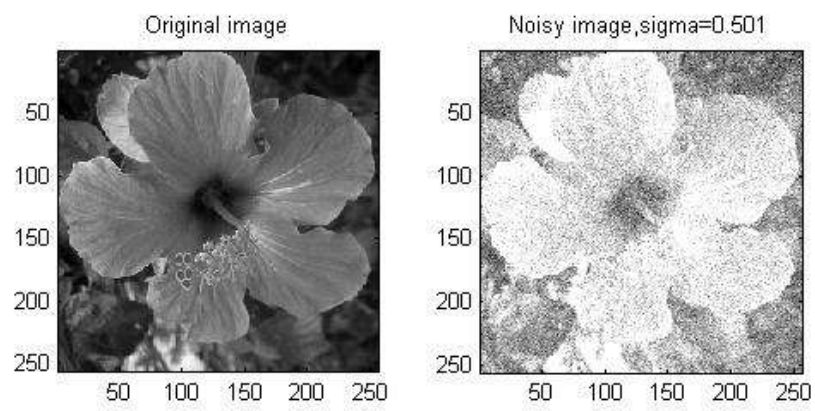


Figure 6.9 – Image originale et image débruitée pour $\sigma = 0.501$.

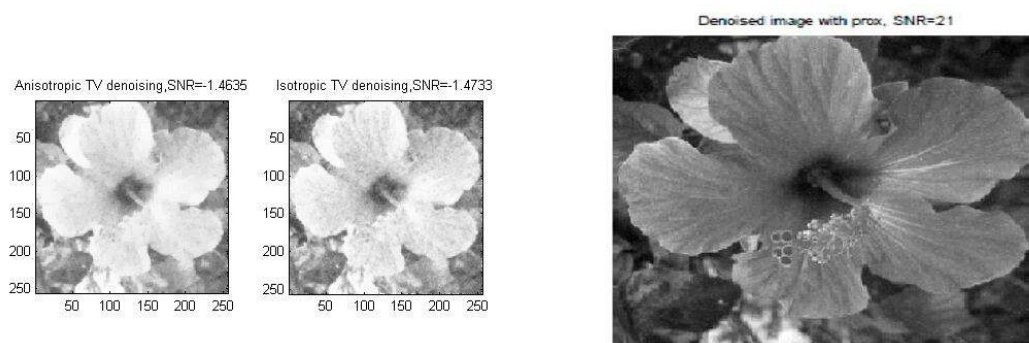


Figure 6.10 – Le SNR pour image débruitée par Prox ; TV anisotrop ; TV isotrop pour $\sigma = 0.501$.

Bibliographie

- [1] J.-F. Aujol, Traitement d'image par approches variationnelles et équations aux dérivées partielles, Semestre d'enseignement UNESCO sur le traitement des images numériques, TUNIS, ENIT du 11 au 16 Avril 2005.
- [2] A. Auslender et Al, Penalty-Proximal in convex programming, J.O.T.A., Vol. 55, No. 1, (1987), pp. 1-21.
- [3] H. H. Bauschke, Adriann S. Lewis†, Dykstra's Algorithm with Bregman Projections : A convexe Proof, Received 18 june 1998, In final from 11 March 1999.
- [4] H. H. Bauschke, Duality for Bregman projections onto translated cones and affine subspaces, Department of Mathematics and Statistics, University of Guelph, Guelph, Ont., Canada N1G2W1, Received 11 February 2002 ; revised 18 October 2002.
- [5] H. H. Bauschke, J. M. Borwein, and P. L. Combettes, Bregman monotone optimization algorithms, SIAM J. Control Optim. 42 (2003), pp. 596-636.
- [6] Maïtine Bergounioux. Quelques méthodes mathématiques pour le traitement d'image. DEA. Cours M2 - Université d'Orléans 2008 – 2009, 2008, pp.110. cel-00125868v4.
- [7] M. Bertero and P. Boccacci, Introduction to Inverse Problems in Imaging (Bristol : Institute of Physics Publishing), 1998.
- [8] Bianchiy and Donatelliz, On generalized iterated Tikhonov regularization with operator dependent seminorms. This is a preprint. The final version is published on ETNA vol. 47, pp.73 – 99, 2017.
- [9] S. Bougueroua and N. Daili, Comparative Analysis of the Prox Penalty and Bregman Algorithms for Image Denoising, Journal of Applied Mathematics, Volume 2023, Article ID 6689311, 15 pages, <https://doi.org/10.1155/2023/6689311>.
- [10] S. Bougueroua and N. Daili, Numerical Comparisons of Different Imaging Algorithms, Scientific Research Publishing, Journal of Applied Mathematics and Physics, 2023, 11, 2671-2690 ; Article ID :1723299, <https://www.scirp.org/journal/jamp>.
- [11] L. M. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, Zh. Vychisl. Mat. Mat. Fiz., Vol. 7, Number 3, 1967, 620-631.
- [12] J. Bush, Bregman Algorithms, Senior Thesis, University of California, Santa Barbara, June 10, 2011.
- [13] D. Butnariu, Y. Censor, and S. Reich, Iterative averaging of entropic projections for solving stochastic convex feasibility problems, Comput. Optim. Appl. 8 (1997), pp. 21-39.

- [14] D. Butnariu and A. N. Iusem, *Totally Convex Functions for Fixed Points Computation and Infinite Dimensional Optimization*, Kluwer Academic Publishers, Dordrecht, 2000.
- [15] C. Brune, A. Sawatzky, and M. Burger, Primal and dual Bregman methods with application to optical nanoscopy, *International Journal of Computer Vision*, 92(2)(2011), pp. 211-229.
- [16] Charles Byrne, Iterative oblique projection onto convex sets and the split feasibility problem, *Inverse Problems* 18 – (2002), pp. 441 – 453.
- [17] J.-F. Cai, S. Osher, and Z. Shen. Split Bregman methods and frame based image restoration, *Multiscale Modeling and Simulation : A SIAM Interdisciplinary Journal*, 2 : 337 – 369, 2009.
- [18] Y. Censor and A. Lent, An iterative row-action method for interval convex programming, *J. Optim. Theory Appl.* 34 (3) (1981), pp.321-353.
- [19] Y. Censor and T. Elfving, A multiprojection algorithm using Bregman projections in a product space *Numer Algorithms*, pp. 221 – 239, 1994.
- [20] A. Chambolle, An Algorithm for Total Variation Minimization and Applications, *J. Math. Imaging Vision* 20 (2004), pp. 89-97.
- [21] T. F. Chan, G. H. Golub, and P. Mulet, A nonlinear primal-primal method for total variation-based image restoration, *SIAM Journal on Scientific Computing*, 20(6)(1999), pp. 1964-1977.
- [22] Y. Chen, J. Wu, and G. Yu, Adaptive Proximal Point Algorithms for Total Variation, *Image Restoration, Stat., Optim. Inf. Comput.* Vol. 3, March 2015, pp. 15-29.
- [23] N. Daili and Kh. Saadi, Nondifferentiable Augmented Lagrangian, ε -Proximal Penalty Methods and Applications, *Malaya J. Math.* 4 (4) (2016), pp. 534-555.
- [24] N. Daili, Some Augmented Lagrangian Algorithms Applied to Convex Nondifferentiable Optimization Problems, *Journal of information & Optimization Sciences (JIOS)*, Vol. 33, No. 4&5, (2012), pp. 487-526.
- [25] N. Daili and Kh. Saadi, Epsilon-Proximal Point Algorithms for Nondifferentiable Convex Optimization Problems and Applications, *AMO - Advanced Modeling and Optimization*, Volume 14, Number 1, (2012), pp. 175-195.
- [26] P. Getreuer, Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman, *Image Processing On Line (IOL)*, 2 (2012), pp. 74-95. <https://dx.doi.org/10.5201/ipol.2012.g-tvd>.
- [27] S. Gheraibia, A. Guesmia and N. Daili, The Robustness of Proximal Penalty Algorithms in Restoration of Noisy Image, Hacettepe University, *Journal of Mathematics and Statistics*, Vol. 46, no. 6, (2017), pp. 1043-1052, *Doi : 10.15672/HJMS.2017.429*.
- [28] S. Gheraibia and N. Daili, Restoration of the Noised Images by the Proximal Penalty Algorithms, *Pac. J. Appl. Math. (PJAM)*, Vol. 7, n°. 3, (2015), pp. 149-161.
- [29] T. Goldstein and S. Osher, The split Bregman method for ℓ_1 -regularized problems, *SIAM J. Imaging Sci.*, 2(2)(2009), 323-343.
- [30] W. Hengartner, M. Lambert, C. Reischer, *Introduction à l'analyse fonctionnelle*, Les Presses de l'Université du Québec, (1981).

- [31] R.-Q. Jia and H. Zhao. A fast algorithm for the total variation model of image denoising, *Advances in Computational Mathematics*, 33 pp. 231 – 241, 2010.
- [32] R.-Q. Jia, H. Zhao, and W. Zhao, Convergence analysis of the Bregman method for the variational model of image denoising, *Applied and Computational Harmonic Analysis*, 27(2009), pp. 367-379.
- [33] S. Kabbadj, Inexact Version of Bregman Proximal Gradient Algorithm, *Hindawi Abstract and Applied Analysis*, Article *ID* 1963980, 11 pages, 2020, <https://doi.org/10.1155/2020/1963980>.
- [34] T. Kato. *Perturbation theory for linear operators*, University of California, Berkely, Berlin, Heidelberg, New york, (1980).
- [35] L. Landweber, An iterative formula for Fredholm integral equations of the first kind, *Am. J. Math* 15 – 24, 1951.
- [36] R. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images* (Dordrecht : Kluwer), 1991.
- [37] Lecheheb, Maouni and Lakhal, *Imagr Restoration Using Nonlinear Eliptic Equation*, *International Journal of Computer Science, Communication and Information Technology (CSCIT)*, Vol.6 pp.32-37, March 2022.
- [38] Chong Lia, Wen Songb and Jen-Chih Yaoc, The Bregman distance, approximate compactness and convexity of Chebyshev sets in Banach spaces, *Journal of Approximation Theory*, 162, pp. 1128 – 1149, (2010).
- [39] Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier, *Solving Inverse Problems with Deep Neural Networks*, arXiv :1803.00092v3 [math.NA] 8 Dec 2019.
- [40] M. Maouni and F.Z.Nouri, *Image Restoration based on a p -Gradient Model*, *International Journal of Applied Mathematics and Statistics*, Vol.41 ; Issue No.11, Year 2013.
- [41] Y. Meyer, (2002). *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, *University Lecture Series*, Vol. 22, Amer. Math. Soc.
- [42] Nguyen Minh Hue, Dang N. H. Thanh, Le Thi Thanh, Nguyen Ngoc Hien, V. B. Surya Prasath. *Image Denoising with Overlapping Group Sparsity and Second Order Total Variation Regularization*. 2019 *6th NAFOSTED Conference on Information and Computer Science (NICS)*.
- [43] Ch. A. Micchelli, L. Shen, and Y. Xux, *Proximity algorithms for image models : Denoising*, *Article in Inverse Problems*, March 2011, *DOI : 10.1088/0266-5611/27/4/045009*.
- [44] M. T. Nair. *Linear operator equations : approximation and regularization*. World Scientific, (2009).
- [45] M. Ngwa and E. Agyingi, *Effect of an External Medium on Tumor Growth-induced Stress*, *IAENG International Journal of Applied Mathematics (JAM)*, Vol. 42, No. 2, (2012), pp. 4-5.
- [46] S. Osher , M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, *Multiscale Model. Simul.* 4(2005), pp. 460-489.

- [47] H. Rezgui, M. Maouni, M. L. Hadji and Gh. Touil, Three Robust Edges Stopping Functions For Image Denoising, *Bol. Soc. Paran. Mat* (33) V. 2022 (40), doi :10.5299/bspm.45945.
- [48] L. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D*, 60 (1992), pp. 259-268.
- [49] F. Schopfer , A. K. Louis, and T. Schuster, Nonlinear iterative methods for linear ill-posed problems in Banach space, *Inverse Problems* 22(2006) 311-329.
- [50] S. Setzer, Operator splittings, Bregman methods and frame shrinkage in image precesing, *Inter. Jour. Compu. Vision*, 92(3)(2011), 265-280.
- [51] E. Y. Sidky et al, Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle-Pock algorithm, *Physics in Medicine and Biology*, Vol. 57, n°.10 (2012), pp. 3065-3091.
- [52] Souilah, Maouni and Slimani, The Existence of Renormalized Solution for Quasilinear Parabolic Problem with Variable Exponents and Measure Data, *Bol. Soc. Paran. Mat.* (3s.) v.2023 (41) : 127. [https://doi :10.5269/bspm.51207](https://doi.org/10.5269/bspm.51207).
- [53] Dang N. H. Thanh, V. B. Surya Prasath, Le Minh Hieu and Sergey Dvoenko, An adaptive method for image restoration based on high-order total variation and inverse gradient, Springer-Verlag, London Ltd., part of Springer Nature 2020. <https://doi.org/10.1007/s11760-020-01657-9>
- [54] Le Thi Thanh, Dang N. H. Thanh, Medical Images Denoising Method Based on Total Variation Regularization and Anscombe Transform, 2019, 19th International Symposium on Communications and Information Technologies (ISCIT).
- [55] A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems*, Winston and Sons, (1977).
- [56] L. Vese and S. J. Osher, Modeling textures with total variation minimization and oscillating patterns in image processing, UCLA C. A. M. Report 02 – 19, *Journal of Scientific Computing*, Vol. 19, Nos. 1–3, December 2003.
- [57] N. Ja. Vilenkin. *Functional analysis*, Wolters-Noordhoff Publishing, (1972).
- [58] Fermn S. Viloche Bazn, Leonardo S. Borges and Juliano B. Francisco, On a generalization of Reginskas parameter choice rule and its numerical realization in large-scale multi-parameter Tikhonov regularization, *Applied Mathematics Computation* 219 (2012) 2100 – 2113.
- [59] Di Zhang and Ting-Zhu Huang, Generalized Tikhonov regularization method for large-scale linear inverse problems, *Journal of Computational Analysis and Applications*, VOL. 15, NO.7, 1317 – 1331, *January* 2013.

Annexes

1 Programme de calcul des mesures de qualité d'image dans MATLAB

```
clc; clearall; closeall;
origImg = imread('cameramann.tif'); % Read Original and Distorted Images
distImg = imread('cameraman.tif');
noOfDim = ndims(origImg); % If the input image is rgb, convert it to gray
image
if(noOfDim == 3)
    origImg = rgb2gray(origImg);
end
noOfDim = ndims(distImg);
if(noOfDim == 3)
    distImg = rgb2gray(distImg);
end
origSiz = size(origImg); % Size Validation
distSiz = size(distImg);
sizErr = isequal(origSiz, distSiz);
if(sizErr == 0)
    disp('Error: Original Image & Distorted Image should be of same dimen-
sions');
    return;
end
MSE = MeanSquareError(origImg, distImg); % Mean Square Error
disp('Mean Square Error = ');
disp(MSE);
PSNR = PeakSignaltoNoiseRatio(origImg, distImg); % Peak Signal to Noise
Ratio
disp('Peak Signal to Noise Ratio = ');
disp(PSNR);
NK = NormalizedCrossCorrelation(origImg, distImg); % Normalized Cross-
Correlation
disp('MNormalized Cross-Correlation = ');
disp(NK);
AD = AverageDifference(origImg, distImg); % Average Difference
```

```

disp('Average Difference = ');
disp(AD);
SC = StructuralContent(origImg, distImg); % Structural Content
disp('Structural Content = ');
disp(SC);
MD = MaximumDifference(origImg, distImg); % Maximum Difference
disp('Maximum Difference = ');
disp(MD);
NAE = NormalizedAbsoluteError(origImg, distImg); % Normalized Absolute Error
disp('Normalized Absolute Error = ');
disp(NAE);

```

2 Un script MATLAB pour le modèle de Tykhonov

```

clc; clear all;
close all;
rng(0)
I = imread('imagee.png'); % Original image.
noisyI = imnoise(I,'gaussian',0.1); % noisy image. and
[M, N] = size(noisyI);
noisyPSNR = psnr(noisyI, I);
disp(['noisyPSNR = ', num2str(noisyPSNR)])
A = noisyI - I;
B = noisyI .* I;
MSE = sum(A(:) .* A(:))/numel(noisyI);
% root mean square error MSE, numel() function returns the number of
matrix elements.
SNR = 10 * log10(sum(I(:) .* I(:))/MSE/numel(noisyI)); % SNR.
PSNR = 10 * log10(255^2/MSE); % peak signal-to-noise ratio PSNR.
display(MSE); % root mean square error MSE.
display(SNR);
L1 = spdiags([ones(M, 1) - 2 * ones(M, 1)ones(M, 1)], -1 : 1, M, M);
L1(1, 1) = -1;
L1(M, M) = -1;
L2 = spdiags([ones(N, 1) - 2 * ones(N, 1)ones(N, 1)], -1 : 1, N, N);
L2(1, 1) = -1;
lambda1 = 0.08; % 1/lambda.
U0 = im2double(noisyI(:));
U = (speye(M*N) - lambda1 * (kron(speye(N), L1) + kron(L2, speye(M)))) \ U0;
disp(['||denoisedI - noisyI||^2 / (MN * sigma^2) - 1 = ', num2str(norm(U -
U0)^2 / (M * N * 0.08) - 1)]).
denoisedI = im2uint8(reshape(U, M, N));
denoisedPSNR = psnr(denoisedI, I);

```

```

disp([t denoisedPSNR = t, num2str(denoisedPSNR)]).
figure; colormapgray;
subplot(1, 2, 1); imshow(noisyI); title(tNoisyimage);
subplot(1, 2, 2); imshow(denoisedI); title(tDenoisedimagebyTyhonov).

```

3 Un script MATLAB pour le modèle ROF

```

clc; clearall;
closeall;
rng(0)
I = imread('tskikda.jpg'); % clean image
sigma2 = 0.08; % variance
noisyI = imnoise(I, 'gaussian', sigma2); % noisy image
[M, N] = size(noisyI);
noisyPSNR = psnr(noisyI, I);
disp([t noisyPSNR = t, num2str(noisyPSNR)])
A = noisyI - I;
B = noisyI .* I;
MSE = sum(A(:) .* A(:)) / numel(noisyI);
% root mean square error MSE, numel() function returns the number of
matrix elements
SNR = 10 * log10(sum(I(:) .* I(:)) / MSE / numel(noisyI)); % SNR
PSNR = 10 * log10(255^2 / MSE) % peak signal-to-noise ratio PSNR
display(MSE);
display(SNR);
b1 = spdiags([-ones(M, 1) ones(M, 1)], 0 : 1, M, M);
b1(M, M) = 0;
D1 = kron(speye(N), b1);
b2 = spdiags([-ones(N, 1) ones(N, 1)], 0 : 1, N, N);
b2(N, N) = 0;
D2 = kron(b2, speye(M));
epsln = 0.0001;
U0 = im2double(noisyI(:));
Uold = U0;
lambda1 = 0.2; % 1/lambda
tol = 10^(-3); % quantization error as a convergence tolerance
for k = 1 : 100
C = spdiags(lambda1 ./ (sqrt((D1 * Uold).^2 + (D2 * Uold).^2) + epsln), 0, M *
N, M * N);
U = (speye(M * N) + D1' * C * D1 + D2' * C * D2) \ U0;
dist = norm(U - Uold, 'inf');
disp([t k = t, num2str(k), t : t, num2str(dist)])
if dist < tol

```

```

break
end
Uold = U;
end
disp(['t||denoisedI - noisyI||^2/(MN * sigma^2) - 1 = t, ...
num2str(norm(U - U0)^2/(M * N * sigma2) - 1)])
denoisedI = im2uint8(reshape(U, M, N));
denoisedPSNR = psnr(denoisedI, I);
disp(['tdenoisedPSNR = t, num2str(denoisedPSNR)])
figure; imshowpair(I, noisyI, 'montage')
figure; colormapgray;
imagesc(denoisedI); axisimage; title('denoisedImage');

```