PolyPublie
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Réduction de bruit dans les signaux numériques via une analyse statistique locale et une estimation globale
Auteur: Author:	Grégoire Morin
Date:	2018
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Morin, G. (2018). Réduction de bruit dans les signaux numériques via une analyse statistique locale et une estimation globale [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/3278/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/3278/
Directeurs de recherche: Advisors:	Antoine Saucier
Programme: Program:	Maîtrise recherche en mathématiques appliquées

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

RÉDUCTION DE BRUIT DANS LES SIGNAUX NUMÉRIQUES VIA UNE ANALYSE STATISTIQUE LOCALE ET UNE ESTIMATION GLOBALE

GRÉGOIRE MORIN
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
AOÛT 2018

© Grégoire Morin, 2018.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

RÉDUCTION DE BRUIT DANS LES SIGNAUX NUMÉRIQUES VIA UNE ANALYSE STATISTIQUE LOCALE ET UNE ESTIMATION GLOBALE

présenté par : MORIN Grégoire
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. AUDET Charles, Ph. D., président
M. SAUCIER Antoine, Ph. D., membre et directeur de recherche
M. ADJENGUE Luc, Ph. D., membre

REMERCIEMENTS

Je tiens à remercier en premier lieu mon directeur de recherche, Antoine Saucier, pour m'avoir fait découvrir le monde de la recherche et m'avoir financé lors de cette maîtrise.

Je remercie également M. Charles Audet et M. Luc Adjengue, respectivement président et membre de mon jury, pour avoir pris le temps de lire attentivement mon mémoire et d'y proposer corrections et améliorations.

Un merci tout spécial à mes parents, les conversations Skype du dimanche soir et les visites dans le bas du Fleuve m'ont toujours apporté lumière, réconfort et délicieuses confitures.

Merci à Jonathan et Farooq pour leur relecture, ce fut grandement apprécié.

Merci à Jorge, coloc et ami, qui me fait découvrir petit à petit les secrets de la sauce piquante et me laisse gagner à Catane.

Enfin, merci à mes deux frères et à tous mes amis qui m'ont toujours supporté, chacun à sa manière, lors de cette maîtrise.

RÉSUMÉ

On encode aujourd'hui une grande quantité d'information de manière numérique. On appelle signal numérique une séquence ordonnée de valeurs discrètes représentant le plupart du temps un échantillonnage d'un signal naturel, un son ou une image par exemple.

Il est fréquent que ces signaux soient dégradés par de l'information parasite qu'on appelle du bruit. Ce bruit provient généralement de l'incertitude intrinsèque aux appareils récoltant les données, par exemple un appareil photo ou un enregistreur audio, d'erreurs commises lors de la compression, la transmission ou la réception des signaux ou encore d'interférences naturelles lors de la prise des données. On a alors parfois recours à des algorithmes de débruitage de signaux numériques afin de réduire ce bruit dans les signaux.

Une des difficultés pour les algorithmes de débruitage est de réduire le bruit numérique sans toutefois altérer le signal d'origine.

On présentera dans ce mémoire un algorithme de débruitage de signaux numériques ayant pour objectif de réduire le bruit dans les signaux tout en limitant la perte de l'information présente dans le signal d'origine. Pour atteindre cet objectif, notre algorithme procèdera en premier lieu à une analyse statistique locale du signal bruité. Au moyen de tests statistiques simples, on construira lors de cette étape un modèle du signal contenant de l'information relative à ses caractéristiques locales. Par exemple, on voudra déterminer les endroits où le signal est lisse et les endroits où il est discontinu. Une fois l'information relative aux caractéristiques locales du signal stockée, ou utilisera celle-ci afin de définir un problème d'optimisation défini sur tout le signal. C'est la solution à ce problème d'optimisation qui constituera l'estimé du signal de notre algorithme. L'approche de débruitage proposée se distingue par l'analyse du signal bruité faite *avant* d'entamer la procédure d'estimation, afin de diriger cette dernière.

Deux hypothèses seront faites sur les signaux à débruiter pour le développement de notre algorithme. On fera premièrement l'hypothèse que les signaux non bruités sont des signaux lisses par morceaux. De plus, on fera l'hypothèse que ces signaux lisses par morceaux sont dégradés par un bruit additif gaussien de moyenne nulle et de variance σ^2 connue.

Suite à la présentation de l'algorithme de débruitage, on comparera les résultats de celui-ci avec d'autres méthodes de débruitage. Dans certains cas, on pourra montrer que notre algorithme obtient des résultats compétitifs avec des méthodes classiques de débruitage de la littérature.

ABSTRACT

Information is nowadays frequently encoded digitally. A digital signal is an ordered sequence of discrete values that represent most of the time a sampled natural signal such as a sound or an image.

These signals rarely come clean. Indeed, they are often degraded by what we call noise. This noise usually comes from errors made by the devices that collect data, such as cameras or audio recorders, or errors made in the compression, transmission or reception of the signals. Also, interfering natural phenomena can degrade the signals of interest. It can be necessary to use denoising algorithms to reduce this noise in the digital signals.

One of the challenges of the denoising algorithms is to attenuate noise in a noisy signal without altering the original signal.

We introduce in this work a denoising algorithm that aim at both reduce noise and limit the loss of relevant information in the original signal. To do so, our algorithm will first perform a local statistical analysis of the noisy signal. This analysis will allow the construction of a model of the pure signal that contains information about its local features, for instance its smooth parts and its discontinuous parts. Once this model is constructed, we define an optimization problem that takes into account information about all the local features extracted during the statistical analysis. The solution of this optimization problem will be our signal estimate. The originality of this approach lies in the statistical analysis that we make on the noisy signal *before* launching the estimation process, in order to lead this process.

We make two hypotheses in this work on the signals to be denoised. We first make the hypothesis that pure signals are piecewise smooth. Also, we consider that these piecewise smooth signals are degraded by a gaussian additive noise that has a zero mean and a known variance σ^2.

After the presentation of our denoising algorithm, we compare its results with other denoising techniques in the literature. In some cases, we will show that our algorithm is competitive with well known denoising algorithms.

TABLE DES MATIÈRES

REMERCIEMENTS iii

RÉSUMÉ iv

ABSTRACT v

TABLE DES MATIÈRES vi

LISTE DES TABLEAUX ix

LISTE DES FIGURES x

LISTE DES ABRÉVIATIONS ET NOTATIONS xii

LISTE DES ANNEXES xiii

CHAPITRE 1 INTRODUCTION 1

1.1 Description mathématique du problème 2

1.2 Algorithmes de débruitage et approche proposée 4

1.3 Plan du mémoire 5

CHAPITRE 2 REVUE DE LITTÉRATURE 6

2.1 Filtres locaux 6

2.2 Filtres non locaux 8

2.3 Filtres fréquentiels 9

2.4 Transformation en bases d'ondelettes 9

2.5 Régularisation 10

2.6 Apprentissage statistique 12

2.7 Méthodes mixtes 12

CHAPITRE 3 PRÉSENTATION DE L'ALGORITHME DE DÉBRUITAGE POUR DES SIGNAUX 1-D 13

3.1 Analyse statistique préliminaire 14

3.1.1 Vecteurs orthogonaux selon les caractéristiques locales d'un signal . . 14

3.1.2 Tests locaux 18

3.1.3 Écriture du modèle composite 24

3.1.4 Lien entre la matrice D et le signal s 27
3.2 Produire un estimé à partir des résultats de l'analyse statistique préliminaire 29
3.3 Résolution analytique du problème de minimisation 32
3.3.1 Conditions nécessaires 32
3.3.2 Remarques sur la matrice A 33
3.3.3 Résolution des conditions de KKT 35
3.3.4 Retour aux exemples 40
3.3.5 Modifications mineures de l'algorithme 45

CHAPITRE 4 GÉNÉRALISATION DE L'ALGORITHME DE DÉBRUITAGE POUR DES SIGNAUX 2-D 52
4.1 Présentation de la méthode 52
4.2 Considérations numériques et algorithme de balayage 59

CHAPITRE 5 RÉSULTATS POUR DES SIGNAUX 1-D 62
5.1 Signaux constants par morceaux 63
5.1.1 Choix des paramètres pour l'algorithme proposé 63
5.1.2 Algorithmes comparés 64
5.1.3 Résumé des paramètres 67
5.1.4 Résultats pour un signal test 67
5.1.5 Résultats pour des signaux aléatoires 74
5.2 Signaux linéaires par morceaux 77
5.2.1 Choix des paramètres pour l'algorithme proposé 77
5.2.2 Algorithmes comparés 77
5.2.3 Résumé des paramètres 78
5.2.4 Résultats pour un signal test 79
5.2.5 Résultats pour des signaux aléatoires 81
5.3 Signaux lisses par morceaux 82
5.3.1 Choix des paramètres 82
5.3.2 Algorithmes comparés 83
5.3.3 Résultats 83
5.3.4 Remarque sur la pondération des contraintes 90
5.4 Discussion sur les résultats de la version 1-D de l'algorithme 91

CHAPITRE 6 RÉSULTATS POUR DES SIGNAUX 2-D 93
6.1 Image synthétique 93
6.2 Images naturelles 95

6.2.1 Choix des paramètres 95
6.2.2 Résultats 95

CHAPITRE 7 CONCLUSION 105
7.1 Synthèse des travaux 105
7.2 Améliorations futures 106

RÉFÉRENCES 108

ANNEXES 112

LISTE DES TABLEAUX

Tableau 1.1 Symboles utilisés dans le modèle mathématique. 5
Tableau 3.1 Vecteurs orthogonaux selon les caractéristiques locales d'un signal. . . . 17
Tableau 3.2 Tests d'orthogonalité, exemple 1. 23
Tableau 3.3 Tests d'orthogonalité, exemple 2. 24
Tableau 3.4 Tests d'orthogonalité, exemple 3. 43
Tableau 3.5 Tests d'orthogonalité avec v_1. 46
Tableau 3.6 Tests d'orthogonalité avec v_1, v_2 et v_3. 47
Tableau 3.7 Tests d'orthogonalité avec v_1, v_2 et v_3, deuxième réalisation. 49
Tableau 3.8 Modification des résultats de l'analyse statistique préliminaire. 50
Tableau 5.1 Paramètres utilisés dans les algorithmes testés pour des signaux constants par morceaux. 67
Tableau 5.2 MSE moyen sur 100 réalisations du bruit pour le signal "Blocks" [1]. . . . 70
Tableau 5.3 Temps de calcul moyen sur 100 réalisations du signal "Blocks", $\sigma = 0.3$. 74
Tableau 5.4 Paramètres utilisés dans les algorithmes testés pour des signaux linéaires par morceaux. 78
Tableau 5.5 MSE moyen sur 100 réalisations du bruit pour le signal "Ramp". 80
Tableau 5.6 Résultats des tests locaux pour une partie du signal "Piece-Regular". . . 87
Tableau 5.7 MSE moyen sur 100 réalisations du bruit pour les signaux "Piece-Regular" et "Piece-Polynomial". 89
Tableau 6.1 MSE des algorithmes de débruitage pour les images tests bruitées, $\sigma = 20$. 96

LISTE DES FIGURES

Figure 1.1 Signaux synthétiques 1-D. 3
Figure 1.2 Signal 2-D. 3
Figure 1.3 Atténuation des variations abruptes d'un signal. 4
Figure 3.1 Exemple d'échantillonnage d'une fonction lisse par morceaux. 13
Figure 3.2 Tests d'orthogonalité, exemple 1 [2]. 22
Figure 3.3 Tests d'orthogonalité, exemple 2. 23
Figure 3.4 Signal estimé, exemple 1. 41
Figure 3.5 Coefficients de s et y dans la base $\mathscr{B}$, exemple 1. 41
Figure 3.6 Signal estimé, exemple 2. 42
Figure 3.7 Coefficients de s et y dans la base $\mathscr{B}$, exemple 2. 43
Figure 3.8 Signal estimé $\hat{s}$, exemple 3 44
Figure 3.9 Coefficients de s et y dans la base $\mathscr{B}$, exemple 3. 45
Figure 3.10 Signal constant pur et bruité. 46
Figure 3.11 Faiblesse et amélioration de l'algorithme. 48
Figure 3.12 Signal bruité y, $\sigma = 0.3$, deuxième réalisation. 48
Figure 3.13 Comparaison des résultats avec et sans le traitement de l'analyse statistique préliminaire. 50
Figure 4.1 Déroulement d'une image. 52
Figure 4.2 Analyse des lignes de l'image. 53
Figure 4.3 Analyse des colonnes de l'image. 55
Figure 4.4 Analyse des diagonales de l'image. 57
Figure 4.5 Découpage d'une image en fenêtres $m \times m$ se chevauchant en leur milieu. 59
Figure 4.6 Algorithme de balayage. 61
Figure 5.1 Signal constant par morceaux "Blocks" pur et bruité. 68
Figure 5.2 Comparaison des algorithmes sur le signal "Blocks". 69
Figure 5.3 Comparaison des algorithmes sur le signal "Blocks" pour un bruit ajouté de variance plus grande. 71
Figure 5.4 Résultats de la version oracle de notre algorithme sur le signal "Blocks". 72
Figure 5.5 Différences de la solution pour les cas $\lambda = 0$ et $\lambda < 0$, signal "Blocks" 16 dB. 73
Figure 5.6 Exemple d'un signal constant par morceaux généré aléatoirement. . . . 75
Figure 5.7 Comparaison des résultats pour des signaux constants par morceaux générés aléatoirement [3]. 75

Figure 5.8 Comparaison des résultats avec la version oracle de l'algorithme pour des signaux constants par morceaux générés aléatoirement. 76
Figure 5.9 Signal linéaire par morceaux "Ramp" pur et bruité. 79
Figure 5.10 Comparaison des algorithmes sur le signal "Ramp". 80
Figure 5.11 Exemple d'un signal linéaire par morceaux généré aléatoirement. 81
Figure 5.12 Comparaison des résultats avec les algorithmes de débruitage par ondelettes pour des signaux linéaires par morceaux générés aléatoirement. . 82
Figure 5.13 Signaux lisses par morceaux "Piece-Regular" et "Piece-Polynomial" purs et bruités. 84
Figure 5.14 Comparaison des algorithmes sur les signaux "Piece-Regular" et "Piece-Polynomial". 85
Figure 5.15 Résultats de la version oracle de notre algorithme sur les signaux lisses par morceaux. 86
Figure 5.16 Exemple d'une discontinuité arrondie par l'algorithme proposé. 86
Figure 5.17 Signal "Piece-Regular" pur et bruité entre les indices 140 et 150. 87
Figure 5.18 Résultats de l'algorithme proposé sur une partie du signal "Piece-Regular". 88
Figure 5.19 Résultat de l'algorithme proposé avec v_1 et v_2. 89
Figure 5.20 Signal ayant une partie constante, une partie linéaire et une partie quadratique. 90
Figure 5.21 Exemple de l'effet de la pondération des contraintes sur le MSE de notre méthode. 91
Figure 6.1 Image constante par morceaux pure et bruitée. 94
Figure 6.2 Résultats de l'algorithme pour une image constante par morceaux. . . . 94
Figure 6.3 Images tests non bruitées. 96
Figure 6.4 Résultats de l'algorithme proposé pour l'image Lena. 98
Figure 6.5 Résultats de l'algorithme proposé pour l'image Barbara. 99
Figure 6.6 Résultats de l'algorithme proposé pour l'image Boats. 100
Figure 6.7 Résultats de l'algorithme proposé pour l'image House. 101
Figure 6.8 Résultats de l'algorithme proposé pour l'image Peppers. 102
Figure 6.9 Première ligne de l'image Lena. 103
Figure B.1 Exemples des signaux "Blocks" et "Ramp" bruités. 117
Figure B.2 Exemples des signaux "Piece-Regular" et "Piece-Polynomial" bruités. . . 118

LISTE DES ABRÉVIATIONS ET NOTATIONS

Abréviations

i.i.d.	indépendantes et identiquement distribuées
MSE	Mean Squared Error
SNR	Signal to Noise Ratio
VT	Variation totale
TIWT	Translation Invariant Wavelet Thresholding
FB	Filtre Bilatéral
NLM	Nonlocal Means
BM3D	Block Matching 3-D

Notations

v_i	$i^{\text{ième}}$ composante d'un vecteur $v \in \mathbb{R}^n$.
$v_{i,j}$	Composante de la ligne i et de la colonne j d'une matrice $v \in \mathbb{R}^{n \times n}$.
$\|.\|$	Norme ℓ^1. Pour $x \in \mathbb{R}^n$, $\|x\| = \sum_{i=1}^n \|x_i\|$.
$\|.\|$	Norme ℓ^2. Pour $x \in \mathbb{R}^n$, $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$.
I	Matrice identité, la taille de celle-ci sera claire selon le contexte.
$\nabla f(x)$	Gradient d'une fonction $f : \mathbb{R}^n \to \mathbb{R}$. $\nabla f(x) = (\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n})^T$.
$\mathcal{C}^n$	Une fonction f est un élément de $\mathcal{C}^n$ si elle est continue et ses dérivées d'ordre 1 à n sont également des fonctions continues.
log	Logarithme naturel.
$\log_{10}$	Logarithme en base 10.
$\mathbb{E}[.]$	Espérance mathématique.
Var[.]	Variance mathématique.

LISTE DES ANNEXES

ANNEXE A CODES 112
ANNEXE B SIGNAUX 1-D 117

CHAPITRE 1 INTRODUCTION

Depuis les 30 dernières années, on assiste à une croissance exponentielle du nombre de données numériques stockées par diverses instances dans le monde. En octobre 2017, le centre du calcul du CERN (Conseil Européen pour la Recherche Nucléaire) battait son propre record en stockant 12.3 pétaoctets de données, un pétaoctet représentant à peu près la capacité de stockage de 15 000 téléphones intelligents de 64 Go (Jarlett, 2017).

L'analyse de ces mégadonnées (*Big Data*) permet des avancées importantes en science, mais aussi dans d'autres domaines tels l'économie ou les télécommunications.

Cependant, cette analyse ne peut pas toujours être faite directement sur les données numériques brutes. En effet, il est courant lors du processus d'acquisition des données, de leur transmission ou de leur réception, que celles-ci soient dégradées par de l'information parasite qu'on appelle du bruit. Ce bruit peut parfois rendre difficile, voire impossible, l'analyse et l'interprétation des signaux numériques sous-jacents. On peut alors avoir recours à des algorithmes de débruitage de signaux numériques afin de diminuer l'importance du bruit dans ces signaux. On présentera dans ce mémoire un algorithme de débruitage de signaux numériques basé sur une analyse statistique locale des signaux.

On portera d'abord une attention dans ce mémoire au débruitage de signaux à une dimension (1-D). La réduction du bruit pour des signaux 1-D trouve des applications dans plusieurs domaines. Little et Jones (2011) mentionnent entre autres des applications en géophysique, en biochimie et en astrophysique. Dans le domaine médical, Lin et al. (2014) proposent une méthode de débruitage pour l'analyse d'électrocardiogrammes.

Une partie importante du domaine du débruitage des signaux numériques est le débruitage d'images, c'est-à-dire de signaux à deux dimensions (2-D). On généralisera dans ce mémoire l'algorithme développé pour des signaux 1-D à des signaux 2-D. Les images numériques sont utilisées dans une panoplie de domaines et il peut être utile de réduire le bruit présent dans celles-ci. Une application d'un algorithme de débruitage d'images numériques est présentée par Zhao et al. (2015) pour des images radars. Dans le domaine médical, Whang et Zhou (2006) introduisent un algorithme pour le débruitage d'images à résonance magnétique. De manière plus générale, l'augmentation du nombre de pixels par unité de surface dans les appareils photo numériques augmente la sensibilité de ceux-ci au bruit. Les compagnies fabriquant ces appareils sont donc intéressées par les avancées dans le domaine du débruitage d'images (Chatterjee et Milanfar, 2010).

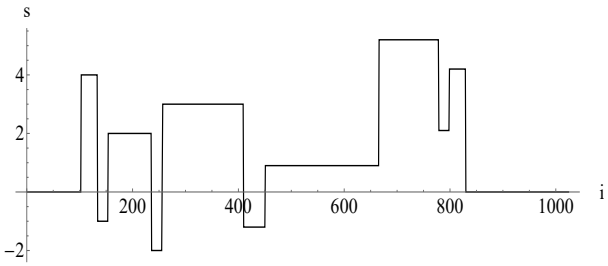
1.1 Description mathématique du problème

On considère dans ce mémoire des observations bruitées provenant d'un signal numérique, appelé signal pur, ou non bruité. On note par s le signal non bruité. Ce signal est représenté par une série temporelle, c'est-à-dire un ensemble de valeurs ordonnées $s = (s_1, ..., s_n)^T \in \mathbb{R}^n$, n représente donc le nombre total d'observations. On note $\mathcal{I}_n := \{1, ..., n\}$ l'ensemble des indices des observations d'un signal. On fait l'hypothèse que le signal pur s est dégradé par un bruit blanc additif, $w \in \mathbb{R}^n$. Les composantes de ce vecteur $w = (w_1, ..., w_n)^T$ sont des variables aléatoires indépendantes et identiquement distribuées (i.i.d.) suivant une loi normale de moyenne nulle et de variance connue σ^2. On note $y \in \mathbb{R}^n$ le signal bruité, résultat de la dégradation additive de s par w. On obtient le modèle mathématique suivant :

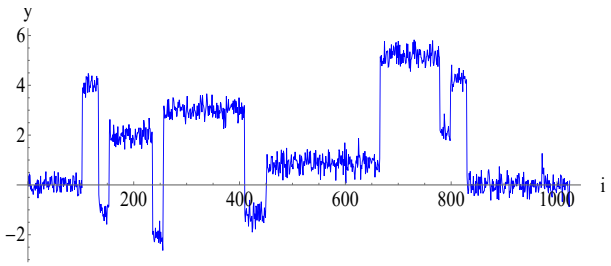
$$y = s + w, \quad \text{où } w_i \sim \mathcal{N}(0, \sigma^2). \tag{1.1}$$

Ce modèle mathématique, avec l'hypothèse d'une dégradation sous forme d'un bruit blanc additif, est largement utilisé dans la littérature et modélise bien la dégradation réelle de signaux numériques dans certaines situations (Buades et al., 2005). Dans le cas où les signaux d'intérêts sont des images, on peut faire exactement la même modélisation en considérant cette fois s, y et w comme étant des matrices dans $\mathbb{R}^{l \times h}$, où l et h représentent le nombre de pixels de l'image en largeur et en hauteur, respectivement. Le nombre total d'observations dans ces signaux est alors le nombre de pixels de l'image, $n = l \times h$.

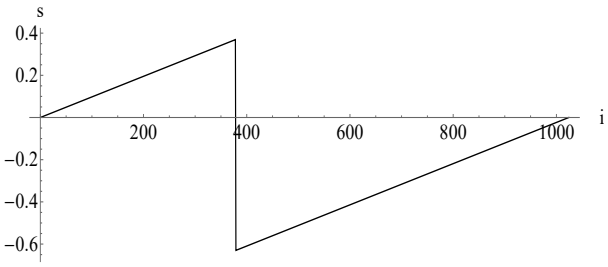
Des exemples de signaux 1-D et 2-D dégradés par un bruit blanc additif sont présentés dans les prochaines figures. Dans la figure 1.1, on présente un exemple d'un signal synthétique 1-D constant par morceaux dans sa version pure (1.1a) et bruitée (1.1b), de même qu'un exemple d'un signal linéaire par morceaux pur (1.1c) et bruité (1.1d). Dans la figure 1.2, on présente cette fois une image dans sa version pure (1.2a) et bruitée (1.2b). À noter que pour une image en niveaux de gris comme dans la figure 1.2, les observations correspondent à l'intensité de chacun des pixels de l'image. Ces pixels prennent des valeurs entières entre 0 et 255, les pixels noirs ont une valeur de 0, les blancs de 255.



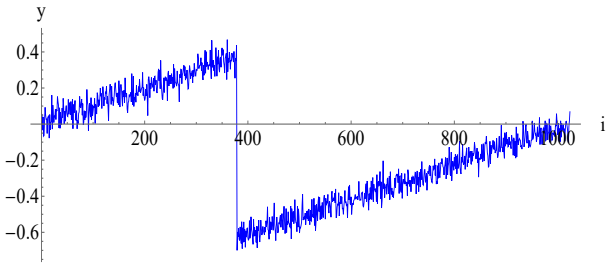
(a) Signal constant par morceaux $s \in \mathbb{R}^{1024}$.



(b) Signal bruité y, $\sigma = 0.3$.



(c) Signal linéaire par morceaux $s \in \mathbb{R}^{1024}$.



(d) Signal bruité y, $\sigma = 0.05$.

Figure 1.1 Signaux synthétiques 1-D.



(a) Image pure $s \in \mathbb{R}^{512\times512}$.



(b) Image bruitée y, $\sigma = 20$.

Figure 1.2 Signal 2-D.

1.2 Algorithmes de débruitage et approche proposée

La tâche des algorithmes de débruitage est d'estimer le signal pur s à partir de l'information bruitée y. Ces algorithmes prennent donc en entrée y et redonne en sortie un estimé de s, noté $\hat{s}$. De manière informelle, pour un estimé $\hat{s}$ produit par un algorithme de débruitage donné, on voudrait éventuellement que $\hat{s} \approx s$. On précisera davantage la façon de juger de l'efficacité d'un algorithme de débruitage dans les chapitres 5 et 6.

Une difficulté pour les algorithmes de débruitage est de réduire le bruit sans toutefois atténuer les variations abruptes du signal pur, qu'on appellera discontinuités. Par exemple, pour le signal bruité 1-D constant par morceaux de la figure 1.1b, on voudrait éventuellement réduire le bruit sur chacun des paliers du signal mais garder les transitions abruptes entre ceux-ci. De même pour l'image bruitée 1.2b, on voudrait réduire le bruit aux endroits où la valeur des pixels varie peu, par exemple sur le mur du fond, tout en gardant nets les endroits où il y a des transitions abruptes, soient les contours de l'image. On présente dans la figure 1.3 le signal estimé $\hat{s}$ produit par un filtre appelé filtre gaussien à partir du signal bruité y de la figure 1.1b. On décrira ce filtre dans le chapitre 2.

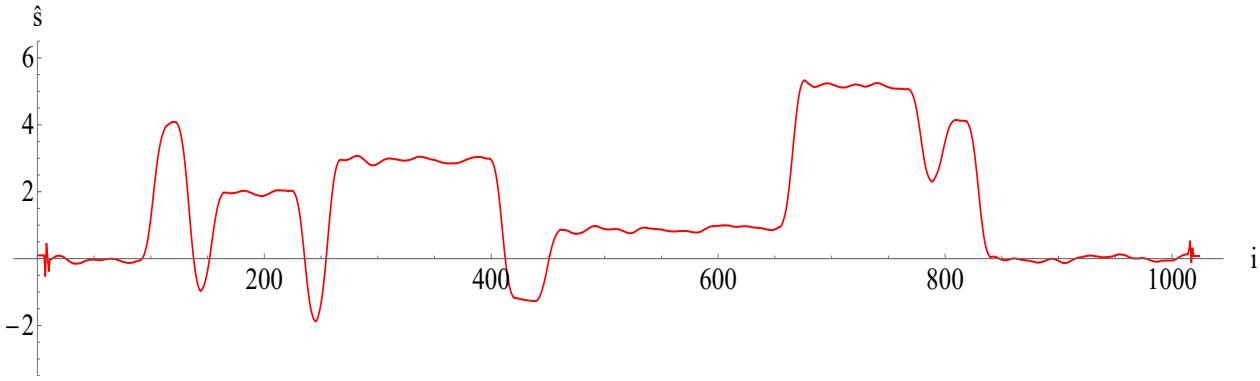


Figure 1.3 Atténuation des variations abruptes d'un signal.

On constate dans la figure 1.3 que le filtre gaussien a permis de diminuer le bruit présent dans le signal de la figure 1.1b mais en se faisant a arrondi les transitions entre chaque palier du signal pur de la figure 1.1a.

L'approche de débruitage qu'on propose dans ce mémoire est d'inspecter le signal bruité y à petite échelle afin d'extraire une multitude de modèles locaux décrivant le signal s à ces endroits. Cette approche se base sur le fait que même si un signal s n'est pas lisse globalement, il peut l'être localement. On cherchera donc à caractériser chacune de ces parties lisses. Tous les modèles locaux seront rassemblés dans un modèle appelé modèle *composite* de s. Ce modèle comportera de l'information locale relative aux endroits lisses et aux endroits

discontinus du signal. Le modèle sera intégré à un problème d'optimisation défini sur tout le signal. C'est la solution de ce problème d'optimisation qui constituera l'estimé $\hat{s}$ de notre méthode. L'avantage de cette approche est de préserver les caractéristiques locales d'un signal s donné, par exemple ses discontinuités, lors du processus de débruitage.

1.3 Plan du mémoire

Le mémoire s'organisera comme suit. On commence, dans le chapitre 2, par une revue de la littérature sur les différentes familles d'algorithmes de débruitage de signaux numériques 1-D ou 2-D. Dans le chapitre 3, on présente en détail le nouvel algorithme proposé dans ce mémoire, dans sa version 1-D. On généralise ensuite dans le chapitre 4 l'algorithme pour des images, en appliquant la version 1-D développée dans le chapitre 3 aux lignes, colonnes et diagonales des images. Dans les chapitres 5 et 6, des résultats de l'algorithme proposé sont présentés, et ceux-ci sont comparés avec d'autres algorithmes de débruitage dans la littérature. Enfin, une conclusion est apportée dans le chapitre 7, entre autres afin de discuter des forces et des faiblesses du nouvel algorithme proposé ainsi que des possibles avenues quant à l'amélioration de celui-ci.

On présente dans le tableau 1.1 la notation qui sera utilisée tout au long du travail.

Tableau 1.1 Symboles utilisés dans le modèle mathématique.

Symbole	Définition
$n \in \mathbb{N}^+$	Nombre d'observations du signal
$\mathcal{I}_n$	Ensemble $\{1, ..., n\}$ des indices du signal
$s \in \mathbb{R}^n$	Signal pur
$y \in \mathbb{R}^n$	Signal bruité
$w \in \mathbb{R}^n$	Bruit additif
$\sigma^2 \in \mathbb{R}$	Variance du bruit additif
$\hat{s} \in \mathbb{R}^n$	Estimé du signal s

CHAPITRE 2 REVUE DE LITTÉRATURE

Il existe une riche littérature dans le domaine du débruitage de signaux numériques, en particulier pour le débruitage d'images. On fait ici un résumé des grandes classes d'algorithmes existantes. On notera pour alléger le texte dans le prochain chapitre la distance entre deux indices i et j d'un signal y par $d_{\mathbb{N}}(i,j)$ selon une métrique quelconque. De même, on notera par $d_{\mathbb{R}}(y_i, y_j)$ la distance entre la valeur de deux observations y_i et y_j. Lorsqu'il s'agira d'images, la notation y_i référera au $i^{\text{ième}}$ pixel d'une image $y \in \mathbb{R}^{l \times h}$.

Des fonctions appelées noyaux (*kernel functions*) sont souvent utilisées dans les algorithmes de débruitage. Ces fonctions servent à pénaliser certaines distances entre les observations y_i d'un signal. On note $K_h : \mathbb{R} \to \mathbb{R}$ un noyau général. Un noyau K_h est une fonction symétrique non négative qui décroit lorsque la distance $d_{\mathbb{N}}$ ou $d_{\mathbb{R}}$ entre deux éléments augmente. Elle atteint son maximum en 0, lorsque la distance entre deux éléments est nulle. Le paramètre h contrôle la vitesse à laquelle décroît K_h. L'exemple classique d'un noyau K_h est le noyau gaussien. On peut définir une version 1-D de ce noyau par

$$G_h(x) := \frac{1}{h\sqrt{2\pi}} \,.\, \exp\left(-\frac{x^2}{2h^2}\right), \tag{2.1}$$

où h est la largeur du noyau gaussien.

On définit le voisinage de rayon p d'une observation d'indice i selon une distance $d_{\mathbb{N}}$ comme étant

$$\mathcal{V}_{i,p} := \{j \in \mathcal{I}_n \mid d_{\mathbb{N}}(i,j) \leq p\}, \tag{2.2}$$

où on rappelle que $\mathcal{I}_n$ est l'ensemble de tous les indices d'un signal.

2.1 Filtres locaux

Le filtrage d'un signal bruité y consiste à remplacer chaque observation bruitée y_i par une moyenne pondérée de la valeur des observations dans leur voisinage. Dans un filtre local, plus une observation y_j est loin de y_i, moins sa valeur contribue au calcul de la valeur débruitée en i. On obtient la formule suivante :

$$\hat{s}_i = \frac{1}{Z_i} \sum_{j \in \mathcal{V}_{i,p}} K_h(\, d_{\mathbb{N}}(i,j)\,) \,.\, y_j, \tag{2.3}$$

où Z_i est un facteur de normalisation :

$$Z_i = \sum_{j \in \mathcal{V}_{i,p}} K_h(\, d_{\mathbb{N}}(i,j)\,).$$

La taille p du voisinage autour d'une observation y_i dans lequel on effectue la moyenne, la fonction K_h ainsi que le paramètre h sont des paramètres à ajuster. Gonzalez et Woods (2002) présente un filtre local utilisant un noyau gaussien, c'est-à-dire $K_h = G_h$ dans l'équation (2.3). On appelle alors ce filtre un filtre gaussien (*gaussian smoothing filter*). La relation entre $\hat{s}$ et y est linéaire pour les filtres gaussiens. Ce filtre atténue effectivement le bruit d'un signal dans ses parties lisses, mais a le défaut d'arrondir les variations abruptes de celui-ci (Jain et Tyagi, 2016). Le signal estimé $\hat{s}$ de la figure 1.3 présenté dans l'introduction a été obtenu en prenant la distance d euclidienne, le paramètre $p = 10$ et le paramètre $h = 5$.

Tomasi et Manduchi (1998) proposent un filtre appelé filtre bilatéral qui vise à préserver les transitions abruptes des signaux. On remarque dans ce travail qu'on peut définir deux distances entre deux observations bruitées y_i et y_j : la distance spatiale entre les deux observations, $d_{\mathbb{N}}(i,j)$, exploitée par le filtre gaussien (voir équation 2.3), et la distance entre les valeurs de ces deux observations, $d_{\mathbb{R}}(y_i, y_j)$ (qu'on appelle parfois distance photométrique dans le cas d'images). Le filtre bilatéral tire parti de ces deux distances et donc de la structure locale de y autour d'une observation y_i. On obtient la valeur débruitée en i suivante :

$$\hat{s}_i = \frac{1}{Z_i} \sum_{j \in \mathcal{V}_{i,p}} K_{h_1}(\, d_{\mathbb{N}}(i,j)\,) \,.\, K_{h_2}(\, d_{\mathbb{R}}(y_i, y_j)\,) \,.\, y_j, \tag{2.4}$$

où Z_i est encore une fois un facteur de normalisation :

$$Z_i = \sum_{j \in \mathcal{V}_{i,p}} K_{h_1}(\, d_{\mathbb{N}}(i,j)\,) \,.\, K_{h_2}(\, d_{\mathbb{R}}(y_i, y_j)\,).$$

Le filtre bilatéral permet de diminuer le poids des observations y_j si celles-ci ont des valeurs très différentes de y_i, et ce même si y_j étaient spatialement près de y_i (si $d_{\mathbb{N}}(i,j)$ était petite). Ceci permet d'améliorer la qualité de l'estimé près des transitions abruptes. Contrairement au filtre gaussien, la relation entre $\hat{s}$ et y est non linéaire. Takeda et al. (2007) généralise l'approche du filtre bilatéral en considérant les filtres locaux d'un point de vue statistique comme des méthodes de régression par noyau (*kernel regression*).

Le fait de tirer parti de la structure locale des observations y_i pour produire un estimé $\hat{s}_i$ se rapproche de la philosophie de l'algorithme qu'on présentera dans la suite. Cependant on remarque que ces filtres utilisent toujours des moyennes locales pour produire une valeur estimée $\hat{s}_i$ en un point donné i. Notre algorithme produira quant à lui l'estimé $\hat{s}$ à partir de la résolution d'un problème d'optimisation impliquant tout le signal, dans lequel l'information relative aux structures locales du signal sera intégrée. Chaque valeur estimée $\hat{s}_i$ sera donc un compromis entre l'analyse locale qu'on a faite autour de y_i et l'optimisation qu'on a faite sur tout le signal.

2.2 Filtres non locaux

Une extension des filtres locaux pour le débruitage d'images a été proposée avec l'algorithme *nonlocal means* introduit par Buades et al. (2005). Tout comme les filtres locaux, cet algorithme définit l'estimé d'un pixel donné $\hat{s}_i$ comme étant une moyenne pondérée d'autres pixels de l'image bruitée. Cependant, cette moyenne ne prend pas en compte dans sa pondération la distance spatiale $d_{\mathbb{N}}(i,j)$ entre les pixels de l'image, d'où le nom *nonlocal means*. L'algorithme pondère la valeur d'un pixel débruité $\hat{s}_i$ par la similitude entre la fenêtre (d'une certaine taille p) au milieu de laquelle se trouve le pixel bruité y_i et les autres fenêtres (de la même taille p) de l'image. Plus la fenêtre dans laquelle se trouve un pixel y_j donné est semblable à celle du pixel y_i, plus son poids est grand. Si on note par Y_i la fenêtre carrée centrée en i de taille $p \times p$ on obtient

$$\hat{s}_i = \frac{1}{Z_i} \sum_{j \in \mathcal{I}_n} K_h(\, d_{\mathbb{R}}(Y_i, Y_j)\,)\,.\, y_j, \tag{2.5}$$

où

$$Z_i = \sum_{j \in \mathcal{I}_n} K_h(\, d_{\mathbb{R}}(Y_i, Y_j)\,).$$

Dans ce cas, $d_{\mathbb{R}}$ réfère à une distance définie sur des matrices $\mathbb{R}^{p \times p}$. L'algorithme *nonlocal means* tire avantage, pour débruiter une image y, du fait que les images naturelles ont un certain degré d'auto-similarité (il existe de nombreuses parcelles semblables entre elles dans l'image). Suite à la création de cet algorithme, plusieurs autres avec la même philosophie ont suivi, réunis sous le nom de méthodes par filtres non locaux. Une amélioration notable de l'algorithme *nonlocal means* a été apportée par Kervrann et Boulanger (2006), où les auteurs proposent une version itérative de l'algorithme. La taille et la forme des fenêtres Y_j changent alors à chaque étape selon la variance de l'estimé en j.

Les méthodes par filtres non locaux sont assez différentes de l'algorithme qu'on présentera plus loin dans le sens où celles-ci ne s'appliquent qu'à des images et tirent parti de l'hypothèse d'auto-similarité mentionnée plus haut. On ne fera pas cette hypothèse dans notre algorithme. Lorsque cette hypothèse n'est pas respectée, l'image estimée souffre de défauts appelés *rare patch effect* (Louchet et Moisan, 2011).

2.3 Filtres fréquentiels

Il est possible d'exprimer un signal donné dans une base représentant les fréquences présentes dans ce signal. Cette expression est rendue possible grâce à la transformée de Fourier. On observe la plupart du temps que l'information contenue dans les signaux naturels est concentrée dans les hautes et les moyennes fréquences. Dans le problème de débruitage qui nous intéresse dans ce mémoire, le bruit ajouté est un bruit blanc. Une caractéristique de ce type de bruit est que ses composantes dans le domaine de Fourier sont distribuées également dans toutes les fréquences. À partir de ces observations, une méthode de débruitage simple consiste à mettre à 0 les composantes associées aux basses fréquences d'un signal bruité y, ces composantes étant censées être associées au bruit. On appelle cette méthode de débruitage le filtrage par un filtre passe-bas. Gonzalez et Woods (2002) présentent en détail cette technique et mettent également en évidence que celle-ci est analogue au filtrage par un filtre gaussien, présenté dans la section 2.1. Les défauts d'un estimé $\hat{s}$ produit par un filtre gaussien, à savoir l'atténuation des variations abruptes dans celui-ci, seront donc également présents dans l'estimé $\hat{s}$ produit suite à un filtrage par un filtre passe-bas.

2.4 Transformation en bases d'ondelettes

Une autre manière d'aborder le problème de débruitage est de procéder par seuillage des coefficients de y dans une base où le signal pur d'origine s possède une représentation creuse (toute l'information du signal est contenue dans un petit nombre de coefficients de la base). Les bases d'ondelettes ont été développées pour avoir cette propriété. Pour une revue de la théorie entourant les bases d'ondelettes et l'utilisation de celles-ci dans le domaine du traitement de signal, voir Mallat (1999). Donoho et Johnstone (1994) présentent la procédure générale de débruitage suivante :

1. Effectuer la transformation en ondelettes de y.
2. Modifier les coefficients de y dans sa base d'ondelettes, selon une certaine règle (seuillage).
3. Effectuer la transformation inverse afin de produire l'estimé $\hat{s}$.

Différentes règles pour le seuillage des coefficients à l'étape 2 ont été développées (Donoho et Johnstone, 1995 ; Nason, 1996 ; Chipman et al., 1997). Un problème des méthodes de débruitage par ondelettes surgit lorsque le signal d'intérêt s possède de petits coefficients, non nuls, dans sa base d'ondelettes. Ces coefficients sont associés aux endroits où le signal possède des discontinuités. Ceux-ci seront alors coupés à l'étape 2, ce qui crée sur l'estimé des artefacts de forme oscillante (*ringing artefacts*) aux endroits des discontinuités du signal (Beheshti et Dahleh, 2005). Ces défauts viennent du fait que les bases d'ondelettes sont fixes et ne s'adaptent pas à la structure particulière du signal y. Coifman et Donoho (1995) proposent une méthode où le signal débruité $\hat{s}$ est une moyenne de plusieurs estimés par seuillage des coefficients d'ondelettes de versions translatées du signal bruité y. Ceci a pour effet de diminuer l'importance des artefacts oscillants. On appelle cette méthode *Translation invariant wavelet thresholding* (TIWT). Foi et al. (2007) proposent une base adaptative permettant de produire un estimé qui préserve mieux les discontinuités d'une image donnée en tenant compte de la structure particulière de celle-ci.

L'estimé $\hat{s}$ qu'on propose dans ce mémoire est la solution d'un problème d'optimisation, l'algorithme ne se base pas en ce sens sur la même approche que les algorithmes par seuillage de coefficients d'ondelettes. Cependant, on verra que l'analyse statistique locale faite sur le signal y dans notre algorithme pourra être vue comme la construction d'une base adaptée au signal s.

2.5 Régularisation

Il est possible de voir le problème de débruitage sous l'angle d'un problème de régularisation. Ce type de problèmes est étudié par Tikhonov et Arsenin (1977). Appliqué au contexte de débruitage, on peut alors définir l'estimé $\hat{s}$ d'un signal s comme étant la solution minimale d'une fonction de coût :

$$\hat{s} \in \underset{x \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|y - x\|^2 + \lambda \, \mathcal{J}(x) \right\}. \tag{2.6}$$

Dans l'équation (2.6), la fonction $\mathcal{J}(x)$ est une fonction de régularisation (qu'on appelle *prior* en anglais) et est construite afin de promouvoir certains types de solutions (par exemple des solutions constantes par morceaux). Le paramètre $\lambda \in \mathbb{R}^+$ contrôle l'importance relative entre le terme d'attachement aux données $\|y-x\|^2$ et le terme de régularisation $\mathcal{J}(x)$. Une fonction de régularisation populaire dans le contexte de débruitage est la fonction de variation totale (VT). L'utilisation de celle-ci dans ce contexte est proposée par Rudin et al. (1992). Cette fonction est définie comme la norme ℓ^1 du gradient de x :

$$\mathcal{J}(x) = |\nabla x|, \tag{2.7}$$

où on approxime $|\nabla x|$ par différences finies. Pour la version 1-D de l'algorithme, on obtient alors

$$\hat{s} \in \underset{x \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|y - x\|^2 + \lambda \sum_{i=1}^{n-1} |x_{i+1} - x_i| \right\}. \tag{2.8}$$

L'utilisation de la norme ℓ^1 dans cette fonction de régularisation permet de produire un estimé $\hat{s}$ préservant les variations abruptes d'un signal s donné (Strong et Chan, 2003). Cependant, cette norme amène également des difficultés analytiques puisque la fonction à minimiser n'est pas dérivable en 0. Différentes méthodes numériques ont été développées pour résoudre (2.8) dans sa version 2-D, voir par exemple Vogel et Oman (1996) et Chambolle (2004). La résolution de (2.8) aura tendance à créer une solution constante par morceaux, ce qui aura pour effet de créer des artefacts en forme d'escaliers (*staircaising effect*) lorsque le signal pur s n'était pas lui-même constant par morceaux (Chambolle et Lion, 1997). Selesnick et al. (2015) propose une version légèrement modifiée de la fonction de régularisation dans le cas de signaux constants par morceaux afin d'améliorer les résultats. On comparera les résultats de ce travail avec l'algorithme proposé dans ce mémoire dans le chapitre 5.

Le choix du paramètre λ dans les problèmes de régularisation est un problème non trivial, une revue de quelques méthodes populaires est faite par Thompson et al. (1991). Pour le cas particulier du problème de variation totale, Ramani et al. (2008) présente une méthode stochastique afin de prédire l'erreur quadratique moyenne (MSE) de l'estimé produit par l'algorithme étant donné un paramètre λ donné. On choisit ensuite λ qui minimise cette prédiction de l'erreur.

L'algorithme présenté dans ce mémoire ressemble à la formulation des algorithmes de régularisation, en ce sens qu'on proposera aussi un estimé $\hat{s}$ comme étant la solution d'un problème de minimisation. Cependant, ce problème de minimisation sera un problème avec contrainte et il n'y aura pas de paramètre λ à choisir. De plus, on fera une analyse statistique préliminaire du signal bruité y afin de construire sur mesure l'analogue de la fonction de régularisation $\mathcal{J}(x)$ pour chaque signal bruité y.

2.6 Apprentissage statistique

Avec le nombre grandissant de données disponibles, les méthodes par apprentissage statistique (*machine learning*) sont de plus en plus populaires dans plusieurs domaines. Des algorithmes performants basés sur ces méthodes ont été proposés pour s'attaquer au problème de débruitage de signaux numériques. Une banque de données contenant beaucoup de signaux purs ainsi que leur version bruitée est utilisée afin d'entraîner ces algorithmes à produire un bon estimé $\hat{s}$. Elad et Aharon (2006) utilisent une méthode d'apprentissage statistique afin de créer une base dans laquelle chaque fenêtre de l'image à débruiter possède une représentation creuse. Burger et al. (2012) propose d'entraîner un réseau de neurones sur une banque d'images afin d'effectuer le débruitage. L'entraînement est basé sur la minimisation de l'erreur quadratique entre les images pures et les images débruitées.

L'optique générale de l'algorithme proposé dans ce mémoire n'est pas le même que les algorithmes cités plus haut, puisqu'on ne supposera pas qu'on a à notre disposition une banque de signaux purs et bruités et l'algorithme ne sera pas construit en ce sens. On fera plutôt une hypothèse de régularité sur les signaux à débruiter, à savoir que ceux-ci sont lisses par morceaux.

2.7 Méthodes mixtes

On a fait plus haut un survol des grandes classes d'algorithmes de débruitage et mentionné les travaux originaux dans chacune de ces classes. Il existe cependant beaucoup de travaux utilisant plusieurs de ces méthodes à la fois, formant ainsi des méthodes mixtes. Mairal et al. (2009) proposent un algorithme de débruitage combinant l'apprentissage statistique avec un filtre non local. Sutour et al. (2014) présente un algorithme de variation totale combiné encore une fois avec un filtre non local. L'algorithme nommé BM3D (pour *Block Matching 3-D*) introduit par Dabov et al. (2007) peut être considéré comme l'état-de-l'art en matière de débruitage d'images. Celui-ci se base sur le principe des filtres non locaux, c'est-à-dire qu'il cherche dans toute l'image bruitée y des fenêtres semblables entre elles. Il regroupe ensuite ces fenêtres semblables dans des ensembles que les auteurs appellent blocs 3-D (d'où le nom de l'algorithme). Le débruitage est alors effectué sur chacun de ces blocs 3-D composés de fenêtres similaires en procédant par seuillage de leur coefficient dans le domaine fréquentiel (transformée de Fourier).

CHAPITRE 3 PRÉSENTATION DE L'ALGORITHME DE DÉBRUITAGE POUR DES SIGNAUX 1-D

L'algorithme de débruitage qu'on propose dans ce mémoire a pour objectif de produire un estimé $\hat{s}$ qui préserve les caractéristiques locales d'un signal pur s. On cherchera à extraire de l'information dans le signal bruité y afin d'éviter entre autres d'atténuer dans le signal estimé $\hat{s}$ les discontinuités présentes dans le signal pur s.

On fait l'hypothèse que le signal discret non bruité $s \in \mathbb{R}^n$ est une version échantillonnée de manière équidistante d'une fonction $h : [0,1] \to \mathbb{R}$ lisse par morceaux, c'est-à-dire que la fonction h ainsi que sa dérivée sont continues sur chaque morceau. Pour un nombre n d'observations, on définit le pas d'échantillonnage $\Delta x := 1/(n-1)$. On obtient une version s échantillonnée de la fonction h de la manière suivante :

$$s_i = h((i-1)\,\Delta x) \quad \text{pour } i = 1, ..., n.$$

On donne ici un exemple d'une fonction h lisse par morceaux :

$$h(x) = \begin{cases} 25\,x^2 + 2.5\,x & \text{si } x \in [0; 0,4[, \\ 3 & \text{si } x \in [0,4; 0,7[, \\ 10\,x - 7 & \text{si } x \in [0,7; 1]. \end{cases}$$

On échantillonne de manière équidistante cette fonction en prenant $n = 11$ observations. On obtient un signal discret $s \in \mathbb{R}^{11}$, où $s_i = h((i-1)\,0.1)$ (voir figure 3.1).

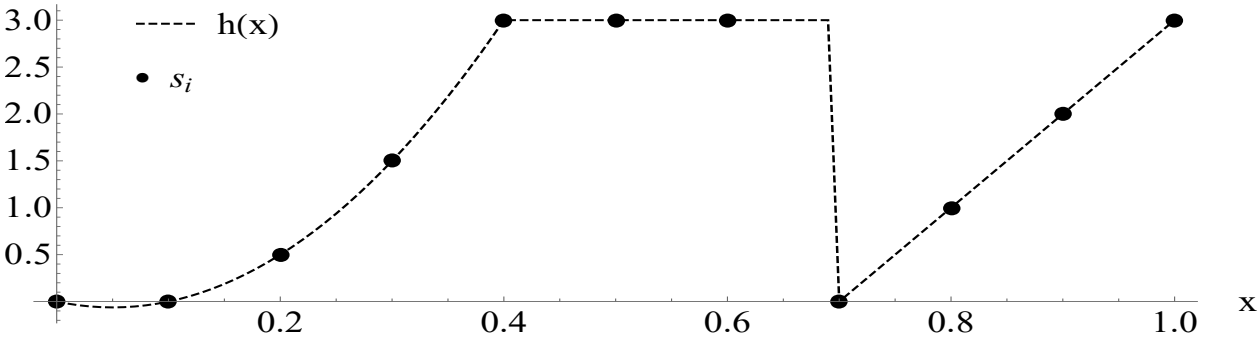


Figure 3.1 Exemple d'échantillonnage d'une fonction lisse par morceaux.

Une partie lisse d'une fonction peut être approximée localement par un polynôme (dévelop-

pement de Taylor). On cherchera dans notre algorithme à déterminer si des polynômes de différents degrés sont de bonnes approximations d'un signal donné localement. C'est le fait de modéliser les différentes parties d'un signal par un polynôme d'un certain degré qu'on appellera dans la suite les caractéristiques locales du signal. Le signal discret s de la figure 3.1 est localement quadratique pour la partie $(s_1, ..., s_5)^T \in \mathbb{R}^5$, localement constant pour la partie $(s_5, ..., s_7)^T \in \mathbb{R}^4$ et localement linéaire pour la partie $(s_8, ..., s_{11})^T \in \mathbb{R}^5$. De plus, la partie du signal $(s_3, ..., s_7)^T \in \mathbb{R}^5$ comporte une singularité en s_5 en ce sens que la fonction h n'est pas dérivable en $x = 0.4$, tout comme la partie $(s_7, ..., s_9)^T \in \mathbb{R}^3$ en s_8 en ce sens que la fonction h n'est pas continue en $x = 0,7$. C'est ce type de caractéristiques locales qu'on cherchera à estimer à partir du signal bruité y.

On appellera dans notre algorithme la modélisation des caractéristiques locales du signal pur s à partir du signal bruité y l'analyse statistique préliminaire de y. Celle-ci permettra la construction d'un modèle dit *composite* réunissant une collection de modèles simples localement adaptés à la forme du signal. L'analyse statistique préliminaire de y sera présentée dans la section 3.1. Suite à l'obtention du modèle composite, il faudra utiliser celui-ci afin de produire un estimé $\hat{s}$ du signal pur s. C'est ce qu'on fera dans les sections 3.2 et 3.3.

3.1 Analyse statistique préliminaire

On montre dans cette section comment on construit le modèle composite de s mentionné plus haut à partir d'une analyse statistique de y. Cette analyse a pour objectif d'obtenir de l'information détaillée et facile à interpréter quant aux caractéristiques locales du signal s. On prendra une multitude de décisions statistiques afin de savoir par exemple si les différentes parties du signal s sont localement constantes, linéaires, quadratiques ou non. On appellera chacune de ces décisions statistiques un modèle local de s. On espère tirer parti de cet ensemble de modèles locaux pour produire un estimé $\hat{s}$ qui préservera bien les caractéristiques locales du signal pur s.

3.1.1 Vecteurs orthogonaux selon les caractéristiques locales d'un signal

On propose dans notre algorithme d'inspecter à petite échelle le signal bruité y. Pour ce faire, on montre qu'il est possible de construire des vecteurs v qui sont orthogonaux localement à une partie d'un signal selon les caractéristiques de cette partie, par exemple si elle est constante, linéaire ou quadratique comme dans la figure 3.1.

Signaux localement constants

Soit un signal s constant sur deux points consécutifs, c'est-à-dire $s_i = s_{i+1} = a$, pour $a \in \mathbb{R}$. On définit le vecteur $\tilde{s} := (s_i, s_{i+1})^T \in \mathbb{R}^2$ représentant cette partie constante du signal. On cherche un vecteur $v = (v_1, v_2)^T \in \mathbb{R}^2$ qui est orthogonal à $\tilde{s}$ pour tout $a \in \mathbb{R}$.

$$v^T \cdot \tilde{s} = 0 \iff v_1\, a + v_2\, a = 0. \tag{3.1}$$

On trouve que n'importe quel vecteur de la forme $v = K\,(-1, 1)^T$, où $K \in \mathbb{R}$ est une constante, satisfait (3.1). En prenant $K = 1$, le vecteur $v = (-1, 1)^T$ est donc orthogonal aux parties constantes sur deux points consécutifs d'un signal. Soit maintenant un signal possédant la même valeur sur deux points non consécutifs, c'est-à-dire $s_i = s_{i+k} = a$, pour $a \in \mathbb{R}$ et $k > 1$. Pour la partie du signal $\tilde{s} := (s_i, s_{i+1}, ..., s_{i+k}) \in \mathbb{R}^{k+1}$, il est également possible de définir un vecteur $v \in \mathbb{R}^{k+1}$ qui lui soit orthogonal :

$$(v_1, 0, ..., 0, v_{k+1}) \cdot \tilde{s} = 0 \iff v_1\, a + v_{k+1}\, a = 0, \tag{3.2}$$

et donc le vecteur $v = (-1, 0, ..., 0, 1)^T \in \mathbb{R}^{k+1}$ est orthogonal aux parties du signal $\tilde{s} = (s_i, s_{i+1}, ..., s_{i+k}) \in \mathbb{R}^{k+1}$ ayant la même valeur en s_i et s_{i+k}. Si un signal est constant entre les indices i et $i + k$, c'est-à-dire $s_i = s_{i+1} = ... = s_{i+k} = a$ pour $a \in \mathbb{R}$, alors on a à notre disposition un certain nombre de relations d'orthogonalité :

$$\begin{aligned}
(-1, 1) \cdot (s_j, s_{j+1})^T &= 0, \quad \text{pour } j = i, ..., i + k - 1; \\
(-1, 0, 1) \cdot (s_j, s_{j+1}, s_{j+2})^T &= 0, \quad \text{pour } j = i, ..., i + k - 2; \\
&\vdots \\
(-1, 0, ..., 0, 1) \cdot (s_j, s_{j+1}, ..., s_{j+k})^T &= 0, \quad \text{pour } j = i.
\end{aligned}$$

Ces relations d'orthogonalité seront utiles dans la suite du travail pour caractériser les endroits où un signal est constant.

Signaux localement linéaires

Soit un signal s linéaire sur trois points consécutifs, c'est-à-dire $s_j = a\, j + b$, où $j = i, i+1, i+2$ et $a, b \in \mathbb{R}$. On cherche $v = (v_1, v_2, v_3) \in \mathbb{R}^3$ orthogonal à $\tilde{s} := (s_i, s_{i+1}, s_{i+2})^T \in \mathbb{R}^3$ pour tout a, $b \in \mathbb{R}$.

$$\begin{aligned} v^T \cdot \tilde{s} = 0 &\iff v_1(a\,i+b) + v_2(a\,(i+1)+b) + v_3(a\,(i+2)+b) = 0, \\ &\iff a\,i\,(v_1+v_2+v_3) + b\,(v_1+v_2+v_3) + a\,v_2 + 2a\,v_3 = 0, \\ &\iff (v_1+v_2+v_3) = 0 \quad \text{et} \quad v_2 = -2v_3, \end{aligned} \tag{3.3}$$

où la dernière équivalence vient du fait que l'orthogonalité doit être satisfaite peu importe a et b dans $\mathbb{R}$. On trouve que n'importe quel vecteur de la forme $v = K\,(1,-2,1)^T$ satisfait (3.3). En prenant $K = 1$, le vecteur $v = (1,-2,1)^T$ est donc orthogonal aux parties d'un signal s qui sont linéaires sur 3 points consécutifs. En se basant exactement sur la même approche que pour les signaux localement constants, on peut également définir un vecteur v qui soit orthogonal à une partie du signal dont les composantes sont linéaires sur trois points équidistants non consécutifs, c'est-à-dire $s_j = a\,j + b$, pour $j = i, i+k, i+2k$, $k > 1$ et $a, b \in \mathbb{R}$. En définissant cette partie du signal comme $\tilde{s} := (s_i, ..., s_{i+k}, ..., s_{i+2k})^T \in \mathbb{R}^{2k+1}$, on cherche $v \in \mathbb{R}^{2k+1}$ orthogonal à $\tilde{s}$ pour tout $a,\, b \in \mathbb{R}$. On trouve naturellement que le vecteur $v = (1, 0, .., 0, -2, 0, ..., 0, 1)^T \in \mathbb{R}^{2k+1}$ possède cette propriété. Celui-ci est donc orthogonal aux parties d'un signal s pour lesquelles 3 composantes équidistantes de distance k sont linéaires. Si une partie d'un signal est linéaire entre les indices i et $i+2k$, c'est-à-dire $s_j = aj + b$ où $j = i, ..., i+2k$, alors on a encore une fois un certain nombre de relations d'orthogonalité :

$$\begin{aligned} (1,-2,1)\cdot(s_j, s_{j+1}, s_{j+2})^T &= 0, \quad \text{pour } j = i, ..., i+2k-2; \\ (1,0,-2,0,1)\cdot(s_j, s_{j+1}, s_{j+2}, s_{j+3}, s_{j+4})^T &= 0, \quad \text{pour } j = i, ..., i+2k-4; \\ &\vdots \\ (1,0,...,0,-2,0,...,0,1)\cdot(s_j, ..., s_{j+k}, ...s_{j+2k})^T &= 0, \quad \text{pour } j = i. \end{aligned}$$

Comme pour les parties constantes d'un signal, ces relations d'orthogonalité seront utiles pour caractériser la linéarité locale d'un signal.

Signaux localement quadratiques, cubiques

La procédure qui permet de trouver des vecteurs v qui soient orthogonaux à des parties d'un signal quadratiques ou cubiques est exactement la même que celle pour les cas constants ou linéaires.

Soit un signal s quadratique sur 4 points consécutifs, c'est-à-dire $s_j = a\,j^2 + b\,j + c$ où $j = i, ..., i+3$ et a, b, $c \in \mathbb{R}$. En posant $\tilde{s} := (s_i, ..., s_{i+3})^T \in \mathbb{R}^4$, on peut trouver $v \in \mathbb{R}^4$ orthogonal à $\tilde{s}$ pour tout a, b, $c \in \mathbb{R}$:

$$v^T \cdot \tilde{s} = 0 \iff v = K\,(1, -3, 3, -1)^T. \tag{3.4}$$

En prenant $K = 1$, on obtient que le vecteur $v = (1, -3, 3, -1)^T$ est orthogonal aux parties d'un signal qui sont quadratiques sur 4 points consécutifs. De même, pour un signal cubique sur 5 points consécutifs, $s_j = a\,j^3 + b\,j^2 + c\,j + d$ où $j = i, ..., i + 4$ et a, b, c, $d \in \mathbb{R}$, on trouve que le vecteur $v = (1, -4, 6, -4, 1)^T$ est orthogonal à $\tilde{s} := (s_i, ..., s_{i+4})^T \in \mathbb{R}^5$ pour tout a, b, c, $d \in \mathbb{R}$.

On pourrait évidemment continuer la construction de vecteurs v orthogonaux à des polynômes d'ordre supérieur, mais nos modèles locaux s'arrêteront à l'ordre trois dans ce travail. De plus, il existe de nombreux autres vecteurs orthogonaux à des polynômes de degré zéro à trois que ceux construits plus haut. Par exemple, le vecteur $v = (1, -1, -1, 1)^T$ est orthogonal aux parties d'un signal linéaires sur quatre points consécutifs. Un choix a donc été fait dans la sélection des vecteurs v présentés ici.

On montre dans le tableau 3.1 un résumé des vecteurs v construits dans cette section et à quelle caractéristique locale ceux-ci sont orthogonaux.

Tableau 3.1 Vecteurs orthogonaux selon les caractéristiques locales d'un signal.

v	Orthogonal aux parties ...
$(-1, 1)^T$	Constantes sur 2 points
$(1, -2, 1)^T$	Linéaires sur 3 points
$(1, -3, 3, -1)^T$	Quadratiques sur 4 points
$(1, -4, 6, -4, 1)^T$	Cubiques sur 5 points

Remarques

On peut voir les vecteurs v construits plus haut comme les approximations en différences finies des dérivées (discrètes) première, seconde, troisième et quatrième du signal s. En effet, par l'approximation de la dérivée par différence avant, on a les relations suivantes.

1. Dérivée première :

$$s'_i \approx (s_{i+1} - s_i) \,/\, ((i + 1) - i) = (-1, 1) \cdot (s_i, s_{i+1})^T.$$

2. Dérivée seconde :

$$\begin{aligned} s_i'' \approx s_{i+1}' - s_i' &= (s_{i+2} - s_{i+1}) - (s_{i+1} - s_i), \\ &= (s_{i+2} - 2s_{i+1} + s_i), \\ &= (1, -2, 1) \cdot (s_i, s_{i+1}, s_{i+2})^T. \end{aligned}$$

3. Dérivée troisième :

$$\begin{aligned} s_i''' \approx s_{i+1}'' - s_i'' &= (s_{i+3} - 2s_{i+2} + s_{i+1}) - (s_{i+2} - 2s_{i+1} + s_i), \\ &= (s_{i+3} - 3s_{i+2} + 3s_{i+1} - s_i), \\ &= (-1, 3, -3, 1) \cdot (s_i, s_{i+1}, s_{i+2}, s_{i+3})^T. \end{aligned}$$

4. Dérivée quatrième :

$$\begin{aligned} s_i'''' \approx s_{i+1}''' - s_i''' &= (s_{i+4} - 3s_{i+3} + 3s_{i+2} - s_{i+1}) - (s_{i+3} - 3s_{i+2} + 3s_{i+1} - s_i), \\ &= (s_{i+4} - 4s_{i+3} + 6s_{i+2} - 4s_{i+1} + s_i), \\ &= (1, -4, 6, -4, 1) \cdot (s_i, s_{i+1}, s_{i+2}, s_{i+3}, s_{i+4})^T. \end{aligned}$$

Ceci est simplement une autre interprétation des vecteurs orthogonaux v. Par exemple, si un signal s est localement constant sur 2 points consécutifs $(s_i, s_{i+1})^T$, alors l'approximation de sa dérivée première $(-1, 1) \cdot (s_i, s_{i+1})^T$ devrait être nulle.

3.1.2 Tests locaux

Les vecteurs v définis plus haut serviront à tester, à partir du signal bruité y, s'il est plausible qu'une certaine partie du signal pur s sous-jacent ait une caractéristique locale donnée.

Matrices circulantes

On définit la matrice circulante H_v selon un vecteur $v \in \mathbb{R}^m$ comme étant

$$H_v := \begin{pmatrix} v^T & & & \\ & v^T & & (0) \\ & & \ddots & \\ & (0) & & \\ & & & v^T \end{pmatrix} \in \mathbb{R}^{(n-(m-1)) \times n}. \tag{3.5}$$

On rappelle que n est la taille des signaux ($y \in \mathbb{R}^n$). Ce type de matrice est dite circulante puisque chaque ligne est obtenue en décalant la précédente d'un élément vers la droite. On donne deux exemples de ces matrices H_v pour les vecteurs $v = (-1, 1)^T \in \mathbb{R}^2$ et $v = (1, -2, 1)^T \in \mathbb{R}^3$:

$$H_{(-1,1)} = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & (0) \\ & & \ddots & \ddots & \\ & (0) & & & \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n}.$$

$$H_{(1,-2,1)} = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & (0) \\ & & \ddots & \ddots & \ddots & \\ & (0) & & & & \\ & & & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(n-2)\times n}.$$

Ces matrices H_v permettent d'effectuer le produit scalaire entre un vecteur $v \in \mathbb{R}^m$ et chaque m-uplet d'un signal $y \in \mathbb{R}^n$. On appelle le $i^{\text{ième}}$ m-uplet d'un vecteur $y \in \mathbb{R}^n$ le vecteur $(y_i, ..., y_{i+m-1})^T \in \mathbb{R}^m$. Si on note par $[z]_i$ la $i^{\text{ième}}$ composante d'un vecteur z, on trouve en effet que

$$[\, H_v\, y\,]_i = v^T \cdot (y_i\,,\, \ldots\,,\, y_{i+m-1})^T.$$

Par exemple, si on prend un signal $y \in \mathbb{R}^5$ et $v = (-1, 1)^T$, on obtient

$$H_v\, y = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} y_2 - y_1 \\ y_3 - y_2 \\ y_4 - y_3 \\ y_5 - y_4 \end{pmatrix}.$$

Tests d'orthogonalité

Les matrices H_v définies plus haut serviront à décrire les tests statistiques qu'on fera à partir des observations bruitées y permettant d'estimer les caractéristiques locales de s. On tire parti pour ceci de notre connaissance du modèle du bruit ajouté $w \in \mathbb{R}^n$, à savoir que les composantes w_i de ce vecteur sont i.i.d. et proviennent d'une loi normale de moyenne nulle et de variance σ^2 connue. On rappelle ici une propriété des variables aléatoires de distribution normale. Si X_1 et X_2 sont deux variables aléatoires indépendantes telles que $X_1 \sim \mathcal{N}(\nu_1, \tau_1^2)$ et $X_2 \sim \mathcal{N}(\nu_2, \tau_2^2)$, où ν_i et τ_i^2 sont la moyenne et la variance de X_i, alors on a que

$$a\,X_1 + b\,X_2 \sim \mathcal{N}(a\,\nu_1 + b\,\nu_2,\, a^2\,\tau_1^2 + b^2\,\tau_2^2), \quad \text{pour tout } a,\, b \in \mathbb{R}. \tag{3.6}$$

On considère maintenant un signal pur $s \in \mathbb{R}^n$ et le vecteur de variables aléatoires $w \in \mathbb{R}^n$ tel que $w_i \sim \mathcal{N}(0, \sigma^2)$ représentant le bruit additif. Étant donné un vecteur $v \in \mathbb{R}^m$ et la matrice H_v qui lui est associé, on a par la propriété (3.6) que

$$[H_v\, w]_i = v^T \cdot (w_i,\, \ldots,\, w_{i+m-1})^T \sim \mathcal{N}(0,\, \|v\|^2\,\sigma^2). \tag{3.7}$$

Soit $\mu_i \in \mathbb{R}$ la valeur du produit scalaire entre un vecteur $v \in \mathbb{R}^m$ et le $i^{\text{ième}}$ m-uplet du signal pur s :

$$\mu_i := [H_v\, s]_i. \tag{3.8}$$

On considère un signal bruité y provenant de ce signal s auquel on ajouté le vecteur de variables aléatoires w. On a que

$$\begin{aligned} [H_v\, y]_i &= [H_v\,(s + w)]_i, \\ &= [H_v\, s]_i + [H_v\, w]_i, \\ &= \mu_i + [H_v\, w]_i. \end{aligned} \tag{3.9}$$

À partir de (3.9) et de la propriété (3.7), on obtient que la composante $[H_v\, y]_i$ suit une loi normale de moyenne μ_i et de variance $\|v\|^2\,\sigma^2$:

$$[H_v\, y]_i \sim \mathcal{N}(\mu_i,\, \|v\|^2\,\sigma^2) \tag{3.10}$$

Grâce à cette distribution, on peut créer un test statistique à partir du signal bruité y afin de déterminer s'il est plausible que la partie du signal pur s entre les indices i et $i + m - 1$ soit orthogonale au vecteur $v \in \mathbb{R}^m$, autrement dit s'il est plausible que $\mu_i = 0$.

Soit une valeur donnée du produit scalaire entre un vecteur $v \in \mathbb{R}^m$ et le $i^{\text{ième}}$ m-uplet d'un signal bruité y. Notre notation pour cette valeur est $[H_v\, y]_i$. Cette valeur provient d'une distribution normale de moyenne μ_i et de variance $\|v\|^2\,\sigma^2$, voir équation 3.10. Le paramètre μ_i de cette distribution est inconnu. On définit un test d'hypothèse pour ce paramètre. On pose comme hypothèse nulle (notée H_0) que ce paramètre μ_i vaut 0. L'hypothèse alternative (notée H_1) correspond à l'hypothèse comme quoi $\mu_i \neq 0$ (test bilatéral). On a

$$\begin{aligned} \mathrm{H}_0 : &\quad \mu_i = 0; \\ \mathrm{H}_1 : &\quad \mu_i \neq 0. \end{aligned}$$

On rejette l'hypothèse H_0 si la valeur de $[H_v\, y]_i$ est significativement plus grande ou plus petite que 0. On considère alors qu'il est peu probable que la loi normale de variance $\|v\|^2\, \sigma^2$ d'où provient la valeur $[H_v\, y]_i$ ait une moyenne μ_i nulle. Étant donné un seuil de rejet α, on peut déterminer une région de rejet de l'hypothèse H_0. Par exemple, pour $\alpha = 0.05$, on rejette l'hypothèse H_0 s'il y a moins de 5 % de chance qu'on obtienne la valeur $[H_v\, y]_i$ si celle-ci vient d'une normale de moyenne nulle et de variance $\|v\|^2\, \sigma^2$. La zone de rejet est déterminée grâce à la table des quantiles de la loi normale centrée réduite. Pour un seuil de rejet α donné, on trouve δ_α tel que

$$\text{si} \quad \Big|\, [H_v\, y]_i \,\Big| \;\geq\; \delta_\alpha\, \|v\|\, \sigma, \quad \text{on rejette } \mathrm{H}_0. \tag{3.11}$$

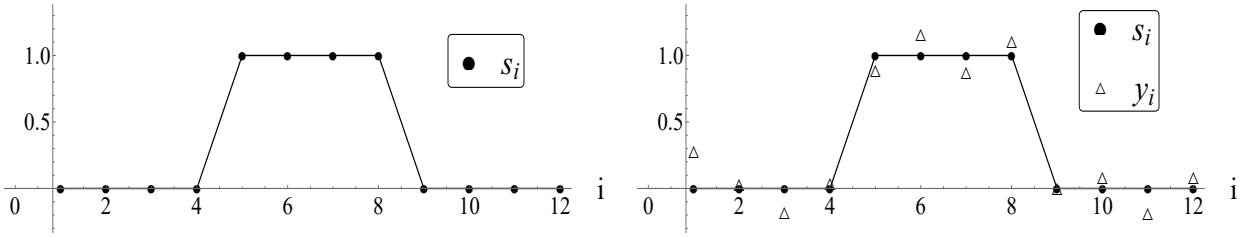
Pour $\alpha = 0.05$, on doit prendre $\delta_\alpha = 1.96$. Ce test sert à classer, étant donné un vecteur $v \in \mathbb{R}^m$, tous les m-uplets d'un signal bruité y dans deux catégories. L'une correspond aux m-uplets d'indice i de y qui ont passé le test statistique selon un certain seuil α, à savoir $\Big|\, [H_v\, y]_i \,\Big| < \delta_\alpha\, \|v\|\, \sigma$. Pour ces m-uplets, on considère plausible que les m-uplets correspondants du signal pur s inconnu étaient orthogonaux au vecteur v, c'est-à-dire qu'on considère plausible que $\mu_i = 0$. L'autre catégorie correspond aux m-uplets d'indice i de y tel que $\Big|\, [H_v\, y]_i \,\Big| \geq \delta_\alpha\, \|v\|\, \sigma$. On considère dans ce cas que les m-uplets correspondants du signal pur s n'étaient plausiblement pas orthogonaux à v.

On a construit dans la section 3.1.1 des vecteurs v qui ont la propriété d'être orthogonaux aux différentes parties d'un signal selon leur caractéristique (constant, linéaire, quadratique, etc). Le test statistique décrit dans cette section permettra de faire une multitude de tests locaux afin de déterminer, selon le signal bruité y, s'il est plausible que le signal pur s sous-jacent ait localement certaines de ces caractéristiques. C'est le fait qu'un m-uplet de s soit plausiblement orthogonal à un vecteur $v \in \mathbb{R}^m$ selon un seuil α qu'on appelle modèle local de s. C'est l'ensemble des résultats des tests d'orthogonalités entre des vecteurs $v \in \mathbb{R}^m$ et les m-uplets du signal bruité y qu'on appelle l'analyse statistique préliminaire de y.

Exemple 1. *On présente ici un exemple avec le signal pur $s \in \mathbb{R}^{12}$ constant par morceaux suivant :*

$$s_i = \begin{cases} 0 & si\ 1 \leq i \leq 4, \\ 1 & si\ 5 \leq i \leq 8, \\ 0 & si\ 9 \leq i \leq 12. \end{cases}$$

On ajoute à ce signal un bruit blanc de moyenne nulle et d'écart-type $\sigma = 0.15$. Voir figure 3.2.



(a) Signal constant par morceaux $s \in \mathbb{R}^{12}$. (b) Signal bruité y, $\sigma = 0.15$.

Figure 3.2 Tests d'orthogonalité, exemple 1 [1].

On teste l'orthogonalité entre tous les couples $(s_i, s_{i+1})^T$ du signal pur et le vecteur $v = (-1, 1)^T$. Si un couple est orthogonal à ce vecteur v, le signal est constant sur ces deux points. On crée le test statistique associé à ce vecteur v avec le signal bruité y. On décide de rejeter l'hypothèse d'orthogonalité H_0 selon un seuil α de 5%. Dans ce cas, on obtient :

$$si \quad \left| [H_v\, y]_i \right| \ \geq\ \delta_\alpha \, \|v\| \, \sigma \ =\ 1.96 \times \sqrt{2} \times 0.15 \ \approx\ 0.416, \quad on\ rejette\ H_0.$$

Dans le tableau 3.2, on montre la valeur absolue du produit scalaire entre chaque couple $(y_i, y_{i+1})^T$ et v. On montre ensuite la décision prise selon le seuil de 5 %, à savoir si on accepte ou non l'hypothèse d'orthogonalité H_0 pour chacun de ces couples. Selon cette analyse statistique préliminaire, on rejette l'hypothèse d'orthogonalité au vecteur $v = (-1, 1)^T$ pour les couples $(s_4, s_5)^T$ et $(s_8, s_9)^T$. Les tests statistiques ont bien fonctionné dans ce cas, puisqu'il s'agit effectivement des deux seuls couples $(s_i, s_{i+1})^T$ du signal pur s qui n'étaient pas constants.

1. On relie les observations s_i seulement pour aider à la visibilité, s est un signal discret.

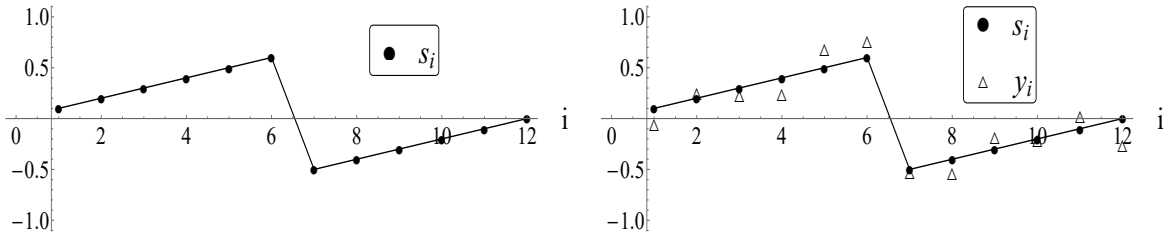
Tableau 3.2 Tests d'orthogonalité, exemple 1.

Indice i	$\lvert\,[H_v\,y]_i\,\rvert = \lvert y_{i+1} - y_i\rvert$	Accepte H_0
1	0.253	Oui
2	0.208	Oui
3	0.219	Oui
4	0.847	Non
5	0.271	Oui
6	0.282	Oui
7	0.228	Oui
8	1.104	Non
9	0.083	Oui
10	0.270	Oui
11	0.273	Oui

Exemple 2. *On présente un exemple avec le signal pur $s \in \mathbb{R}^{12}$ linéaire par morceaux suivant :*

$$s_i = \begin{cases} 0.1\,i & si\ 1 \le i \le 6, \\ 0.1\,i - 1.2 & si\ 7 \le i \le 12. \end{cases}$$

On ajoute à ce signal un bruit blanc de moyenne nulle et d'écart-type $\sigma = 0.15$. Voir figure 3.3.



(a) Signal linéaire par morceaux $s \in \mathbb{R}^{12}$. (b) Signal bruité y, $\sigma = 0.15$.

Figure 3.3 Tests d'orthogonalité, exemple 2.

Cette fois, on teste l'orthogonalité entre tous les triplets $(s_i, s_{i+1}, s_{i+2})^T$ et le vecteur $v = (1, -2, 1)^T$. L'orthogonalité à ce vecteur v implique la linéarité du triplet. On crée le test statistique associé à ce v avec le signal bruité y. On décide encore une fois de rejeter l'hypothèse d'orthogonalité H_0 selon un seuil α de 5%. On obtient :

$$si \quad \left| [H_v \cdot y]_i \right| \geq \delta_\alpha \|v\| \sigma = 1.96 \times \sqrt{6} \times 0.15 \approx 0.720, \quad on\ rejette\ H_0.$$

Dans le tableau 3.3, on montre le résultat des tests statistiques sur les triplets de y. Selon cette analyse statistique préliminaire, on rejette l'hypothèse d'orthogonalité au vecteur $v = (1, -2, 1)^T$ pour les triplets $(s_5, s_6, s_7)^T$ et $(s_6, s_7, s_8)^T$. Les tests statistiques ont encore une fois bien fonctionné dans ce cas, puisqu'il s'agit des deux seuls triplets $(s_i, s_{i+1}, s_{i+2})^T$ du signal pur qui n'étaient pas linéaires.

Tableau 3.3 Tests d'orthogonalité, exemple 2.

Indice i	$\lvert [H_v\, y]_i \rvert$	Accepte H_0
1	0.307	Oui
2	0.015	Oui
3	0.435	Oui
4	0.361	Oui
5	1.363	Non
6	1.277	Non
7	0.357	Oui
8	0.372	Oui
9	0.256	Oui
10	0.514	Oui

Dans les deux derniers exemples, deux paramètres ont été choisis, soient le seuil α de rejet de l'hypothèse d'orthogonalité ainsi que le vecteur v pour lequel on teste l'orthogonalité. On discutera du choix de ces paramètres dans les chapitres 5 et 6.

Le modèle composite de s sera constitué des résultats de l'analyse statistique préliminaire, c'est-à-dire du résultat de tous ces tests d'orthogonalité avec des vecteurs v. On verra dans la prochaine section comment on rassemble ces résultats de manière compacte.

3.1.3 Écriture du modèle composite

On stocke l'information relative aux résultats de l'analyse statistique préliminaire avec des vecteurs v sous forme matricielle en modifiant les matrices H_v associées à ces vecteurs.

Soient un vecteur $v \in \mathbb{R}^m$, sa matrice H_v correspondante, un signal bruité $y \in \mathbb{R}^n$ provenant d'un signal pur $s \in \mathbb{R}^n$ et un seuil de rejet α de l'hypothèse d'orthogonalité. On définit un ensemble d'indices $\mathcal{I}_v$ contenant les indices des m-uplets de s pour lesquels on a rejeté l'hypothèse d'orthogonalité avec le vecteur v selon le seuil α suite à l'analyse statistique préliminaire :

$$\mathcal{I}_v := \left\{ i \in \{1, ..., n-(m-1)\} \ : \ \left| [H_v\, y]_i \right| \ \geq \ \delta_\alpha \, \|v\| \, \sigma \right\}.$$

On définit ensuite la matrice D_v comme étant la matrice H_v pour laquelle on a retiré les lignes d'indices $i \in \mathcal{I}_v$.

Retour aux exemples

Dans l'exemple 1 de la section précédente (3.1.2), pour le signal constant par morceaux $s \in \mathbb{R}^{12}$ et le vecteur $v = (-1, 1)^T$, l'ensemble $\mathcal{I}_v$ est $\{4, 8\}$, voir tableau 3.2. Ainsi, la matrice D_v correspondant à cette analyse statistique préliminaire est la matrice H_v pour laquelle on a retiré la $4^{\text{ième}}$ et la $8^{\text{ième}}$ ligne :

$$D_{(-1,1)} = \begin{pmatrix} -1 & 1 & & & & & & & & & & \\ & -1 & 1 & & & & & & & & & \\ & & -1 & 1 & & & & & & (0) & & \\ & & & & -1 & 1 & & & & & & \\ & & & & & -1 & 1 & & & & & \\ & & & & & & -1 & 1 & & & & \\ & & (0) & & & & & & -1 & 1 & & \\ & & & & & & & & & -1 & 1 & \\ & & & & & & & & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{9\times 12}.$$

Dans l'exemple 2, pour le signal linéaire par morceaux $s \in \mathbb{R}^{12}$ et le vecteur $v = (1, -2, 1)^T$, l'ensemble $\mathcal{I}_v$ est $\{5, 6\}$, voir tableau 3.3. La matrice D_v correspondant à cette analyse est la matrice H_v pour laquelle on a retiré la $5^{\text{ième}}$ ligne et la $6^{\text{ième}}$:

$$D_{(1,-2,1)} = \begin{pmatrix} 1 & -2 & 1 & & & & & & & & & \\ & 1 & -2 & 1 & & & & & & (0) & & \\ & & 1 & -2 & 1 & & & & & & & \\ & & & 1 & -2 & 1 & & & & & & \\ & & & & & & 1 & -2 & 1 & & & \\ & & & & & & & 1 & -2 & 1 & & \\ & & (0) & & & & & & 1 & -2 & 1 & \\ & & & & & & & & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{8\times 12}.$$

Pseudo code pour la construction de D_v

On présente le pseudo code correspondant à la construction d'une matrice D_v étant donné un vecteur $v \in \mathbb{R}^m$, un vecteur $y \in \mathbb{R}^n$, un niveau de bruit σ et un seuil δ_α.

Algorithme 1 : Construction de la matrice D_v

Entrée : $y \in \mathbb{R}^n$, $\sigma \in \mathbb{R}$, $v \in \mathbb{R}^m$, $\delta_\alpha \in \mathbb{R}$.
Sortie : $D_v \in \mathbb{R}^{l \times n}$.

I. 1. Poser $D_v := \{\}$.
2. Poser $u_0 := \mathbf{0} \in \mathbb{R}^n$.
3. Construire H_v selon la définition (3.5).

II. Pour $i = 1, ..., n - (m-1)$:
si $\left| [H_v\, y]_i \right| < \delta_\alpha \|v\| \sigma$,
Poser $u_1 = u_0$.
Remplacer $u_1(i : i + m - 1) = v$.
Ajouter à D_v la ligne u_1.

III. Fin.

L'implémentation de l'algorithme proposé dans ce mémoire a été faite à l'aide du langage Mathematica. Les codes des différents algorithmes présentés dans ce mémoire, dont l'algorithme 1 plus haut, peuvent être trouvés à partir du lien mis dans l'annexe A.

Superposition des matrices D_v

Ce sont les matrices D_v qui permettent de stocker les résultats des tests d'orthogonalité avec un vecteur v donné. La raison pour laquelle on stocke l'information de cette manière deviendra claire dans la section 3.2, où on utilisera les résultats de l'analyse statistique préliminaire de y pour produire un estimé $\hat{s}$ du signal pur s.

Lorsque l'on testera plusieurs vecteurs v sur un signal bruité y, on aura plusieurs matrices D_v associées. Soit un ensemble $\{v_1, ..., v_M\}$ de M vecteurs $v_k \in \mathbb{R}^{m_k}$. On définit la matrice D (sans indice) comme étant la superposition de toutes les matrices D_{v_k} pour $k = 1, ..., M$:

$$D := \begin{pmatrix} D_{v_1} \\ D_{v_2} \\ \vdots \\ D_{v_M} \end{pmatrix} \in \mathbb{R}^{l \times n}. \tag{3.12}$$

Le nombre de lignes de D (noté l dans la définition 3.12) est inconnu en général puisque celui-ci dépend du résultat des tests d'orthogonalité entre le signal bruité y et les vecteurs v_k. Cependant, le nombre de colonnes de D sera toujours égal à n, la taille du signal bruité $y \in \mathbb{R}^n$. L'ordre dans lequel on superpose les matrices D_{v_k} n'a pas d'importance dans la définition de D. C'est la matrice D ainsi définie qui constitue le modèle composite du signal s et qui sera utilisé afin de produire un estimé $\hat{s}$. Pour un ensemble de vecteurs $v_1, ..., v_M$, cette matrice contient l'information quant aux endroits où l'on considère plausible que le signal pur s soit orthogonal avec ces vecteurs v_k. Puisque l'orthogonalité à ces vecteurs v_k peut être interprétée comme une caractérisation locale du signal, l'information contenue dans D constitue en quelque sorte une description des détails du signal.

3.1.4 Lien entre la matrice D et le signal s

Notre modèle composite d'un signal s est construit sous la forme d'une matrice D. Il existe un lien entre D et s, qu'on introduit à partir de l'exemple suivant.

Soit un signal constant $s \in \mathbb{R}^n$ ($s_1 = s_2 = ... = s_n$). On peut représenter ce signal comme étant un élément du noyau d'une application linéaire bien choisie. En effet, comme s est constant dans notre exemple, on a que

$$(s_2 - s_1) = (s_3 - s_2) = ... = (s_n - s_{n-1}) = 0.$$

Ainsi, si on définit l'application linéaire B comme étant

$$B := \begin{pmatrix} -1 & 1 & & & & \\ & -1 & 1 & & (0) & \\ & & \ddots & \ddots & & \\ & (0) & & & & \\ & & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(n-1) \times n},$$

on a que

$$\begin{aligned} B\,s &= (s_2 - s_1,\, s_3 - s_2,\, ...,\, s_n - s_{n-1})^T \in \mathbb{R}^{n-1}, \\ &= (0,\, ...,\, 0)^T \in \mathbb{R}^{n-1}. \end{aligned}$$

On a donc $s \in \ker B$. On peut remarquer que notre définition de H_v avec $v = (-1, 1)^T$ est exactement celle de l'application B, voir définition (3.5). De plus, si on construit D_v avec ce vecteur v, on enlèvera les lignes i de H_v pour lesquelles l'orthogonalité entre v et $(s_i, s_{i+1})^T$ n'est pas plausible au sens qu'on lui a donné plus haut. Or si l'on ne fait pas d'erreurs dans ces tests statistiques locaux (par exemple si le bruit ajouté est faible), on ne devrait pas enlever de lignes à H_v puisque tous les couples $(s_i, s_{i+1})^T$ sont effectivement orthogonaux à $v = (-1, 1)^T$. Comme on effectue l'analyse statistique préliminaire avec seulement un vecteur v dans cet exemple, on a que le modèle composite, par la définition (3.12), est $D = D_v$. On a donc que $D = D_v = H_v = B$ et par le fait même que $s \in \ker D$.

Cet exemple permet de voir la construction du modèle composite de s décrite plus haut comme étant la recherche d'une application linéaire D pour laquelle le signal pur s (qui est inconnu) serait un élément de son noyau ($s \in \ker D$). En effet, puisqu'on retire des matrices H_v les lignes où s n'est pas orthogonal à v, il ne reste dans D_v que des lignes où s est orthogonal à v. Ainsi, si on ne fait pas d'erreurs lors de l'analyse statistique préliminaire, alors on aura $D\,s = 0$ et $s \in \ker D$.

Revenons encore une fois aux exemples de la section 3.1.2. Pour l'exemple 1, si on applique la matrice D obtenue à partir des résultats l'analyse statistique préliminaire (tableau 3.2), on obtient

$$D\,s = \begin{pmatrix} -1 & 1 & & & & & & & & & & \\ & -1 & 1 & & & & & & & & & \\ & & -1 & 1 & & & & & & (0) & & \\ & & & & -1 & 1 & & & & & & \\ & & & & & -1 & 1 & & & & & \\ & & & & & & -1 & 1 & & & & \\ & & (0) & & & & & & -1 & 1 & & \\ & & & & & & & & & -1 & 1 & \\ & & & & & & & & & & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

On a donc bien dans cet exemple la relation $s \in \ker D$.

Pour l'exemple 2, on obtient

$$D\,s = \begin{pmatrix} 1 & -2 & 1 & & & & & & & & & \\ & 1 & -2 & 1 & & & & & & (0) & & \\ & & 1 & -2 & 1 & & & & & & & \\ & & & 1 & -2 & 1 & & & & & & \\ & & & & & & 1 & -2 & 1 & & & \\ & & & & & & & 1 & -2 & 1 & & \\ & & (0) & & & & & & 1 & -2 & 1 & \\ & & & & & & & & & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.6 \\ -0.5 \\ -0.4 \\ -0.3 \\ -0.2 \\ -0.1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

On trouve encore une fois que $s \in \ker D$. On utilisera dans la prochaine section cette propriété du modèle composite stocké dans la matrice D afin de produire un estimé $\hat{s}$ du signal pur s.

3.2 Produire un estimé à partir des résultats de l'analyse statistique préliminaire

On produit un estimé $\hat{s}$ du signal pur s en résolvant un problème de minimisation sous contrainte d'inégalité. La fonction à minimiser est construite à partir du modèle composite de s qu'on a construit suite à l'analyse statistique préliminaire de y.

On rappelle ici un résultat provenant de la définition de la variance d'une variable aléatoire. Si X est une variable aléatoire d'espérance $\mathbb{E}[X]$, alors on a que

$$\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2. \tag{3.13}$$

Cette formule s'applique à chacune des composantes w_i du vecteur $w \in \mathbb{R}^n$ représentant le bruit blanc additif dans notre modélisation du problème de débruitage. Le calcul suivant permet de définir la contrainte d'inégalité qu'on utilisera dans le problème d'optimisation. L'espérance de la distance euclidienne au carré entre le signal bruité y et le signal pur s est donnée par

$$\begin{aligned}
\mathbb{E}\left[\ \|s-y\|^2\ \right] &= \mathbb{E}\left[\ \|s-(s+w)\|^2\ \right], \\
&= \mathbb{E}\left[\ \|w\|^2\ \right], \\
&= \sum_{i=1}^{n} \mathbb{E}\left[w_i^2\right], \\
&= \sum_{i=1}^{n} \left(\mathrm{Var}[w_i] + \mathbb{E}[w_i]^2\right), \\
&= \sum_{i=1}^{n} \left(\sigma^2 + 0\right), \\
&= n\,\sigma^2. \qquad (3.14)
\end{aligned}$$

Pour que l'estimé $\hat{s}$ demeure cohérent avec le signal bruité y, on propose suite à ce calcul la contrainte suivante :

$$\|\hat{s}-y\|^2 \leq n\,\sigma^2. \qquad (3.15)$$

On veut donc que $\hat{s}$ ne s'éloigne pas trop de y et demeure dans la boule centrée en y de rayon $\sqrt{n}\,\sigma$. La fonction à minimiser sous cette contrainte est construite à partir du lien entre D et s présenté dans la section 3.1.4. Dans cette section, on a montré que la construction du modèle composite de s pouvait être vue comme la recherche d'une application D tel que $s \in \ker D$. Dans le cas où on a bien cette relation entre D et s, on obtient que

$$\begin{aligned}
D\,s = 0 \quad &\Longleftrightarrow \quad \|D\,s\|^2 = 0, \\
&\Longleftrightarrow \quad s^T\,D^T D\,s = 0, \\
&\Longleftrightarrow \quad s^T\,A\,s = 0,
\end{aligned}$$

avec $A := D^T D \in \mathbb{R}^{n\times n}$. Cette matrice A est symétrique et semi définie positive par construction. Plus un vecteur $x \in \mathbb{R}^n$ est près de l'espace vectoriel $\ker D$, plus la valeur de la fonction $f(x) := x^T\,A\,x$ est près de 0. On propose de définir l'estimé du signal s comme étant une solution au problème de minimisation suivant :

$$\boxed{\begin{cases} \min\limits_{x\in\mathbb{R}^n} & x^T A x, \\ \text{s.c.} & \|x-y\|^2 \ \leq\ n\,\sigma^2. \end{cases}} \qquad (3.16)$$

La fonction objectif $f(x) = x^T A\, x$ est convexe (donc continue). En effet, la matrice hessienne de f est la matrice A qui est semi définie positive. Comme le domaine des solutions réalisables $\Omega_y := \{x \in \mathbb{R}^n \mid \|x - y\|^2 \leq n\sigma^2\}$ est un domaine compact et que f est continue, le minimum du problème (3.16) existe. Il est possible que plusieurs solutions $x^* \in \mathbb{R}^n$ atteignent ce minimum. On verra comment on choisit $\hat{s}$ l'estimé du signal parmi ces solutions x^* dans la section 3.3.

Le problème de minimisation (3.16) peut être vu comme une somme de pénalisations sur certains m-uplets du signal estimé. Supposons qu'on teste, lors de l'analyse statistique préliminaire, l'orthogonalité de chaque couple $(s_i, s_{i+i})^T$ d'un signal s au vecteur $v = (-1, 1)^T$ en utilisant le test statistique de la section 3.1.2. Suite à cette analyse statistique, on aura un ensemble d'indices i pour lesquels on considère plausible que les couples $(s_i, s_{i+1})^T$ soient orthogonaux à v, c'est-à-dire les indices i pour lesquels on a accepté H_0. Ces indices représentent les endroits où on considère plausible que le signal pur s est constant sur deux points consécutifs. Les indices de ces couples se trouvent dans l'ensemble $\{\mathcal{I}_n \setminus \mathcal{I}_v\}$, c'est-à-dire l'ensemble $\mathcal{I}_n$ de tous les indices i moins l'ensemble $\mathcal{I}_v$ des indices des couples pour lesquels on a rejeté l'hypothèse d'orthogonalité au vecteur $v = (-1, 1)^T$. La fonction objectif à minimiser dans ce cas est

$$\begin{aligned} f(x) &= x^T\, A\, x, \\ &= \sum_{i \,\in\, \mathcal{I}_n \setminus \mathcal{I}_v} (\, x_{i+1} - x_i \,)^2 . \end{aligned}$$

Comme (3.16) est un problème de minimisation, la fonction f pénalise les vecteurs x qui s'écartent des modèles locaux qu'on a faits dans l'analyse statistique préliminaire de y. En effet, si i est dans l'ensemble $\{\mathcal{I}_n \setminus \mathcal{I}_v\}$, alors le terme $(\, x_{i+1} - x_i \,)^2$ apparaît dans la fonction f. Or on cherche un estimé $\hat{s}$ pour lequel cette quantité $(\, \hat{s}_{i+1} - \hat{s}_i \,)^2$ sera petite puisqu'on a fait l'hypothèse que le signal pur s était constant à cet endroit. La fonction f tiendra compte simultanément de tous les modèles locaux obtenus dans l'analyse statistique préliminaire de y et produira un estimé $\hat{s}$ faisant un compromis entre ces modèles. Ce compromis est effectué par le biais de la contrainte d'attachement aux données, $\|\hat{s} - y\|^2 \leq n\sigma^2$. Aux endroits où on a fait l'hypothèse que le signal pur s n'était pas constant sur deux points, c'est-à-dire pour les indices $i \in \mathcal{I}_v$, on ne pénalise pas les grandes valeurs de l'estimé $(\, \hat{s}_{i+1} - \hat{s}_i \,)^2$. On peut comparer l'algorithme proposé ici, avec $v = (-1, 1)^T$, avec l'algorithme de variation totale présenté dans la section 2.5. Dans l'équation (2.8), la fonction de régularisation était de la forme

$$\mathcal{J}(x) = \sum_{i=1}^{n-1} | \, x_{i+1} - x_i \, |.$$

La fonction $\mathcal{J}(x)$ ne s'adapte pas au signal bruité y et pénalise tous les couples $(x_i, x_{i+1})^T$. On espère, grâce au modèle composite construit dans la section précédente, gagner en termes de précision de l'estimé par rapport aux méthodes de variation totale. En effet, on tire parti d'une modélisation des caractéristiques locales de s pour construire sur mesure une fonction de pénalisation adaptée à chaque signal bruité y. Le fait d'utiliser davantage d'information devrait se traduire par une augmentation de la précision de l'estimé. En contrepartie, l'analyse statistique préliminaire de y constitue une charge de calcul supplémentaire. On comparera les résultats entre l'algorithme proposé et les méthodes de variation totale dans le chapitre 5.

3.3 Résolution analytique du problème de minimisation

On propose dans cette section un algorithme afin de produire une solution au problème de minimisation (3.16). Cette solution est donnée de manière explicite grâce aux conditions nécessaires d'optimalité de Karush-Kuhn-Tucker. C'est cette solution qui constituera l'estimé $\hat{s}$ du signal pur s.

3.3.1 Conditions nécessaires

On rappelle ici un résultat général sur la résolution d'un problème de minimisation d'une fonction objectif non linéaire sujette à une contrainte d'inégalité non linéaire. Soit le problème d'optimisation général suivant

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} & h(x), \\ \text{s.c.} & g(x) \leq d, \end{cases} \tag{3.17}$$

où $g(x) : \mathbb{R}^n \to \mathbb{R}$ et $h(x) : \mathbb{R}^n \to \mathbb{R}$ sont des fonctions $\mathcal{C}^1$ et d est un nombre réel.

Théorème (Conditions nécessaires de Karush-Kuhn-Tucker (KKT))**.** *Si x^* est une solution du problème (3.17) et que $\nabla g(x^*) \neq 0$, alors il existe un multiplicateur de Lagrange $\lambda^* \in \mathbb{R}$ tel que*

$$\begin{aligned} \nabla h(x^*) &= \lambda^* \nabla g(x^*), \\ g(x^*) &\leq d, \\ \lambda^* (g(x^*) - d) &= 0, \\ \lambda^* &\leq 0. \end{aligned}$$

La démonstration de ce résultat peut se trouver dans plusieurs livres d'optimisation non linéaire, par exemple dans Bazaraa et al. (2006).

On revient maintenant au cas particulier du problème de minimisation (3.16). Étant donné $y \in \mathbb{R}^n$, on définit les fonctions $h(x)$ et $g(x)$ ainsi que $d \in \mathbb{R}$ comme suit :

$$\begin{aligned} h(x) &:= f(x), \\ g(x) &:= \|x - y\|^2, \\ d &:= n\,\sigma^2. \end{aligned}$$

Par les conditions nécessaires de KKT données plus haut, on doit donc trouver $x \in \mathbb{R}^n$ et $\lambda \in \mathbb{R}$ tel que

$$A\,x = \lambda\,(x - y), \tag{3.18}$$

$$\|x - y\|^2 \leq n\,\sigma^2, \tag{3.19}$$

$$\lambda\,(\|x - y\|^2 - n\,\sigma^2) = 0, \tag{3.20}$$

$$\lambda \leq 0. \tag{3.21}$$

La condition (3.20) se nomme la condition de complémentarité. On pourra séparer la résolution du problème en deux cas, soit le cas où $\lambda = 0$ et le cas où $\lambda < 0$. Ce dernier cas correspond au fait que la contrainte est active et la solution sera située sur la frontière $g(x) = d$. Avant d'aller plus loin dans la résolution de ces deux cas, il est utile de faire quelques remarques sur la matrice A.

3.3.2 Remarques sur la matrice A

La matrice $A = D^T D \in \mathbb{R}^{n \times n}$ est symétrique. Cette caractéristique nous assure par le théorème spectral qu'il existe une base orthonormale de vecteurs propres de A (Friedberg et al., 1979). On note $\mathscr{B}$ une telle base, $\mathscr{B} := \{\varphi_1, \varphi_2, ..., \varphi_n\}$. On suppose que les vecteurs $\varphi_i \in \mathbb{R}^n$ de $\mathscr{B}$ sont classés en ordre décroissant selon leur valeur propre μ_i associé, c'est-à-dire $\mu_1 \geq \mu_2 \geq ... \geq \mu_n$. Comme la matrice A est semi définie positive, ses valeurs propres sont toutes plus grandes ou égales à 0. On a donc que $\mu_n \geq 0$. Le prochain résultat facilitera la

résolution de (3.16).

Lemme. *La matrice A possède toujours au moins une valeur propre nulle par construction, donc $\mu_n = 0$.*

Démonstration. Les vecteurs v qu'on utilise dans la construction des matrices circulantes H_v ont la propriété d'être orthogonaux à des vecteurs qui sont des polynômes échantillonnés. Un polynôme de degré 0 (une fonction constante) est un cas particulier des polynômes de degrés supérieurs. Toutes les parties de taille m d'un vecteur constant $u = (1, 1, ..., 1)^T \in \mathbb{R}^n$ peuvent donc être considérées comme des polynômes échantillonnés de degré quelconque. Ainsi tous les m-uplets du vecteur constant u seront orthogonaux à un vecteur $v \in \mathbb{R}^m$ donné puisque ces m-uplets sont tous à la fois constants, linéaires, quadratiques, etc. Le vecteur constant u est donc dans le noyau des matrices H_v pour tous les vecteurs v qu'on a construits dans la section 3.1.1. Lorsqu'on construit les matrices D_v, on retire éventuellement des lignes aux matrices H_v correspondantes. Le vecteur u demeure donc dans le noyau des matrices D_v. Comme le modèle global D est une superposition des matrices D_v, u est également dans le noyau de D. Un élément du noyau de D est aussi un élément du noyau de A :

$$u \in \ker D \implies D^T D\, u = 0 \implies u \in \ker A.$$

Comme il existe un vecteur $u \in \mathbb{R}^n$ non trivial tel que $A\, u = 0$, A possède au moins une valeur propre nulle et $\mu_n = 0$. □

On note dans la suite par l'entier $k \in \{1, .., n-1\}$ l'indice jusqu'auquel les valeurs propres μ_i sont non nulles. Autrement dit $\mu_i > 0$ si $i \leq k$ et $\mu_{k+1} = \mu_{k+2} = ... = \mu_n = 0$. Comme $\mathscr{B}$ est une base de $\mathbb{R}^n$, tout vecteur $u \in \mathbb{R}^n$ peut s'écrire comme une combinaison linéaire des éléments de $\mathscr{B}$. On note par $\tilde{u}_i$ les composantes d'un vecteur u dans la base $\mathscr{B}$:

$$u = \sum_{i=1}^{n} \tilde{u}_i\, \varphi_i. \tag{3.22}$$

Comme $\mathscr{B}$ est une base orthonormale, on a que pour tout vecteur $u \in \mathbb{R}^n$,

$$\|u\|^2 = \sum_{i=1}^{n} \tilde{u}_i^2. \tag{3.23}$$

Si un vecteur $u \in \mathbb{R}^n$ est un élément du noyau de A, alors u est une combinaison linéaire des vecteurs propres φ_i dont la valeur propre μ_i est nulle, autrement dit $\tilde{u}_i = 0$ si $i \leq k$. On a

$$u \in \ker A \implies u = \sum_{i=k+1}^{n} \tilde{u}_i \, \varphi_i. \tag{3.24}$$

3.3.3 Résolution des conditions de KKT

On cherche maintenant des solutions aux conditions de KKT (3.18) à (3.21). On sépare la résolution des conditions en deux : le cas $\lambda = 0$ et le cas $\lambda < 0$.

Cas 1 : $\lambda = 0$

Si $\lambda = 0$, la condition (3.18) implique que

$$A\,x = 0, \tag{3.25}$$

donc $x \in \ker A$. D'après la remarque (3.24), on doit avoir

$$x = \sum_{i=k+1}^{n} \tilde{x}_i \, \varphi_i. \tag{3.26}$$

En substituant l'expression de x en (3.26) dans la condition (3.19) on obtient

$$\begin{aligned} \|x - y\|^2 \leq n\sigma^2 &\iff \sum_{i=1}^{n} (\tilde{x}_i - \tilde{y}_i)^2 \leq n\,\sigma^2 \\ &\iff \sum_{i=1}^{k} \tilde{y}_i^2 + \sum_{i=k+1}^{n} (\tilde{x}_i - \tilde{y}_i)^2 \leq n\,\sigma^2. \end{aligned} \tag{3.27}$$

Le deuxième terme dans le membre de gauche de (3.27) est positif. Pour que cette inégalité puisse être satisfaite, on doit donc avoir

$$\sum_{i=1}^{k} \tilde{y}_i^2 \leq n\,\sigma^2. \tag{3.28}$$

Si (3.28) n'est pas satisfait, il est impossible que (3.27) soit satisfait, et donc il est impossible que $\lambda = 0$. Lorsque (3.28) est vrai, x satisfait les conditions (3.18) à (3.21) peu importe ses composantes $\tilde{x}_i$ dans la base $\mathscr{B}$ pour $i > k$, tant que celles-ci vérifient l'inégalité (3.27). On propose de faire le choix suivant pour ces composantes :

$$\tilde{x}_i = \tilde{y}_i, \quad \text{pour } i = k+1, ..., n. \tag{3.29}$$

On remarque qu'avec ce choix, si (3.28) est vrai, alors (3.27) est vrai. L'estimé $\hat{s}$ proposé dans le cas $\lambda = 0$ est donc, avec (3.26) et le choix (3.29), donné par :

$$\hat{s} = \sum_{i=k+1}^{n} \tilde{y}_i \, \varphi_i. \tag{3.30}$$

Comme cet estimé $\hat{s}$ est dans le noyau de A, $\hat{s}$ atteint le minimum global de la fonction $f(x) = \frac{1}{2} x^T A \, x$ puisque ce minimum est 0. On est donc assuré que $\hat{s}$ est bien une solution du problème de minimisation (3.16).

Cette solution $\hat{s}$ est la projection du vecteur y (le signal bruité) sur l'espace engendré par les vecteurs propres φ_i de A dont la valeur propre est nulle. Cet espace correspond en fait au noyau de A. On rappelle que le modèle composite construit dans la section 3.1 peut être vu comme la recherche d'une application D pour laquelle le signal pur s serait un élément de son noyau. On fait l'hypothèse qu'on réussit à construire une telle matrice D et qu'on a bien $s \in \ker D$. Si s est dans le noyau de D, alors s est dans le noyau de A car $A = D^T D$. Le signal s peut donc être exprimé comme une combinaison linéaire des vecteurs propres φ_i dont la valeur propre est nulle (remarque 3.24). Par la modélisation de la dégradation de s par le bruit blanc w on obtient dans ce cas :

$$\begin{aligned}
y = s + w \iff & \sum_{i=1}^{n} \tilde{y}_i \, \varphi_i = \sum_{i=1}^{n} (\tilde{s}_i + \tilde{w}_i) \, \varphi_i \\
\iff & \sum_{i=1}^{k} \tilde{y}_i \, \varphi_i + \sum_{i=k+1}^{n} \tilde{y}_i \, \varphi_i = \sum_{i=1}^{k} (\tilde{s}_i + \tilde{w}_i) \, \varphi_i + \sum_{i=k+1}^{n} (\tilde{s}_i + \tilde{w}_i) \, \varphi_i \\
\iff & \sum_{i=1}^{k} \tilde{y}_i \, \varphi_i + \sum_{i=k+1}^{n} \tilde{y}_i \, \varphi_i = \sum_{i=1}^{k} \tilde{w}_i \, \varphi_i + \sum_{i=k+1}^{n} (\tilde{s}_i + \tilde{w}_i) \, \varphi_i \\
\implies & \tilde{y}_i = \tilde{w}_i \quad \text{pour } i = 1, ..., k.
\end{aligned} \tag{3.31}$$

Ainsi, les composantes du signal bruité $\tilde{y}_i$ pour $i \leq k$ sont égales à des composantes du bruit dans la base $\mathscr{B}$. Il n'y a pas d'information quant au signal pur s dans ces composantes lorsque $s \in \ker D$. La solution $\hat{s}$ proposée en (3.30) met à 0 toutes les composantes $\tilde{y}_i$ pour $i \leq k$, c'est-à-dire les composantes de y qui ne sont pas dans le noyau de A. Par cette approche, on enlève *seulement* du bruit. Lorsque k est près de n, c'est-à-dire lorsque peu de valeurs propres de A sont nulles et donc que la dimension de $\ker A$ est petite (ce qui sera le cas en pratique), on peut voir la base $\mathscr{B}$ comme une base dans laquelle le signal pur s possède une représentation creuse. Toute l'information de s est concentrée dans les composantes du noyau de A, celles associées aux vecteurs φ_i pour i allant de $k+1$ à n.

Le fait d'exprimer le signal bruité y dans la base $\mathscr{B}$ et de mettre à 0 certaines de ses composantes dans cette base (les coefficients $\tilde{y}_i$ pour i de 1 à k) ressemble aux algorithmes de débruitage par seuillage des coefficients d'ondelettes (voir revue de littérature section 2.4). Contrairement aux algorithmes utilisant la transformée en ondelettes, la base $\mathscr{B}$ dépend dans notre algorithme du signal y puisque la matrice D est construite grâce à l'analyse statistique préliminaire de y. Cette base est donc adaptative et tient compte des discontinuités du signal.

Cas 2 : $\lambda < 0$

Si $\lambda < 0$, à partir de la condition (3.18) et de la représentation de x et y dans la base $\mathscr{B}$, on doit avoir que

$$\begin{aligned}
A\,x = \lambda\,(x-y) &\iff (\lambda\,I - A)\,x = \lambda\;y \\
&\iff (\lambda\,I - A)\sum_{i=1}^{n}\tilde{x}_i\,\varphi_i = \lambda\sum_{i=1}^{n}\tilde{y}_i\,\varphi_i \\
&\iff \sum_{i=1}^{n}\tilde{x}_i\,(\lambda\,I - A)\,\varphi_i = \sum_{i=1}^{n}\lambda\,\tilde{y}_i\,\varphi_i \\
&\iff \sum_{i=1}^{n}\tilde{x}_i\,(\lambda - \mu_i)\,\varphi_i = \sum_{i=1}^{n}\lambda\,\tilde{y}_i\,\varphi_i \\
&\iff \tilde{x}_i\,(\lambda - \mu_i) = \lambda\,\tilde{y}_i \quad \text{pour} \quad i = 1, ..., n, \qquad (3.32)
\end{aligned}$$

où $I \in \mathbb{R}^{n\times n}$ est la matrice identité. On rappelle que $\mu_i \geq 0$ est la valeur propre de A associée au vecteur propre φ_i. Comme $\lambda < 0$, on a que $(\lambda - \mu_i) \neq 0$ pour tout i de 1 à n. Par (3.32), on trouve que x doit satisfaire

$$x = \sum_{i=1}^{n}\tilde{x}_i\,\varphi_i = \sum_{i=1}^{n}\frac{\lambda\,\tilde{y}_i}{\lambda - \mu_i}\,\varphi_i. \qquad (3.33)$$

Comme $\lambda < 0$, la condition de complémentarité (3.20) combinée avec l'expression de x en (3.33) donne

$$\begin{aligned}
\|x-y\|^2 = n\,\sigma^2 &\iff \sum_{i=1}^{n}(\tilde{x}_i - \tilde{y}_i)^2 = n\,\sigma^2 \\
&\iff \sum_{i=1}^{n}\left(\frac{\lambda\,\tilde{y}_i}{\lambda - \mu_i} - \tilde{y}_i\right)^2 = n\,\sigma^2 \\
&\iff \sum_{i=1}^{n}\left(\frac{\mu_i\,\tilde{y}_i}{\lambda - \mu_i}\right)^2 = n\,\sigma^2. \qquad (3.34)
\end{aligned}$$

La valeur de λ cherchée est une racine de la fonction $\mathscr{L} : (-\infty, 0) \to \mathbb{R}$ définit par

$$\mathscr{L}(\lambda) := \sum_{i=1}^{n} \left(\frac{\mu_i \tilde{y}_i}{\lambda - \mu_i} \right)^2 - n\,\sigma^2. \tag{3.35}$$

Puisque $\mu_i = 0$ si $i > k$, $\mathscr{L}$ prend la forme

$$\mathscr{L}(\lambda) = \sum_{i=1}^{k} \left(\frac{\mu_i \tilde{y}_i}{\lambda - \mu_i} \right)^2 - n\,\sigma^2. \tag{3.36}$$

Une courte étude de la fonction $\mathscr{L}$ est nécessaire. En dérivant $\mathscr{L}$ on obtient

$$\mathscr{L}'(\lambda) = -2 \sum_{i=1}^{k} \frac{(\mu_i \tilde{y}_i)^2}{(\lambda - \mu_i)^3}. \tag{3.37}$$

Puisque $\mu_i > 0$ si $i \leq k$, on remarque que

$$\mathscr{L}'(\lambda) > 0 \quad \text{lorsque} \quad \lambda < 0. \tag{3.38}$$

On trouve à partir de (3.36) que

$$\lim_{\lambda \to -\infty} \mathscr{L}(\lambda) = -n\,\sigma^2 < 0, \tag{3.39}$$

ainsi que

$$\lim_{\lambda \to 0^-} \mathscr{L}(\lambda) = \sum_{i=1}^{k} \tilde{y}_i^2 - n\,\sigma^2. \tag{3.40}$$

Par les remarques (3.38) à (3.40), on trouve que la fonction $\mathscr{L}$ possède une unique racine sur son domaine $(-\infty, 0)$ si et seulement si

$$\sum_{i=1}^{k} \tilde{y}_i^2 - n\,\sigma^2 > 0. \tag{3.41}$$

Ainsi, si (3.41) n'est pas satisfait, $\mathscr{L}$ ne possède pas de racine sur $(-\infty, 0)$ et il est impossible que $\lambda < 0$. Si (3.41) est vrai, $\mathscr{L}$ possède une unique racine λ^* sur $(-\infty, 0)$. On pose alors l'estimé $\hat{s}$ comme étant la solution obtenue en remplaçant λ par cette racine λ^* dans l'expression (3.33) :

$$\hat{s} = \sum_{i=1}^{n} \frac{\lambda^* \, \tilde{y}_i}{\lambda^* - \mu_i} \, \varphi_i. \tag{3.42}$$

Comme $\mathscr{L}'(\lambda) > 0$ pour $\lambda \in (-\infty, 0\,)$, on peut trouver la racine λ^* de la fonction $\mathscr{L}$ numériquement grâce à la méthode de Newton. Une itération de la méthode de Newton est donnée par

$$\begin{aligned} \lambda_{j+1} &= \lambda_j - \frac{\mathscr{L}(\lambda_j)}{\mathscr{L}'(\lambda_j)}, \\ &= \lambda_j + \frac{\displaystyle\sum_{i=1}^{k} \left(\frac{\mu_i \, \tilde{y}_i}{\lambda_j - \mu_i} \right)^2 - n \, \sigma^2}{2 \displaystyle\sum_{i=1}^{k} \frac{(\mu_i \, \tilde{y}_i)^2}{(\lambda_j - \mu_i)^3}}. \end{aligned} \tag{3.43}$$

En initialisant $j = 0$, on trouve un premier estimé de la racine $\lambda_0 < 0$ de $\mathscr{L}$ par une méthode de bissection. À partir de cet estimé λ_0, on trouve λ_{j+1} grâce à la formule (3.43) et on fixe un critère d'arrêt ε sur l'écart $|\lambda_{j+1} - \lambda_j|$. Lorsque $|\lambda_{j+1} - \lambda_j| \leq \varepsilon$, on pose $\lambda^* = \lambda_{j+1}$. On prendra dans la suite $\varepsilon = 10^{-12}$.

Choix entre le cas 1 et le cas 2

Il est impossible qu'un vecteur $x^* \in \mathbb{R}^n$ et un multiplicateur $\lambda^* \in \mathbb{R}$ satisfaisants les conditions de KKT (3.18) à (3.21) puissent être obtenus simultanément dans les deux cas discutés plus haut. En effet, on a que $\lambda^* = 0$ si et seulement si la condition

$$\sum_{i=1}^{k} \tilde{y}_i^2 \leq n \, \sigma^2 \tag{3.28}$$

est respectée. De plus, $\lambda^* < 0$ si et seulement si la condition

$$\sum_{i=1}^{k} \tilde{y}_i^2 - n \, \sigma^2 > 0 \tag{3.41}$$

est respectée. Il n'y a donc pas d'ambiguïté entre les deux cas possibles et l'estimé $\hat{s}$ pourra toujours être trouvée dans l'un ou l'autre de ces deux cas puisqu'il existe toujours une solution au problème de minimisation (3.16).

Pseudo code pour la résolution du problème de minimisation

On présente dans l'algorithme 2 le pseudo code utilisé afin de trouver une solution au problème de minimisation (3.16). Cette solution correspond à notre estimé $\hat{s}$. Cet algorithme prend en entrée la matrice D construite suite à l'analyse statistique préliminaire d'un signal bruité y.

Algorithme 2 : Résolution du problème d'optimisation

Entrée : $y \in \mathbb{R}^n$, $\sigma \in \mathbb{R}$, $D \in \mathbb{R}^{l \times n}$.
Sortie : $\hat{s} \in \mathbb{R}^n$.

I.
1. Calculer $A = D^T D \in \mathbb{R}^{n \times n}$.
2. Calculer $\boldsymbol{\mu} := (\mu_1, ..., \mu_n)^T \in \mathbb{R}^n$ les valeurs propres de A, en ordre décroissant.
3. Calculer $B := \begin{pmatrix} \varphi_1^T \\ \vdots \\ \varphi_n^T \end{pmatrix} \in \mathbb{R}^{n \times n}$ les vecteurs propres orthonormaux de A.
4. Calculer $k \in \{1, ..., n-1\}$ indice jusqu'auquel μ_i est non nul.

II.
1. Calculer $\tilde{y} = B \cdot y \in \mathbb{R}^n$ les composantes de y dans la base $\mathscr{B}$.
2. Calculer $\text{Cas}_{1,2} = \sum_{i=1}^{k} \tilde{y}_i^2 - n\,\sigma^2 \in \mathbb{R}$.

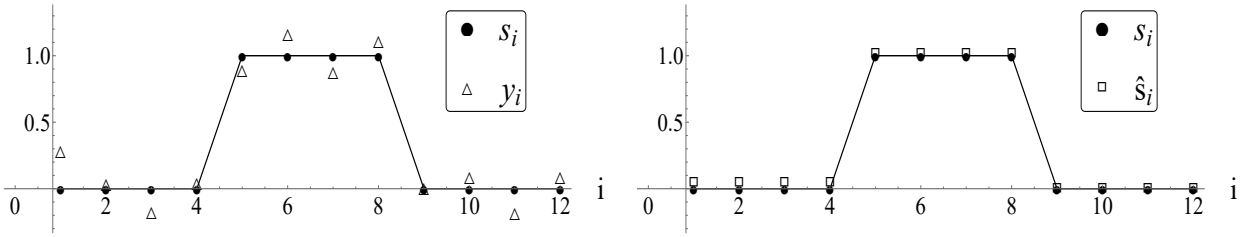
III. Si $\text{Cas}_{1,2} \leq 0$ (Cas $\lambda = 0$)
Calculer $\hat{s} = \sum_{i=k+1}^{n} \tilde{y}_i\, \varphi_i$.
Sinon (Cas $\lambda < 0$)
Définir $\mathscr{L}(\lambda) := \sum_{i=1}^{k} \left(\frac{\mu_i\, \tilde{y}_i}{\lambda - \mu_i} \right)^2 - n\,\sigma^2$.
Trouver la racine λ^* de $\mathscr{L}$.
Calculer $\hat{s} = \sum_{i=1}^{n} \frac{\lambda^* \tilde{y}_i}{\lambda^* - \mu_i}\, \varphi_i$.

IV. Fin.

3.3.4 Retour aux exemples

On a présenté dans la section 3.1.2 deux exemples pour lesquels on a construit, grâce à l'analyse statistique préliminaire de y, la matrice D représentant notre modèle composite de s.

On montre dans la figure 3.4a le signal pur et bruité de l'exemple 1. On montre dans la figure 3.4b l'estimé $\hat{s}$ produit par notre algorithme en utilisant la matrice D construite suite à l'analyse statistique préliminaire.



(a) Signal pur s et bruité y, $\sigma = 0.15$.

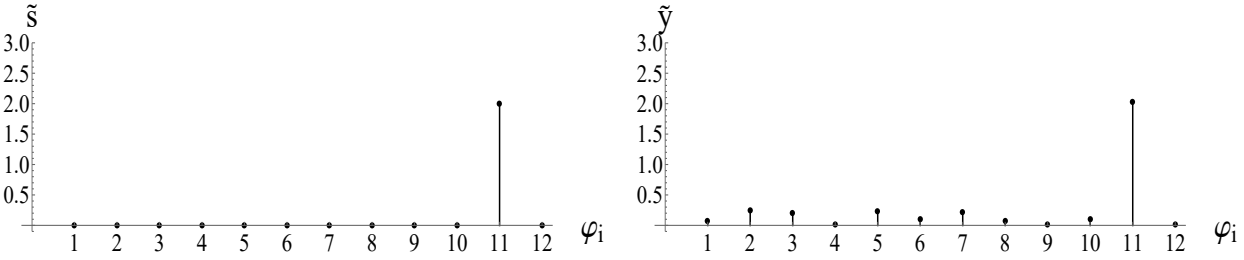
(b) Signal pur s et estimé $\hat{s}$.

Figure 3.4 Signal estimé, exemple 1.

L'estimé $\hat{s}$ dans la figure 3.4b est très proche du signal pur s. On donnera un critère objectif afin de quantifier la précision d'un estimé dans le chapitre 5. Le nombre de valeurs propres non nulles de $A = D^T D$ dans ce cas est $k = 9$. La matrice A possède donc $n - k = 3$ valeurs propres nulles. On a obtenu dans cet exemple que

$$\sum_{i=1}^{9} \tilde{y}_i^2 - n\sigma^2 \approx 0.220 - 0.27 = -0.05 < 0.$$

On a donc résolu le problème de minimisation par le cas $\lambda = 0$ et la solution $\hat{s}$ est la projection de y sur le noyau de A. On a déjà montré dans la section 3.1.4 que la matrice D dans cet exemple avait bien la propriété $s \in \ker D$. Ainsi, les coefficients $\tilde{s}_i$ de s dans la base $\mathscr{B}$ devraient être nuls pour $i \leq 9$. On montre dans la figure 3.5 la valeur des coefficients $\tilde{s}_i$ et $\tilde{y}_i$ (en valeur absolue) de s et y dans la base $\mathscr{B}$.



(a) Coefficients de s dans la base $\mathscr{B}$.

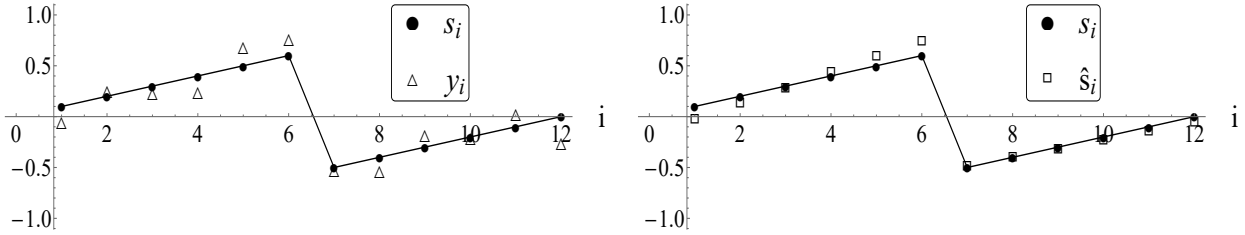
(b) Coefficients de y dans la base $\mathscr{B}$.

Figure 3.5 Coefficients de s et y dans la base $\mathscr{B}$, exemple 1.

On voit dans la figure 3.5a que $\tilde{s}_i = 0$ pour $i \leq 9$. En fait, une seule composante de s est non nulle dans la base $\mathscr{B}$, à savoir $\tilde{s}_{11}$, celle associée au vecteur propre φ_{11}. Le signal s possède donc une représentation creuse dans la base $\mathscr{B}$. En mettant à 0 tous les coefficients $\tilde{y}_i$ de y

pour $i \leq 9$, le signal estimé $\hat{s}$ se rapproche de la représentation du signal pur s dans la base $\mathscr{B}$. C'est la raison pour laquelle l'algorithme semble effectivement avoir fait un travail de débruitage dans la figure 3.4b. Dans cet exemple, la matrice D est construite en testant s'il est plausible que le signal pur soit constant sur deux points consécutifs. L'analyse statistique préliminaire a permis d'obtenir une base $\mathscr{B}$ bien adaptée aux caractéristiques locales du signal. On retrouve bien les trois paliers du signal pur s dans le signal estimé $\hat{s}$, sans erreur au niveau des discontinuités, situées entre s_4 et s_5 et entre s_8 et s_9.

On montre dans la figure 3.6a le signal pur et bruité de l'exemple 2. On montre dans la figure 3.6b l'estimé $\hat{s}$ produit par notre algorithme en utilisant la matrice D construite suite à l'analyse statistique préliminaire.



(a) Signal pur s et bruité y, $\sigma = 0.15$.

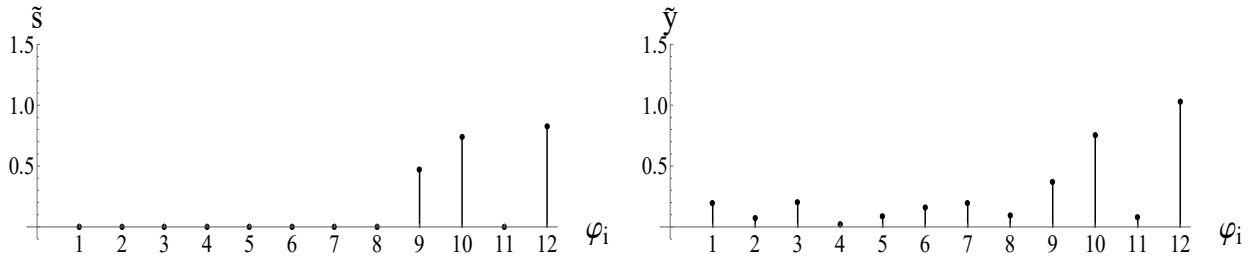
(b) Signal pur s et estimé $\hat{s}$.

Figure 3.6 Signal estimé, exemple 2.

Cette fois encore, le signal estimé $\hat{s}$ dans la figure 3.6b est proche du signal pur s. On a obtenu dans ce cas que A possède $k = 8$ valeurs propres non nulles et donc $n - k = 4$ valeurs propres nulles. On a obtenu dans cet exemple que

$$\sum_{i=1}^{8} \tilde{y}_i^2 - n\sigma^2 \approx 0.167 - 0.27 = -0.103 < 0.$$

On a donc résolu le problème de minimisation par le cas $\lambda = 0$ et la solution $\hat{s}$ est la projection de y sur le noyau de A. On montre dans la figure 3.7 les coefficients s et y dans la base $\mathscr{B}$. Comme prévu, les coefficients $\tilde{s}_i$ de s sont nuls pour $i \leq 8$ puisqu'on avait dans cet exemple aussi la relation $s \in \ker D$. On effectue donc un travail de débruitage en mettant à 0 les coefficients $\tilde{y}_i$ correspondants dans l'estimé de $\hat{s}$. Les deux parties linéaires du signal pur sont bien préservées dans l'estimé $\hat{s}$ (figure 3.6b).



(a) Coefficients de s dans la base $\mathscr{B}$.

(b) Coefficients de y dans la base $\mathscr{B}$.

Figure 3.7 Coefficients de s et y dans la base $\mathscr{B}$, exemple 2.

On présente maintenant un troisième exemple où l'estimé $\hat{s}$ est trouvé en résolvant le problème d'optimisation à partir du cas $\lambda < 0$. On utilise pour cet exemple le même signal pur s et bruité $y \in \mathbb{R}^{12}$ que l'exemple 1, voir figure 3.4a. En plus de tester l'orthogonalité de chaque couple de s au vecteur $v_1 := (-1, 1)^T$, on teste également l'orthogonalité de chaque triplet de s au vecteur $v_2 := (1, -2, 1)^T$. On rejette l'hypothèse d'orthogonalité entre les triplets de s et ce vecteur v selon un seuil α de 5 % :

$$\text{si} \quad \left| [H_{v_2}\, y]_i \right| \;\geq\; \delta_\alpha \, \|v_2\| \, \sigma \;=\; 1.96 \times \sqrt{6} \times 0.15 \;\approx\; 0.720, \quad \text{on rejette } \mathrm{H}_0.$$

Dans le tableau 3.4, on montre la valeur absolue du produit scalaire entre chaque triplet $(y_i, y_{i+1}, y_{i+2})^T$ de y et v_2. On montre dans la colonne de droite la décision prise selon le seuil de 5 %, à savoir si on accepte ou non l'hypothèse d'orthogonalité H_0 pour chacun de ces triplets.

Tableau 3.4 Tests d'orthogonalité, exemple 3.

Indice i	$\lvert [H_{v_2}\, y]_i \rvert$	Accepte H_0
1	0.045	Oui
2	0.427	Oui
3	0.628	Oui
4	0.576	Oui
5	0.553	Oui
6	0.510	Oui
7	1.332	Non
8	1.187	Non
9	0.353	Oui
10	0.543	Oui

Selon ce test, on rejette l'hypothèse d'orthogonalité au vecteur $v_2 = (1, -2, 1)^T$ pour les

triplets $(s_7, s_8, s_9)^T$ et $(s_8, s_9, s_{10})^T$. On fait donc l'hypothèse suite à cette analyse statistique que les triplets du signal pur $(s_3, s_4, s_5)^T$ et $(s_4, s_5, s_6)^T$ sont linéaires, ce qui n'est pas le cas (voir figure 3.4a). La matrice D_{v_2} construite à partir de ce test n'a pas la propriété $s \in \ker D_{v_2}$:

$$D_{v_2}\, s = \begin{pmatrix} 1 & -2 & 1 & & & & & & & & \\ & 1 & -2 & 1 & & & & & & & \\ & & 1 & -2 & 1 & & & & & & \\ & & & 1 & -2 & 1 & & & & & \\ & & & & 1 & -2 & 1 & & & & \\ & & & & & 1 & -2 & 1 & & & \\ & & (0) & & & & & 1 & -2 & 1 & \\ & & & & & & & & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

La matrice D représentant le modèle composite du signal s est la superposition de la matrice D_{v_1} qu'on a obtenue dans l'exemple 1 et la matrice D_{v_2} obtenue dans ce troisième exemple (voir définition (3.12)). On montre dans la figure 3.8 l'estimé $\hat{s}$ produit avec cette matrice D.

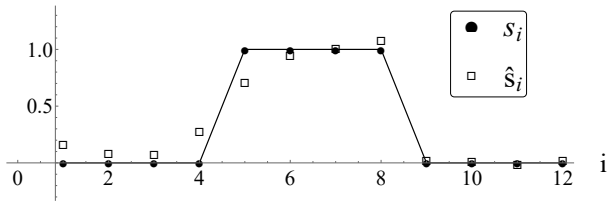
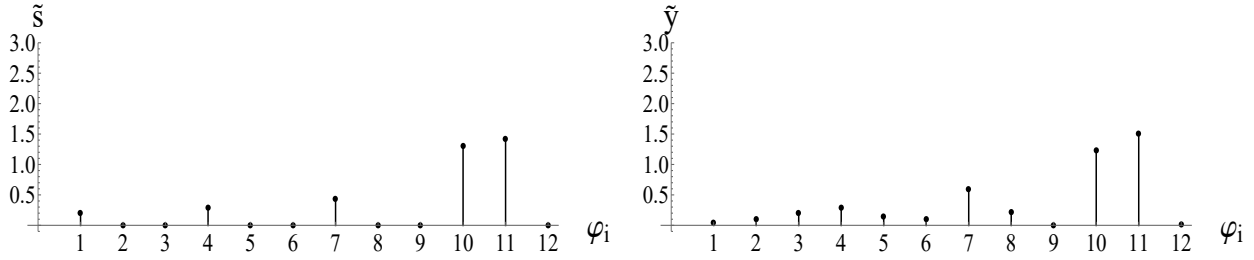


Figure 3.8 Signal estimé $\hat{s}$, exemple 3

Cette fois, l'estimé $\hat{s}$ ne préserve par aussi bien la discontinuité du signal pur entre s_4 et s_5 que l'estimé de l'exemple 1 (voir figure 3.4b). Ceci est dû au fait que le problème de minimisation a contraint les triplets $(\hat{s}_3, \hat{s}_4, \hat{s}_5)^T$ et $(\hat{s}_4, \hat{s}_5, \hat{s}_6)^T$ à ne pas s'écarter de la linéarité. La matrice A possède $k = 10$ valeurs propres non nulles et donc $n - k = 2$ valeurs propres nulles. On a obtenu dans cet exemple que

$$\sum_{i=1}^{10} \tilde{y}_i^2 - n\sigma^2 \approx 2.075 - 0.27 = 1.805 > 0.$$

On a donc résolu le problème de minimisation par le cas $\lambda < 0$. Comme D n'a pas la propriété $s \in \ker D$, les composantes $\tilde{s}_i$ de s dans la base $\mathscr{B}$ des vecteurs propres de A ne sont pas nécessairement nulles pour $i \leq k$. On montre dans la figure 3.9 la valeur absolue des coefficients de s et y dans la base $\mathscr{B}$.



(a) Coefficients de s dans la base $\mathscr{B}$.

(b) Coefficients de y dans la base $\mathscr{B}$.

Figure 3.9 Coefficients de s et y dans la base $\mathscr{B}$, exemple 3.

On voit dans la figure 3.9a que les composantes $\tilde{s}_i$ de s dans la base $\mathscr{B}$ ne sont pas nulles pour $i \leq k$. La base $\mathscr{B}$ est moins bien adaptée au signal s que dans les deux premiers exemples dans le sens où la représentation de s dans cette base n'est pas creuse.

3.3.5 Modifications mineures de l'algorithme

L'estimé produit par notre algorithme tel que présenté dans les sections précédentes présente parfois des artefacts indésirables qu'il est possible d'éviter grâce à des modifications mineures de l'algorithme. On présente ces modifications afin d'améliorer l'estimé $\hat{s}$ dans le cas où le signal pur s est constant.

Utilisation de plusieurs vecteurs v_k

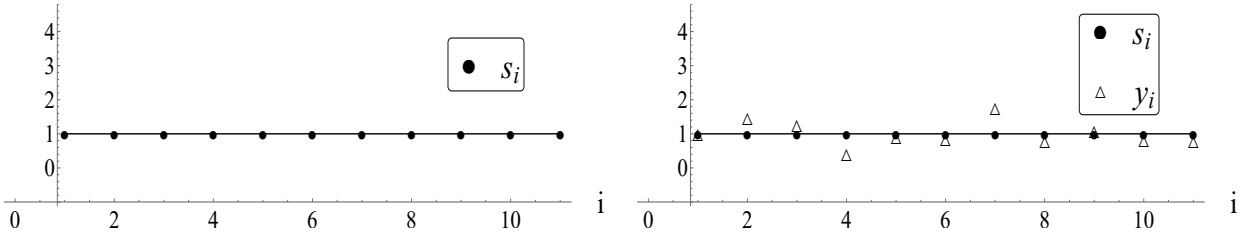
On définit dans cette partie le vecteur v_k pour $k = 1, 2, 3, ...$ comme suit :

$$v_k := (-1, \underbrace{0, ..., 0,}_{k-1} 1)^T \in \mathbb{R}^{k+1}. \tag{3.44}$$

Un vecteur v_k est donc de dimension $m_k := k + 1$. Avec cette définition, on a $v_1 = (-1, 1)^T$. On a déjà introduit ces vecteurs v_k dans la section 3.1.1. Ceux-ci serviront à déterminer les parties constantes d'un signal.

Les artefacts apparaissant parfois dans l'estimé ainsi que notre façon de les éviter sont présentés à partir d'exemples. On présente dans la figure 3.10a un signal constant $s \in \mathbb{R}^{11}$.

Ce signal est dégradé dans la figure 3.10b par un bruit blanc w d'écart-type $\sigma = 0.3$ et de moyenne nulle.



(a) Signal constant $s \in \mathbb{R}^{11}$.

(b) Signal bruité y, $\sigma = 0.3$.

Figure 3.10 Signal constant pur et bruité.

On effectue l'analyse statistique préliminaire du signal bruité y de la figure 3.10b avec le vecteur $v_1 = (-1, 1)^T$ et un seuil α de 0.05. On montre dans le tableau 3.5 les résultats de cette analyse statistique préliminaire. On rejette l'hypothèse d'orthogonalité au vecteur v_1 lorsque le produit scalaire entre celui-ci et $(y_i, y_{i+1})^T$ est plus grand que 0.832.

Tableau 3.5 Tests d'orthogonalité avec v_1.

Indice i	$\lvert y_{i+1} - y_i \rvert$	Accepte H_0
1	0.481	Oui
2	0.210	Oui
3	0.867	Non
4	0.524	Oui
5	0.068	Oui
6	0.903	Non
7	0.958	Non
8	0.276	Oui
9	0.253	Oui
10	0.025	Oui

Pour les couples $(y_i, y_{i+1})^T$ d'indices i qui n'ont pas passé le test d'orthogonalité, on fait l'hypothèse que les couples du signal pur s correspondants n'étaient pas constants à ces endroits. On ne pénalisera pas dans le problème d'optimisation les écarts entre les valeurs $\hat{s}_i$ et $\hat{s}_{i+1}$ pour ces indices i. Il s'agit dans cet exemple d'erreurs statistiques, c'est-à-dire qu'on a rejeté H_0 alors que H_0 était vrai. On présente dans la figure 3.11a l'estimé $\hat{s}$ produit à partir des résultats de l'analyse statistique préliminaire du tableau 3.5. Comme les écarts entre les valeurs des couples $(\hat{s}_3, \hat{s}_4)^T$, $(\hat{s}_6, \hat{s}_7)^T$ et $(\hat{s}_7, \hat{s}_8)^T$ ne sont pas pénalisés, le signal

estimé possède des plateaux et une valeur isolée qui ne sont pas présents dans le signal pur s de la figure 3.10a.

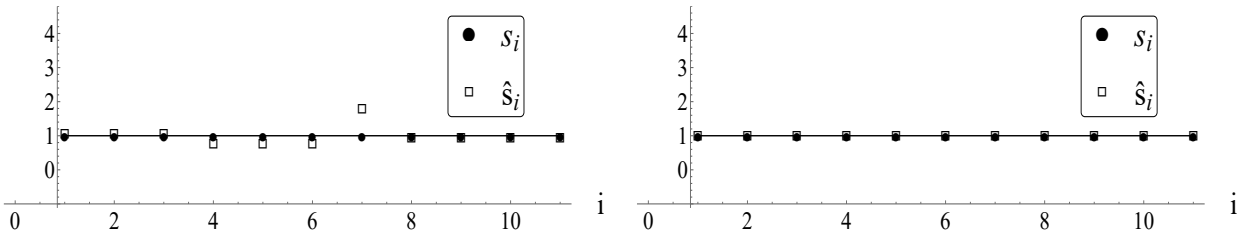
Une manière d'éviter ces artefacts dans l'estimé est d'utiliser plusieurs vecteurs v_k associés aux parties constantes d'un signal dans l'analyse statistique préliminaire. On montre dans le tableau 3.6 les résultats de l'analyse statistique préliminaire de y effectuée avec les vecteurs v_1, v_2 et v_3, définis en (3.44) toujours avec un seuil α de 0.05.

Tableau 3.6 Tests d'orthogonalité avec v_1, v_2 et v_3.

Indice i	$\lvert y_{i+1} - y_i \rvert$	Accepte H_0	$\lvert y_{i+2} - y_i \rvert$	Accepte H_0	$\lvert y_{i+3} - y_i \rvert$	Accepte H_0
1	0.481	Oui	0.271	Oui	0.596	Oui
2	0.210	Oui	1.078	Non	0.553	Oui
3	0.867	Non	0.343	Oui	0.411	Oui
4	0.524	Oui	0.456	Oui	1.359	Non
5	0.068	Oui	0.834	Non	0.124	Oui
6	0.903	Non	0.055	Oui	0.221	Oui
7	0.958	Non	0.682	Oui	0.935	Non
8	0.276	Oui	0.023	Oui	0.002	Oui
9	0.253	Oui	0.278	Oui		-
10	0.025	Oui		-		-

En effectuant plusieurs tests statistiques avec des vecteurs v_k de dimensions différentes, on diminue le pourcentage de tests statistiques pour lesquelles on fait des erreurs, c'est-à-dire de rejeter H_0 alors que H_0 est vrai. De plus, certains tests statistiques avec un vecteur v_{k+1} permettent de diminuer l'influence d'une erreur faite dans un test statistique avec un vecteur v_k. Par exemple, le couple $(y_7, y_8)^T$ échoue le test d'orthogonalité avec v_1. Cependant, les triplets $(y_6, y_7, y_8)^T$ et $(y_7, y_8, y_9)^T$ ont tous deux passé le test d'orthogonalité avec v_2. L'algorithme pénalise donc les écarts entre les valeurs de l'estimé $\hat{s}_6$ et $\hat{s}_8$ et entre les valeurs de l'estimé $\hat{s}_7$ et $\hat{s}_9$. Ceci pénalise indirectement les écarts entre $\hat{s}_7$ et $\hat{s}_8$. On présente dans la figure 3.11b l'estimé $\hat{s}$ produit à partir des résultats de l'analyse statistique préliminaire du tableau 3.6. La matrice D utilisée dans ce cas est la superposition des matrices D_{v_1}, D_{v_2} et D_{v_3}.

En comparant l'estimé $\hat{s}$ produit produit à partir des tests statistiques avec v_1 de la figure 3.11a et celui produit à partir des tests avec v_1, v_2 et v_3 de la figure 3.11b, on remarque que ce dernier estimé est plus proche du signal pur s. Les plateaux et la valeur isolée présents dans l'estimé de la figure 3.11a n'apparaissent pas dans l'estimé de la figure 3.11b. On a utilisé dans l'analyse statistique préliminaire trois vecteurs v_k. Le nombre de vecteurs v_k qu'on utilise est un paramètre à ajuster. On discutera du choix de ce paramètre dans le chapitre 5.



(a) Signal estimé, utilisation de v_1. (b) Signal estimé, utilisation de v_1 à v_3.

Figure 3.11 Faiblesse et amélioration de l'algorithme.

Traitement des résultats de l'analyse statistique préliminaire pour des signaux constants par morceaux

On fera dans cette section l'hypothèse que les signaux dans leur version continue sont des signaux constants par morceaux dont les morceaux sont de mesure non nulle. De plus, on suppose que l'échantillonnage de ces signaux est assez fin pour assurer que chaque morceau constant du signal discret contient au moins trois points.

On présente dans la figure 3.12 le même signal pur $s \in \mathbb{R}^{11}$ constant que pour l'exemple précédent dégradé par une deuxième réalisation du bruit w d'écart-type $\sigma = 0.3$. On trouve un deuxième signal bruité y.

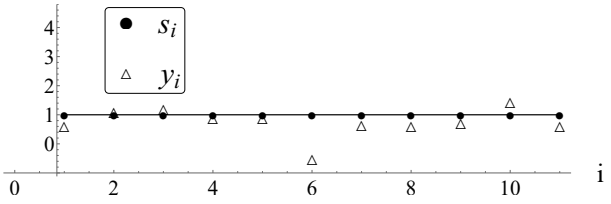


Figure 3.12 Signal bruité y, $\sigma = 0.3$, deuxième réalisation.

L'observation y_6 du signal bruité de la figure 3.12 est très éloignée des autres observations dans son voisinage. Ceci est dû à une grande valeur de la composante w_6 du bruit ajouté. Le tableau 3.7 présente les résultats de l'analyse statistique préliminaire de ce signal y avec les vecteurs v_1 à v_3 et le seuil $\alpha = 0.05$. Comme l'observation y_6 est très éloignée des autres observations dans son voisinage, on constate dans le tableau 3.7 qu'on rejette l'hypothèse d'orthogonalité pour tous les m_k-uplets qui contiennent cette observation. On ne pénalise donc pas les écarts entre les valeurs de ces m_k-uplets dans le signal estimé $\hat{s}$. Ceci fait en sorte que l'estimé $\hat{s}$ contient une valeur isolée en $\hat{s}_6$. La figure 3.13a présente l'estimé $\hat{s}$ produit

suite à l'analyse statistique préliminaire du tableau 3.7.

Tableau 3.7 Tests d'orthogonalité avec v_1, v_2 et v_3, deuxième réalisation.

Indice i	$\lvert y_{i+1} - y_i \rvert$	Accepte H_0	$\lvert y_{i+2} - y_i \rvert$	Accepte H_0	$\lvert y_{i+3} - y_i \rvert$	Accepte H_0
1	0.483	Oui	0.589	Oui	0.288	Oui
2	0.106	Oui	0.195	Oui	0.185	Oui
3	0.301	Oui	0.292	Oui	1.699	Non
4	0.010	Oui	1.398	Non	0.252	Oui
5	1.408	Non	0.261	Oui	0.277	Oui
6	1.146	Non	1.131	Non	1.220	Non
7	0.015	Oui	0.073	Oui	0.797	Oui
8	0.089	Oui	0.815	Oui	0.022	Oui
9	0.724	Oui	0.111	Oui	-	-
10	0.835	Oui	-	-	-	-

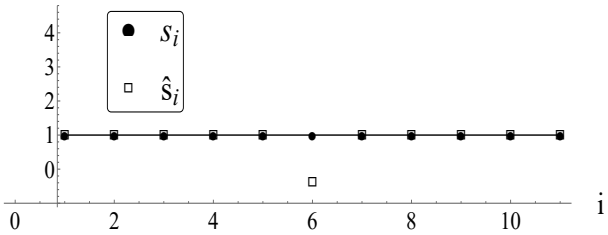
Pour éviter ce type d'artefact dû à une valeur élevée d'une composante du bruit ajouté w, on peut procéder à un traitement des résultats des tests statistiques. Ce traitement se base sur l'hypothèse que les paliers du signal pur s constant par morceaux contiennent au moins 3 points. On peut, à partir de cette hypothèse, changer les résultats des tests d'orthogonalité effectués lors de l'analyse statistique préliminaire de y afin qu'il n'y ait pas d'incohérences. On procède de la manière suivante. Lorsqu'on rejette l'hypothèse d'orthogonalité entre un couple $(s_i, s_{i+1})^T$ et le vecteur v_1, on devrait rejeter également l'hypothèse d'orthogonalité entre les triplets $(s_{i-1}, s_i, s_{i+1})^T$ et $(s_i, s_{i+1}, s_{i+2})^T$ et v_2 puisque le signal pur s ne possède pas de paliers de seulement un point par hypothèse. Si une de ces deux hypothèses n'est pas rejetée, on change le résultat du test d'orthogonalité entre $(s_i, s_{i+1})^T$ et le vecteur v_1 et on accepte l'hypothèse d'orthogonalité entre ceux-ci. On considère alors qu'il s'agissait d'une erreur de faire l'hypothèse que le couple du signal pur $(s_i, s_{i+1})^T$ n'était pas constant. On fait de même pour les triplets $(s_i, s_{i+1}, s_{i+2})^T$. Si on rejette l'hypothèse d'orthogonalité entre $(s_i, s_{i+1}, s_{i+2})^T$ et v_2, on devrait également rejeter l'hypothèse d'orthogonalité entre les quadruplets $(s_{i-1}, s_i, s_{i+1}, s_{i+2})^T$ et $(s_i, s_{i+1}, s_{i+2}, s_{i+3})^T$ et le vecteur v_3 puisque le signal s ne possède pas de paliers de seulement deux points par hypothèse. Si une de ces deux hypothèses n'est pas rejetée, on change le résultat du test d'orthogonalité entre $(s_i, s_{i+1}, s_{i+2})^T$ et le vecteur v_2 et on accepte l'hypothèse d'orthogonalité entre ceux-ci.

On présente dans le tableau 3.8 les changements de résultats des tests d'orthogonalité avec v_1 et v_2 (en gras) pour le signal y de la figure 3.12.

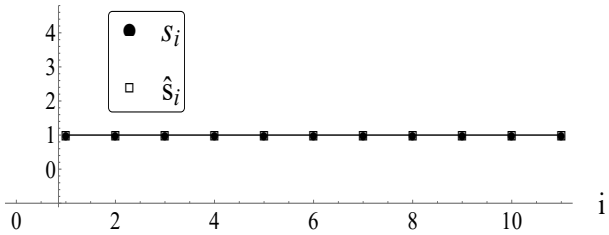
Suite au traitement des résultats de l'analyse statistique préliminaire de y, l'hypothèse d'orthogonalité est acceptée pour tous les m_k-uplets de s pour $k = 1, 2$. On montre dans la figure 3.13 l'estimé $\hat{s}$ produit avec ces nouvelles hypothèses.

Tableau 3.8 Modification des résultats de l'analyse statistique préliminaire.

Indice i	$\lvert y_{i+1} - y_i \rvert$	Accepte H_0	$\lvert y_{i+2} - y_i \rvert$	Accepte H_0	$\lvert y_{i+3} - y_i \rvert$	Accepte H_0
1	0.483	Oui	0.589	Oui	0.288	Oui
2	0.106	Oui	0.195	Oui	0.185	Oui
3	0.301	Oui	0.292	Oui	1.699	Non
4	0.010	Oui	1.398	**Oui**	0.252	Oui
5	1.408	**Oui**	0.261	Oui	0.277	Oui
6	1.146	**Oui**	1.131	**Oui**	1.220	Non
7	0.015	Oui	0.073	Oui	0.797	Oui
8	0.089	Oui	0.815	Oui	0.022	Oui
9	0.724	Oui	0.111	Oui	-	-
10	0.835	Oui	-	-	-	-



(a) Signal estimé sans le traitement de l'analyse statistique préliminaire.



(b) Signal estimé avec le traitement de l'analyse statistique préliminaire.

Figure 3.13 Comparaison des résultats avec et sans le traitement de l'analyse statistique préliminaire.

On pénalise maintenant les écarts entre les valeurs des m_k-uplets qui contiennent $\hat{s}_6$. L'erreur de l'estimé présenté dans la figure 3.13a n'est plus visible dans l'estimé de la figure 3.13b. On présente dans l'algorithme 3 le pseudo code utilisé pour le traitement des résultats de l'analyse statistique préliminaire dans le cas où le signal est constant par morceaux.

Algorithme 3 : Traitement des résultats de l'analyse statistique préliminaire

Soient v_1, v_2, ... , v_M un ensemble de M vecteurs, $v_k \in \mathbb{R}^{m_k}$ pour $k = 1, ..., M$ comme défini en (3.44).
Soient y un signal bruité, σ l'écart-type du bruit et δ_α le facteur associé à un seuil de rejet α.

I. Construire H_{v_k} pour k de 1 à m selon la définition (3.5).

II. Pour $k = 1, ..., m-1$:

Pour $i = 2, ..., n - (m_k - 2)$:

si $\left| \left[H_{v_k}\, y \right]_i \right| \geq \delta_\alpha \left\| v_k \right\| \sigma$,

si $\left| \left[H_{v_{k+1}}\, y \right]_{i-1} \right| < \delta_\alpha \left\| v_{k+1} \right\| \sigma$ **ou** $\left| \left[H_{v_{k+1}}\, y \right]_{i} \right| < \delta_\alpha \left\| v_{k+1} \right\| \sigma$,

Accepter H_0 l'hypothèse d'orthogonalité entre $\left[H_{v_k}\, s \right]_i$ et v_k.

III. Fin.

CHAPITRE 4 GÉNÉRALISATION DE L'ALGORITHME DE DÉBRUITAGE POUR DES SIGNAUX 2-D

L'algorithme présenté au chapitre 3 se généralise facilement à des images (signaux 2-D). On peut "dérouler" une image selon ses lignes ou ses colonnes par exemple pour former un signal 1-D. On cherchera à estimer les caractéristiques locales d'une image au moyen de la même analyse statistique préliminaire qu'en une dimension, cette fois appliquée aux lignes, colonnes et diagonales de l'image.

4.1 Présentation de la méthode

On présente la version 2-D de l'analyse statistique préliminaire de notre algorithme à partir de l'image synthétique de taille $l \times l = 4 \times 4$ de la figure 4.1. Dans cette image, les pixels noirs ont une valeur de 0, les blancs de 255. Par convention, le pixel d'une image de coordonnées $(i, j) = (1, 1)$ se trouve en haut à gauche de l'image. On déroule l'image dans le sens de ses lignes pour obtenir un signal 1-D s de taille $n = l^2 = 16$. On note par k les indices des composantes du signal 1-D $s = (s_1, ..., s_k, ..., s_n)^T$ représentant l'image déroulée.

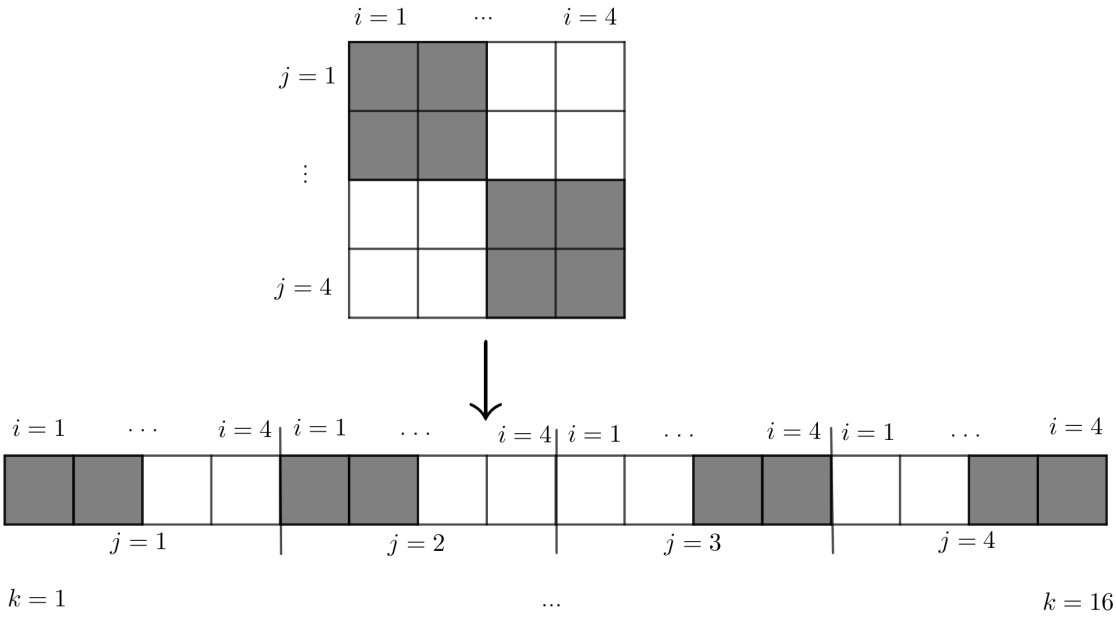


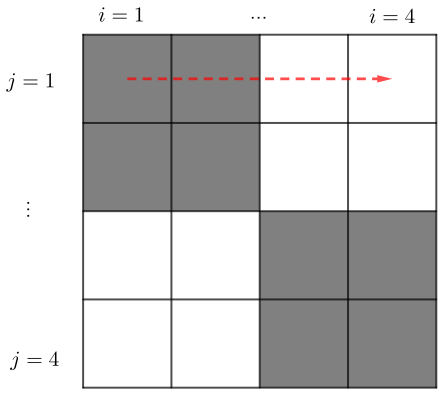
Figure 4.1 Déroulement d'une image.

Pour simplifier la présentation de la méthode, on effectue l'analyse statistique préliminaire

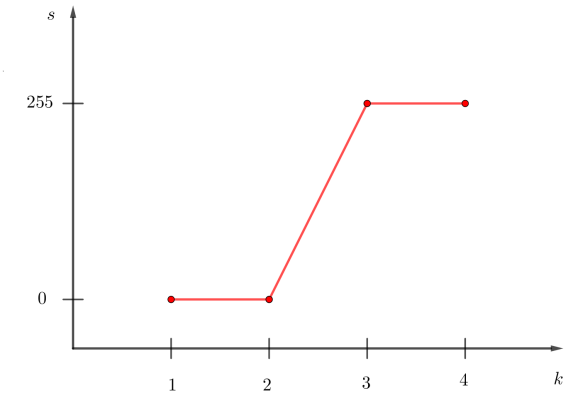
pour cette image avec le vecteur $v = (-1, 1)^T \in \mathbb{R}^2$. La même analyse pourrait être faite avec n'importe quel vecteur $v \in \mathbb{R}^m$ construit dans la section 3.1.1. De plus, on fera l'analyse statistique préliminaire sur le signal pur directement, c'est-à-dire l'image non bruitée de la figure 4.1.

Analyse des lignes de l'image

On effectue l'analyse statistique préliminaire telle que décrite dans la section 3.1 sur chacune des lignes de l'image. Chacune de ces lignes est considérée comme un signal 1-D. On présente dans la figure 4.2b le signal 1-D représentant la première ligne de l'image de la figure 4.1. On montre dans la figure 4.2c les indices, notés par une croix, correspondants à cette ligne dans le signal 1-D s représentant l'image déroulée. Ces indices sont $k = 1, 2, 3, 4$.



(a) Première ligne de l'image.



(b) Signal 1-D représentant la première ligne de l'image.



(c) Correspondance entre les indices de la première ligne de l'image et les indices de s.

Figure 4.2 Analyse des lignes de l'image.

Lors de l'analyse statistique préliminaire de ce signal 1-D, on détermine que le vecteur $v =$

$(-1,1)^T$ et les couples $(s_1,s_2)^T$ et $(s_3,s_4)^T$ sont orthogonaux puisque $s_1 = s_2 = 0$ et $s_3 = s_4 = 255$. De plus, v et $(s_2,s_3)^T$ ne sont pas orthogonaux puisque $s_2 = 0 \neq 255 = s_3$. La matrice D construite à partir de ces résultats des tests locaux est

$$D = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 4}.$$

Cette matrice D correspond à l'analyse statistique préliminaire effectuée sur la première ligne de l'image. Cependant, l'objectif est d'obtenir un modèle composite pour toute l'image, qu'on représentera comme un modèle composite pour le signal 1-D $s \in \mathbb{R}^{16}$. On doit faire correspondre cette analyse statistique préliminaire sur la première ligne de l'image aux bons indices du signal s. Comme s représente l'image déroulée dans le sens des lignes, on doit ajouter $l \times (l-1)$ colonnes de zéros à la matrice D trouvée plus haut, où on rappelle que l est la dimension des côtés de l'image $\in \mathbb{R}^{l\times l}$. On obtient

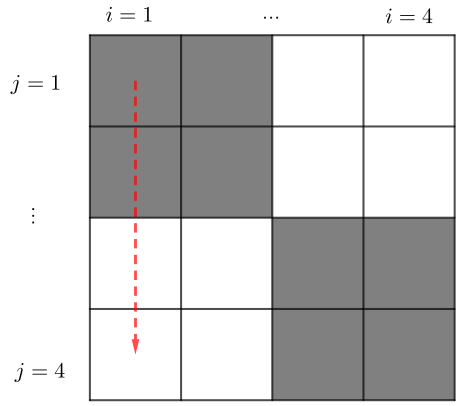
$$D_{\text{Ligne 1}} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 16}.$$

On procède de la même manière pour les trois autres lignes de l'image. On obtient la matrice D_H représentant l'analyse statistique effectuée sur les lignes de l'image. Cette matrice est la superposition des matrices $D_{\text{Ligne } j}$ pour j allant de 1 à 4 :

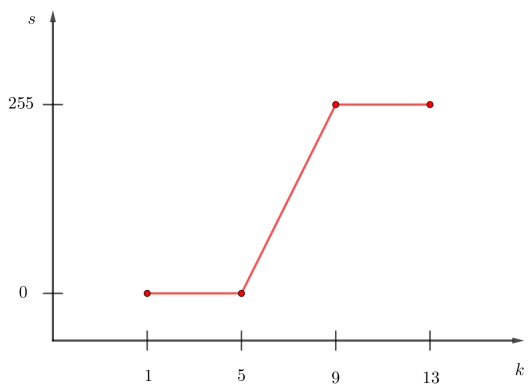
$$D_H = \begin{pmatrix} -1 & 1 & 0 & 0 & & & & & & & & & & & & \\ 0 & 0 & -1 & 1 & & & & & & & & & (0) & & & \\ & & & & -1 & 1 & 0 & 0 & & & & & & & & \\ & & & & 0 & 0 & -1 & 1 & & & & & & & & \\ & & & & & & & & -1 & 1 & 0 & 0 & & & & \\ & & & & & & & & 0 & 0 & -1 & 1 & & & & \\ & & & (0) & & & & & & & & & -1 & 1 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{8\times 16}.$$

Analyse des colonnes

On effectue maintenant l'analyse statistique préliminaire sur chacune des colonnes de l'image. On présente dans la figure 4.3b le signal 1-D représentant la première colonne de l'image. On montre dans la figure 4.3c les indices, notés par une croix, correspondants à cette ligne dans le signal 1-D s représentant l'image déroulée. Ces indices sont $k = 1, 5, 9, 13$.



(a) Première colonne de l'image.



(b) Signal 1-D représentant la première colonne de l'image.



(c) Correspondance entre les indices de la première colonne de l'image et les indices de s.

Figure 4.3 Analyse des colonnes de l'image.

Le signal 1-D correspondant à la première colonne de l'image est le même que celui correspondant à la première ligne. La matrice D obtenue suite à l'analyse statistique préliminaire de ce signal est donc encore une fois

$$D = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 4}.$$

On doit faire correspondre les résultats de ces tests d'orthogonalité entre les couples de la première colonne et le vecteur v avec les bons couples du signal s représentant l'image déroulée. On a déterminé ici que les couples $(s_1, s_5)^T$ et $(s_9, s_{13})^T$ sont orthogonaux à v et que le couple $(s_5, s_9)^T$ ne l'est pas. On doit ajouter $l - 1$ colonnes de zéros entre les colonnes de la matrice D trouvée plus haut. On obtient

$$D_{\text{Colonne 1}} = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 16}.$$

On procède de la même manière pour les trois autres colonnes de l'image. On obtient la matrice D_V représentant l'analyse statistique effectuée sur les colonnes de l'image. Cette matrice est la superposition des matrices $D_{\text{Colonne } i}$ pour i allant de 1 à 4 :

$$D_V = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & & (0) & \\ & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & & \\ & & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & \\ & (0) & & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{8\times 16}.$$

Analyse des diagonales

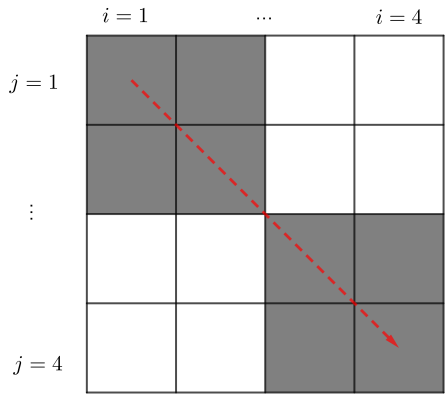
On procède de la même manière pour l'analyse des diagonales de l'image. On présente dans la figure 4.4b le signal 1-D représentant la première diagonale de l'image ainsi que la correspondance entre les indices de cette diagonale et les indices du signal 1-D représentant l'image déroulée (figure 4.4c). Ces indices sont $k = 1, 6, 11, 16$.

Le signal 1-D représentant la première diagonale de l'image est constant. On trouve donc suite à l'analyse statistique préliminaire de cette diagonale la matrice D suivante :

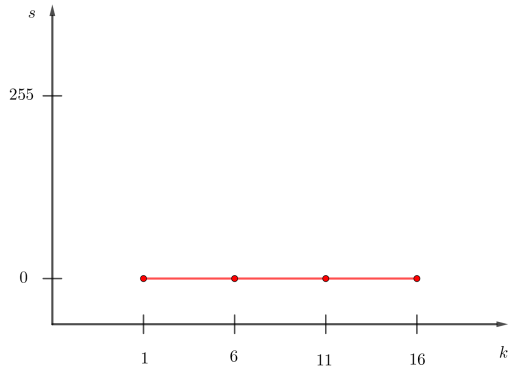
$$D = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 4}.$$

On doit faire correspondre les résultats de ces tests locaux sur la diagonale de l'image avec les bons indices du signal s. On a déterminé ici que les couples $(s_1, s_6)^T$, $(s_6, s_{11})^T$ et $(s_{11}, s_{16})^T$ sont orthogonaux à v. On doit ajouté l colonnes de zéros entre chacune des colonnes de la matrice D trouvée plus haut. La diagonale principale est notée par l'indice 0. On obtient

$$D_{\text{Diagonale } 0} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 16}.$$



(a) Première diagonale de l'image.



(b) Signal 1-D représentant la première diagonale de l'image.



(c) Correspondance entre les indices de la première diagonale de l'image et et les indices de s.

Figure 4.4 Analyse des diagonales de l'image.

On procède de la même manière pour analyser les autres diagonales de l'image. Les diagonales supérieures sont notées par les entiers positifs et les diagonales inférieures par les entiers négatifs. On ne peut analyser les diagonales de l'image dont le nombre de pixels est plus petit que m la taille du vecteur $v \in \mathbb{R}^m$ avec lequel on effectue les tests statistiques. On obtient la matrice D_{d_1} représentant l'analyse statistique préliminaire effectuée sur les diagonales de l'image en superposant les matrices $D_{\text{Diagonale } i}$ pour i allant de $-(l-m)$ à $(l-m)$:

$$D_{d_1} = \left.\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right\} \begin{matrix} D_{\text{Diagonale } -2} \\ \\ \\ D_{\text{Diagonale } 0} \\ \\ \\ D_{\text{Diagonale } 2} \end{matrix}$$
$$\in \mathbb{R}^{5\times 16}.$$

Aucun des couples des diagonales -1 et 1 ne sont orthogonaux au vecteur $v = (-1, 1)^T$ dans l'exemple traité ici, c'est pourquoi les matrices $D_{\text{Diagonale } -1}$ et $D_{\text{Diagonale } 1}$ n'apparaissent pas dans D_{d_1}.

On effectue l'analyse statistique des anti diagonales de l'image également. L'anti diagonale principale est celle partant du coin inférieur gauche de l'image et allant jusqu'au coin supérieur droit. On note D_{d_2} la superposition des matrices D représentant les analyses sur chaque anti diagonales. Dans cet exemple, on aura $D_{d_1} = D_{d_2}$.

Modèle global

Le modèle global de l'image obtenu à partir de toutes ces analyses statistiques préliminaires sur les lignes, les colonnes et les diagonales de l'image est constitué de la superposition des matrices D_H, D_V, D_{d_1} et D_{d_2} :

$$D = \begin{pmatrix} D_H \\ D_V \\ D_{d_1} \\ D_{d_2} \end{pmatrix}$$

On utilise cette matrice D pour résoudre le problème de minimisation (3.16). L'estimé obtenu $\hat{s}$ sera un signal 1-D de taille l^2. Il suffit ensuite de reformer un signal 2-D de dimensions $l \times l$ à partir de cet estimé 1-D, en suivant la même convention que lorsqu'on a déroulé l'image originale. C'est ce signal 2-D qui constitue l'estimé produit par l'algorithme de débruitage proposé.

Remarque

Les mêmes paramètres que dans le cas 1-D doivent être choisis lors du débruitage de signaux 2-D. En effet, on doit choisir les vecteurs v avec lesquels on effectue l'analyse statistique préliminaire ainsi que le seuil de rejet α.

4.2 Considérations numériques et algorithme de balayage

La généralisation de l'algorithme pour des signaux 2-D présentée dans la dernière section est coûteuse en termes de temps de calcul pour des images réelles de dimensions 512×512 par exemple. On doit alors lancer notre algorithme de débruitage pour un signal 1-D de taille 512^2. On peut accélérer le temps de calcul en découpant l'image à débruiter en fenêtres carrées de moins grandes dimensions. Le débruitage se fait alors indépendamment sur chacune de ces fenêtres, et l'estimé global de l'image est constitué de tous les estimés produits pour chaque fenêtre, replacés au bon endroit. On a cependant constaté que l'estimé global produit suite à cette procédure présente des artefacts au niveau des frontières de chaque fenêtre. On montre dans la figure 4.6c un exemple d'une image estimée par cette procédure. On peut voir dans cette image que des lignes sont visibles entre chaque fenêtre. Ces artefacts sont dus au fait que le débruitage se fait dans chaque fenêtre indépendamment des autres. On propose un algorithme de balayage afin de réduire l'importance de ces artefacts.

Lors du découpage de l'image, on choisit des fenêtres carrées se chevauchant en leur milieu. On présente dans la figure 4.5 une illustration de la procédure de découpage de l'image en fenêtres carrées de taille $m \times m$.



Figure 4.5 Découpage d'une image en fenêtres $m \times m$ se chevauchant en leur milieu.

Comme les fenêtres se chevauchent, un pixel donné aura éventuellement plusieurs valeurs estimées puisque plusieurs fenêtres de taille $m \times m$ contiendront ce pixel. Un pixel peut être contenu dans une, deux ou quatre fenêtres lorsque celles-ci se chevauchent en leur milieu. Tous les pixels qui ne sont pas sur les bords de l'image seront contenus dans quatre fenêtres. On pose la valeur de l'estimé global pour un pixel donné comme étant une moyenne pondérée des ses différentes valeurs estimées. Plus un pixel est près du centre d'une fenêtre donnée, plus le poids de la valeur estimée de ce pixel dans cette fenêtre sera grand par rapport aux poids des valeurs estimées de ce pixel dans les autres fenêtres qui le contiennent. Soit un pixel y_i d'indice i d'une image y. Soit $M \in \{1, 2, 4\}$ le nombre de fenêtres de dimensions $m \times m$ contenant ce pixel. Soit $\hat{s}_{i,k}$ la valeur estimée de ce pixel dans la fenêtre k, pour $k = 1, ..., M$. On propose la formule générale suivante pour l'estimé global $\hat{s}_i$ de ce pixel

$$\hat{s}_i = \frac{1}{Z_i} \sum_{k=1}^{M} p_{i,k}\, \hat{s}_{i,k},$$

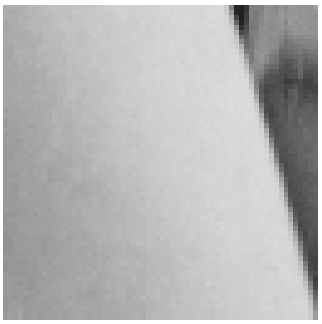
où Z_i est un est facteur de normalisation :

$$Z_i = \sum_{k=1}^{M} p_{i,k}.$$

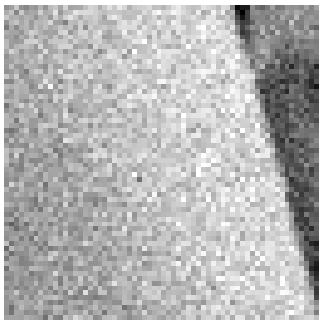
Soit z_k la position du pixel milieu de la fenêtre k et z_i la position du pixel estimé $\hat{s}_i$. Les poids $p_{i,k}$ sont calculés de la manière suivante :

$$p_{i,k} = \exp\left(-\frac{\|z_i - z_k\|^2}{2h^2}\right).$$

Le poids $p_{i,k}$ décroit lorsque la distance entre la position z_i du pixel estimé $\hat{s}_i$ dans la fenêtre k et la position z_k du centre de cette fenêtre augmente. Le paramètre h contrôle la vitesse à laquelle décroît ce poids. On présente dans la figure 4.6d le résultat de notre algorithme lorsqu'on applique cet algorithme de balayage. On a choisi dans cet exemple des fenêtres carrées de taille $m \times m = 16 \times 16$ dans le découpage. Les artefacts visibles entre chaque fenêtre dans l'image débruitée de la figure 4.6c sont beaucoup moins importants dans l'image de la figure 4.6d. On a posé $h = m/4 = 4$ pour cette simulation. D'autres résultats de notre algorithme en version 2-D seront présentés dans le chapitre 6.



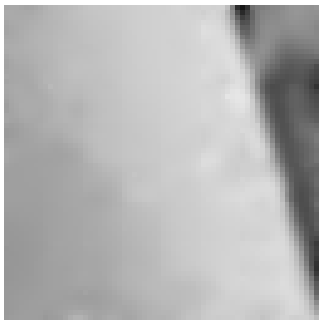
(a) Image pure 64×64.



(b) Image bruitée.



(c) Image débruité sans l'algorithme de balayage.



(d) Image débruité avec l'algorithme de balayage.

Figure 4.6 Algorithme de balayage.

CHAPITRE 5 RÉSULTATS POUR DES SIGNAUX 1-D

On compare dans ce chapitre les performances de notre algorithme avec d'autres algorithmes de la littérature pour le débruitage de signaux 1-D. Le critère de performance utilisé pour comparer les algorithmes de débruitage est l'erreur quadratique moyenne, qu'on note MSE dans la suite, pour *Mean Squared Error*. Étant donné un signal $s \in \mathbb{R}^n$ et un estimé $\hat{s}$ de ce signal produit par un algorithme de débruitage, on définit le MSE comme suit :

$$\text{MSE } (\hat{s}) := \frac{1}{n} \|\hat{s} - s\|^2. \tag{5.1}$$

Plus le MSE d'un estimé $\hat{s}$ est petit, moins l'erreur est grande et donc plus l'algorithme ayant produit cet estimé est performant. Tous nos tests seront effectués sur des signaux synthétiques, on peut donc toujours calculer le MSE à posteriori puisqu'on connaît le signal pur s. On peut comparer le MSE d'un estimé $\hat{s}$ avec la variance σ^2 du bruit ajouté. En effet, on a que $\mathbb{E}\left[\frac{1}{n}\|y - s\|^2\right] = \sigma^2$. Le MSE d'un algorithme de débruitage devrait être plus petit que σ^2. Si ce n'est pas le cas, l'estimé $\hat{s}$ n'est pas un meilleur estimé de s que le signal bruité y en termes de l'erreur quadratique moyenne.

On comparera visuellement les estimés $\hat{s}$ produits par chacun des algorithmes afin de voir si ceux-ci préservent bien certaines caractéristiques du signal pur s. On définit le vecteur erreur par

$$\text{erreur } := \hat{s} - s \in \mathbb{R}^n. \tag{5.2}$$

On présentera ce vecteur pour les algorithmes testés afin de voir si l'erreur est plus grande au niveau des discontinuités du signal s par exemple.

On définit également le rapport signal sur bruit, noté SNR pour *Signal to Noise Ratio*, comme étant le logarithme en base 10 du ratio entre la variance du signal pur s et la variance σ^2 du bruit ajouté :

$$\text{SNR } := 10 \log_{10}\left(\frac{\text{Var } [s]}{\sigma^2}\right). \tag{5.3}$$

L'unité de mesure pour cette définition du SNR est le décibel, noté dB. La mesure du SNR permet d'avoir un critère qui ne dépend pas de l'amplitude d'un signal donné pour mesurer le niveau de dégradation de celui-ci. Plus le SNR d'un signal y est grand, moins ce signal

est bruité. On divise ce chapitre en trois sections, correspondant aux trois types de signaux 1-D qu'on se propose de débruiter. La section 5.1 traite du cas où les signaux purs sont des signaux constants par morceaux, la section 5.2 du cas où les signaux sont linéaires par morceaux et la section 5.3 du cas plus général où les signaux sont lisses par morceaux.

5.1 Signaux constants par morceaux

On fait l'hypothèse dans cette section qu'on sait à priori que le signal pur s est un signal constant par morceaux. On supposera également que chaque palier des signaux constants par morceaux contient au moins trois points.

5.1.1 Choix des paramètres pour l'algorithme proposé

Deux paramètres doivent être choisis dans l'algorithme proposé dans ce mémoire. Il s'agit des vecteurs $v \in \mathbb{R}^m$ avec lesquels on effectue les tests locaux dans l'analyse statistique préliminaire et du seuil α de rejet de l'hypothèse d'orthogonalité lors de ces tests locaux.

Choix des vecteurs v pour l'analyse statistique préliminaire

Si on sait que le signal pur s sous-jacent à un signal bruité y est constant par morceaux, alors il est inutile de tester si le signal est localement linéaire ou quadratique. On a construit dans la section 3.1.1 des vecteurs ayant la propriété d'être orthogonaux aux parties constantes d'un signal. Ce sont donc ces vecteurs qui seront utilisés dans l'analyse statistique préliminaire de y dans le cas constant par morceaux. On reprend ici la définition d'un vecteur v_k donnée dans la section 3.3.5 :

$$v_k := (-1, \underbrace{0, ..., 0,}_{k-1} 1)^T \in \mathbb{R}^{k+1}. \tag{5.4}$$

Dans un exemple de la section 3.3.5, on a utilisé les vecteurs v_1, v_2 et v_3 pour l'analyse statistique préliminaire d'un signal constant. L'ajout des vecteurs v_2 et v_3 a permis d'éviter la présence d'artefacts dans l'estimé produit par notre algorithme. Soit l'ensemble $\{v_1, v_2, ..., v_M\}$ de M vecteurs v_k définis comme en (5.4). On doit choisir le paramètre M correspondant au nombre de vecteurs v_k qu'on utilisera dans l'analyse statistique préliminaire. Comme on suppose dans les expériences suivantes que les paliers des signaux constants par morceaux ont au moins 3 points, on prendra dans la suite $M = 3$. On effectue le traitement des résultats de l'analyse statistique préliminaire tel que présenté dans l'algorithme 3 de la section 3.3.5

avec les trois vecteurs v_1, v_2 et v_3.

Choix du seuil α de rejet de l'hypothèse d'orthogonalité

On peut changer le seuil de rejet α de l'hypothèse d'orthogonalité dans l'analyse statistique préliminaire afin d'être soit plus restrictif ou plus permissif. Dans tous les exemples présentés jusqu'à maintenant, le seuil de rejet $\alpha = 0.05$ a été utilisé. Ce seuil est classique en statistique pour les tests d'hypothèses et c'est la raison pour laquelle il a été choisi. Une optimisation de la valeur du seuil α a été tentée. Pour un signal bruité y donné, il est souvent possible de déterminer un seuil α améliorant l'estimé $\hat{s}$ produit par rapport à celui produit avec le seuil $\alpha = 0.05$. Cependant, cette modification du seuil est dépendante de chaque signal bruité y et n'est donc pas utile de manière générale. On propose dans la suite de garder le seuil $\alpha = 0.05$.

5.1.2 Algorithmes comparés

On compare les résultats de l'algorithme proposé pour des signaux constants par morceaux avec deux familles d'algorithmes de débruitage, soient le débruitage par seuillage des coefficients d'ondelettes et le débruitage par des méthodes de variation totale (VT). On a choisi ces deux familles car elles contiennent des variantes bien adaptées au débruitage de signaux 1-D constants par morceaux. Dans chacune des deux familles d'algorithmes de débruitage, nous testerons deux de ces variantes.

Algorithmes par transformation en bases d'ondelettes

Le débruitage par seuillage des coefficients d'ondelettes nécessite un certain nombre de paramètres à ajuster. Le choix de la base d'ondelettes dans laquelle on décompose le signal bruité y est important. On voudra sélectionner une base d'ondelettes dans laquelle le signal pur s possède une représentation creuse. Pour des signaux constants par morceaux, la base d'ondelettes de Haar est appropriée puisqu'elle est constituée de fonctions orthogonales aux polynômes de degré 0, c'est-à-dire aux fonctions constantes, dans sa version continue. Pour une description détaillée de la base d'ondelettes de Haar ainsi que pour les autres bases d'ondelettes qu'on utilisera dans la suite, voir Mallat (1999). On note $\breve{y} \in \mathbb{R}^n$ les composantes de y dans une base d'ondelettes et W l'opérateur effectuant le changement de base :

$$\breve{y} = W(y).$$

Les algorithmes de débruitage par ondelettes modifient certaines composantes dans $\breve{y}$. On

note t la fonction modifiant ces composantes. Le signal estimé $\hat{s}$ est produit en effectuant la transformation inverse des composantes $\breve{y}$ modifiés par la fonction t :

$$\hat{s} = W^{-1}\left(t\left(\breve{y}\right)\right).$$

Plusieurs fonctions t sont proposées dans la littérature. Le principe général de ces fonctions est de mettre à 0 ou de diminuer les composantes de $\breve{y}$ qui sont plus petites qu'un certain seuil δ. Les deux fonctions t les plus utilisées sont les fonctions par seuillage dur et par seuillage doux (*hard thresholding* et *soft thresholding*). La fonction t_H pour le seuillage dur est donnée par

$$t_H(\breve{y}_i) := \begin{cases} 0 & \text{si } |\breve{y}_i| \leq \delta, \\ \breve{y}_i & \text{si } |\breve{y}_i| > \delta. \end{cases} \tag{5.5}$$

La fonction t_S pour le seuillage doux est donnée par

$$t_S(\breve{y}_i) := \begin{cases} 0 & \text{si } |\breve{y}_i| \leq \delta, \\ \operatorname{sgn}(\breve{y}_i)\left(\breve{y}_i - \delta\right) & \text{si } |\breve{y}_i| > \delta, \end{cases} \tag{5.6}$$

où la fonction $\operatorname{sgn}(x)$ retourne -1 si $x < 0$, 0 si $x = 0$ et 1 si $x > 0$. Le seuil δ peut être choisi de plusieurs façons. Donoho et Johnstone (1994) introduisent un seuil appelé seuil universel. Il est défini comme suit :

$$\delta_{\text{Univ}} := \sigma\sqrt{2\,\log n}, \tag{5.7}$$

où n est la taille du signal $y \in \mathbb{R}^n$ et σ est l'écart-type du bruit ajouté. On utilisera ce seuil dans la suite. Lors de la décomposition de y dans la base d'ondelettes, on doit choisir le nombre de niveaux de décomposition dans cette base. On choisit de prendre dix niveaux de décomposition. On applique les fonctions t_H ou t_S sur les coefficients liés aux détails du signal. Coifman et Donoho (1995) ont amélioré le débruitage par ondelettes en prenant comme estimé final $\hat{s}$ la moyenne de plusieurs estimés produit à partir du signal bruité y dont tous les indices ont été décalé d'un certain entier vers la gauche ou vers la droite. Les composantes y_i décalées à une extrémité du signal sont alors replacées à l'autre extrémité (permutation circulaire). On appelle cette méthode *Translation invariant wavelet thresholding* (TIWT). Cette méthode permet de diminuer l'importance des artefacts oscillants (*ringing artefacts*) présents dans l'estimé produit suite au seuillage des coefficients d'ondelettes. On

utilisera cette méthode dans la suite. On effectuera les n permutations possibles du signal bruité $y \in \mathbb{R}^n$, et l'estimé final sera la moyenne des n signaux estimés à partir de ceux-ci.

La décomposition discrète d'un signal dans une base d'ondelettes et les fonctions de seuillage t_H et t_S sont déjà implémentées dans Mathematica. Nous avons implémenté la méthode TIWT. Le code ainsi que des exemples d'utilisations sont disponibles à partir du lien se trouvant dans l'annexe A.

Note : Il existe de multiples façons de choisir les paramètres mentionnés plus haut. Il existe également de nombreuses variantes aux algorithmes de débruitage par transformation en base d'ondelettes. L'objectif est ici de comparer notre algorithme avec une procédure usuelle.

Algorithmes de variation totale

Deux variantes des algorithmes de variation totale sont testées. Selesnick et al. (2015) proposent la généralisation suivante des algorithmes de variation totale en 1-D :

$$\hat{s} \in \underset{x \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|x - y\|^2 + \ \lambda \sum_{i=1}^{n-1} \phi(x_{i+1} - x_i) \right\}. \tag{5.8}$$

Pour l'algorithme de variation totale original, la fonction $\phi(x)$ est la fonction valeur absolue :

$$\phi(x) = \phi_{L1}(x) := |x|. \tag{5.9}$$

Selesnick et al. (2015) proposent une fonction $\phi(x)$ non convexe qui améliore l'estimé $\hat{s}$ par rapport à la fonction $\phi_{L1}(x)$:

$$\phi(x) = \phi_{\arctan}(x; a) := \frac{2}{a\sqrt{3}} \left(\arctan\left(\frac{1 + 2\,a\,|x|}{\sqrt{3}} \right) - \frac{\pi}{6} \right), \tag{5.10}$$

où a est un paramètre à ajuster. Dans le travail de Selesnick et al. (2015), le paramètre de régularisation λ dans (5.8) est fixé selon la règle $\lambda = \frac{\sqrt{n}\,\sigma}{4}$. Le paramètre a dans la fonction $\phi_{\arctan}(x; a)$ est ensuite fixé à $a = \frac{1}{4\,\lambda}$. Ce sont ces valeurs de paramètres qui seront utilisées dans la suite.

On présentera les résultats obtenus en utilisant les deux fonctions (5.9) et (5.10) dans l'algorithme (5.8). Le code Matlab est disponible pour les deux variantes présentées ici dans l'article de Selesnick et al. (2015). Ce code est présenté dans l'annexe A.

5.1.3 Résumé des paramètres

On présente dans le tableau 5.1 un résumé des paramètres utilisés dans les cinq algorithmes testés dans le cas de signaux constants par morceaux.

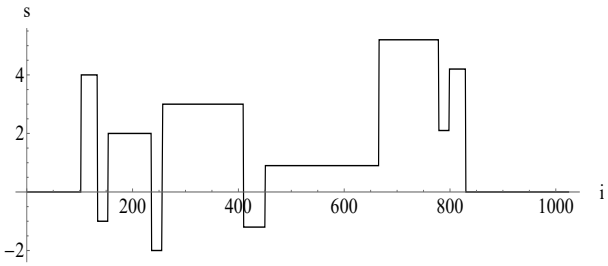
Tableau 5.1 Paramètres utilisés dans les algorithmes testés pour des signaux constants par morceaux.

Méthode	Paramètres
TIWT, seuil doux	Famille : Haar Niveaux de décomposition : 10 $\delta_{\text{Univ}} = \sigma\sqrt{2\log n}$ Nombre de permutations : n
TIWT, seuil dur	Famille : Haar Niveaux de décomposition : 10 $\delta_{\text{Univ}} = \sigma\sqrt{2\log n}$ Nombre de permutations : n
VT, ϕ_{L1}	$\lambda = \frac{\sqrt{n}\,\sigma}{4}$
VT, ϕ_{arctan}	$\lambda = \frac{\sqrt{n}\,\sigma}{4}$ $a = \frac{1}{4\,\lambda}$
Algorithme proposé	$\{v_1, v_2, v_3\}$ comme définis en (5.4) $\alpha = 0.05$

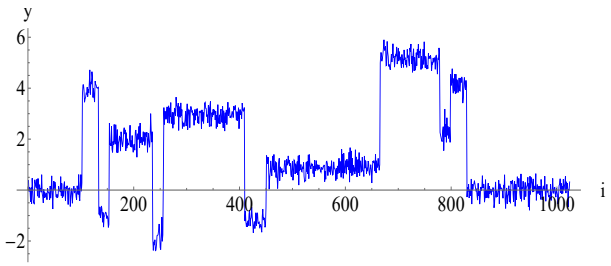
5.1.4 Résultats pour un signal test

On présente dans la figure 5.1 le signal constant par morceaux pur et bruité déjà présenté dans l'introduction. Ce signal provient d'un paquet MatLab nommé WaveLab (https ://stat-web.stanford.edu/ wavelab/). On peut obtenir le signal en appelant la fonction MakeSignal de WaveLab puis en spécifiant "Blocks" et la taille n du signal en entrée. Dans cet exemple, on a posé $n = 1024$. Ce signal est souvent utilisé comme signal test dans la littérature. La fonction MakeSignal peut être trouvée dans l'annexe A.

On montre dans la figure 5.2 l'estimé $\hat{s}$ (à gauche) et le vecteur erreur (à droite) pour chaque algorithme à partir du signal bruité y de la figure 5.1b.

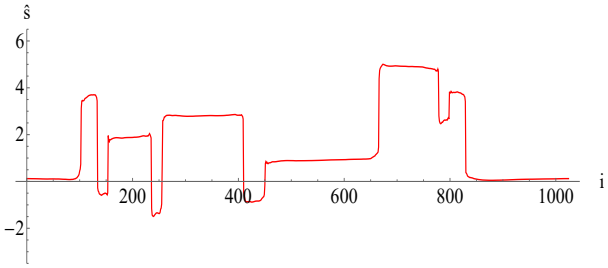


(a) Signal pur.

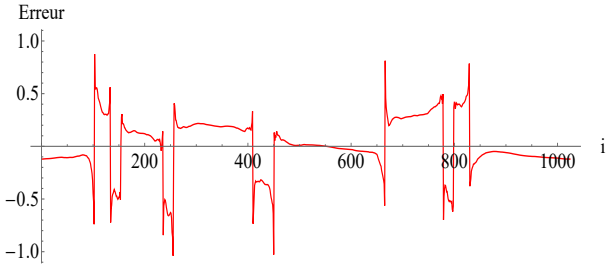


(b) Signal bruité, $\sigma = 0.3$, SNR = 16.1 dB.

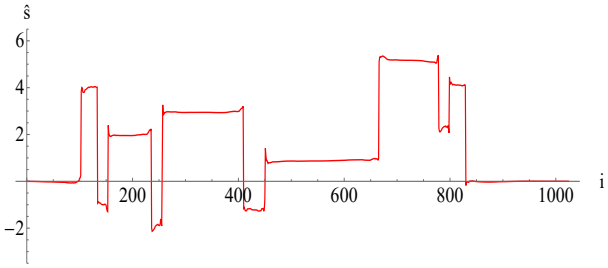
Figure 5.1 Signal constant par morceaux "Blocks" pur et bruité.



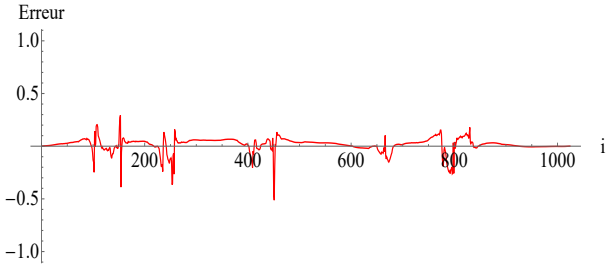
(a) TIWT, seuil doux.



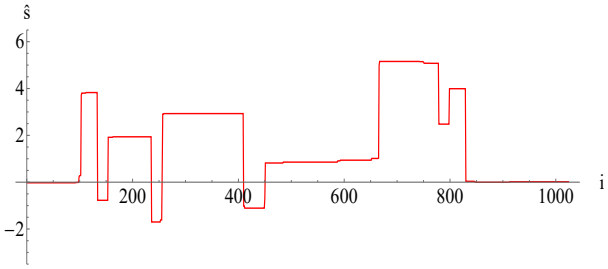
(b) Erreur de l'estimé, MSE = 62.073×10^{-3}.



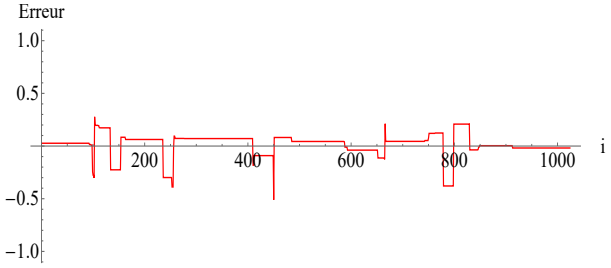
(c) TIWT, seuil dur.



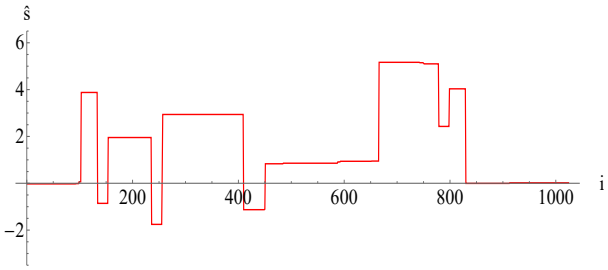
(d) Erreur de l'estimé, MSE = 4.678×10^{-3}.



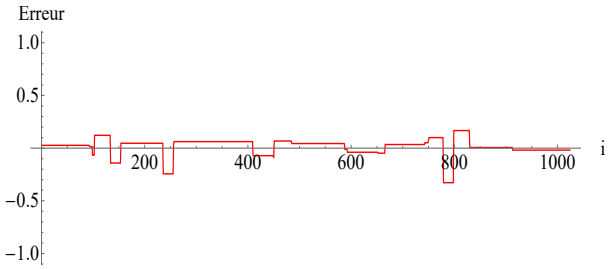
(e) VT, ϕ_{L1}.



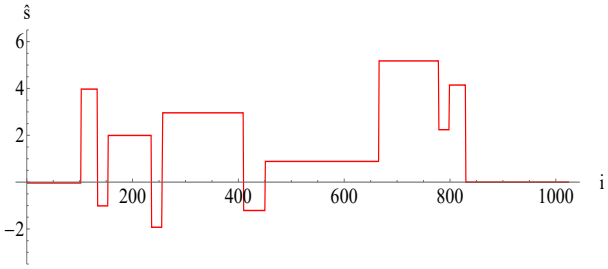
(f) Erreur de l'estimé, MSE = 11.501×10^{-3}.



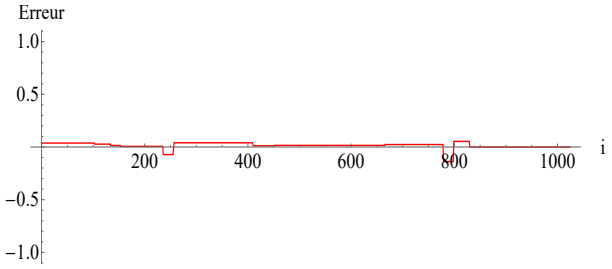
(g) VT, ϕ_{arctan}.



(h) Erreur de l'estimé, MSE $= 6.863 \times 10^{-3}$.



(i) Algorithme proposé.



(j) Erreur de l'estimé, MSE $= 1.090 \times 10^{-3}$.

Figure 5.2 Comparaison des algorithmes sur le signal "Blocks".

L'algorithme proposé donne le meilleur estimé en termes de MSE par rapport aux quatre autres algorithmes de la littérature présentés. L'estimé produit par la méthode d'ondelettes avec le seuil doux, présenté dans la figure 5.2a, arrondit les transitions entre les paliers. Cela se traduit par de grandes erreurs au niveau de chacune de ces transitions (figure 5.2b). De plus, on constate dans cette dernière figure que la hauteur de chacun des paliers du signal pur n'est pas bien estimée. Des artefacts sont visibles au niveau de chacun des changements de paliers dans l'estimé produit par la méthode d'ondelettes avec le seuil dur, présenté dans la figure 5.2c. C'est ce type d'artefacts qu'on appelle artefacts oscillants. Cette caractéristique des méthodes par ondelettes est appelée effet de pseudo-Gibbs. Bien que la méthode TIWT permette de diminuer l'importance de cet effet, on constate que l'erreur, à cause de ces artefacts, est plus grande au niveau de chaque changement de paliers (figure 5.2d). Les transitions abruptes sont bien estimées dans le cas des algorithmes de variation totale, présentés dans les figures 5.2e et 5.2g. Cependant, on constate que la hauteur de chaque palier n'est pas bien estimée dans le cas de l'algorithme avec la fonction ϕ_{L1} (figure 5.2f). La fonction ϕ_{arctan} permet de diminuer ces erreurs, sans toutefois les éliminer complètement (figure 5.2h). L'estimé produit par notre algorithme préserve à la fois parfaitement les transitions abruptes du signal ainsi que le niveau de chaque palier (figure 5.2i). En effet, on constate dans la figure 5.2j que l'erreur est presque nulle sur tous les morceaux du signal. De plus l'erreur n'est pas plus grande au niveau des transitions de ces morceaux, ce qui signifie que les discontinuités

du signal pur sont bien estimées.

Le tableau 5.2 présente une moyenne du MSE obtenue pour chaque algorithme sur 100 réalisations du bruit pour trois valeurs de SNR, soient 10 dB, 16 dB et 20 dB pour le même signal "Blocks". L'écart-type du bruit ajouté σ correspondant à ces valeurs de SNR est respectivement approximativement de 0.606, 0.303 et 0.191. Des exemples du signal "Blocks" dégradé par ces trois niveaux de bruit peuvent être trouvés dans l'annexe B.

Tableau 5.2 MSE moyen sur 100 réalisations du bruit pour le signal "Blocks" [1].

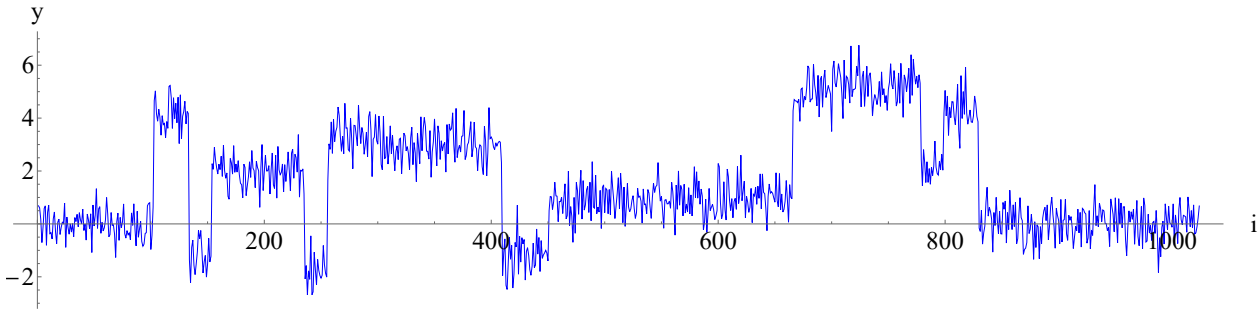
Méthode	SNR = 10 dB ($\times 10^{-2}$)	SNR = 16 dB ($\times 10^{-3}$)	SNR = 20 dB ($\times 10^{-4}$)
TIWT, seuil doux	20.8 ± 0.3	58.3 ± 0.8	231 ± 4
TIWT, seuil dur	$\mathbf{2.4 \pm 0.1}$	4.7 ± 0.2	16.1 ± 0.5
VT, ϕ_{L1}	3.7 ± 0.1	9.1 ± 0.4	37 ± 1
VT, $\phi_{\arctan}$	2.8 ± 0.1	5.1 ± 0.2	14.8 ± 0.8
Algorithme proposé	4.4 ± 0.3	$\mathbf{1.8 \pm 0.4}$	$\mathbf{5.8 \pm 0.4}$

Lorsque le SNR est de 16 et 20 dB, l'algorithme proposé obtient le MSE le plus bas en moyenne. L'erreur est diminuée d'un facteur 1.6 par rapport à l'algorithme par ondelettes avec seuil dur lorsque le SNR est de 16 dB, d'un facteur 1.6 par rapport à l'algorithme de variation totale avec $\phi_{\arctan}$ lorsque le SNR est de 20 dB [2]. Lorsque le SNR est de 10 dB, ce qui correspond à un niveau de bruit plus grand, l'algorithme par seuillage dur des coefficients d'ondelettes obtient le MSE le plus bas en moyenne. L'erreur de notre algorithme est plus grande d'un facteur 1.3 par rapport à ce dernier.

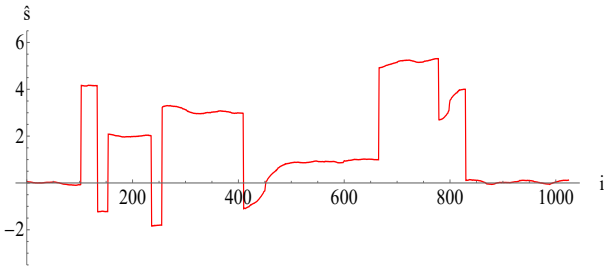
On présente dans la figure 5.3 le signal "Blocks" bruité ayant un SNR de 10 dB, ainsi que les estimés $\hat{s}$ produits par l'algorithme proposé et l'algorithme de débruitage par ondelette avec le seuil dur. L'estimé produit par l'algorithme proposé présenté dans la figure 5.3b souffre d'erreurs au niveau de certains changements de paliers. Les discontinuités présentes dans le signal pur aux endroits de ces changements de paliers sont arrondies dans le signal estimé. De plus chaque palier n'est pas parfaitement constant. Ces erreurs sont visibles dans la figure 5.3c. Ces erreurs proviennent de la difficulté à extraire les caractéristiques locales du signal lors de l'analyse statistique préliminaire lorsque le bruit augmente. Deux types d'erreurs sont possibles lors de l'analyse statistique préliminaire. Les écarts entre les paliers du signal sont parfois comparables aux écarts entre deux composantes consécutives du bruit. On considère dans ce cas qu'il n'y a pas de discontinuité à ces endroits, ce qui arrondit les transitions

1. L'incertitude sur les mesures ($\pm$) correspond à deux fois l'écart-type empirique de la moyenne.
2. Le MSE est une mesure de l'erreur au carré. Pour calculer ces facteurs, on divise le MSE le plus grand par le plus petit puis on extrait la racine : $\sqrt{\frac{4.7}{1.8}} \approx 1.6$ et $\sqrt{\frac{14.8}{5.8}} \approx 1.6$.

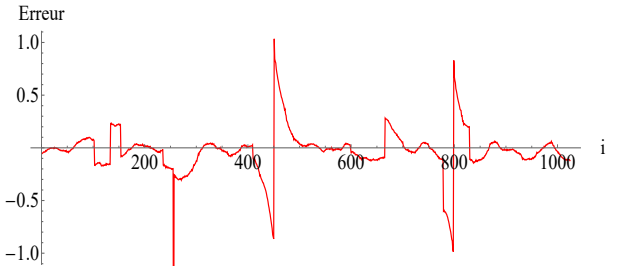
entre ces paliers. De plus, il est possible qu'un faux changement de palier soit détecté lorsque l'écart entre deux valeurs consécutives du bruit est plus grand que le seuil de rejet. Cela fait en sorte que les paliers ne sont pas parfaitement constants dans la figure 5.3b.



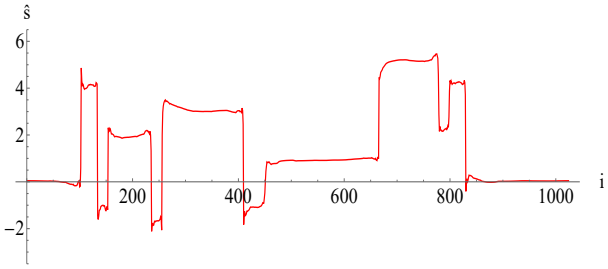
(a) Signal bruité "Blocks", $\sigma \approx 0.606$, SNR = 10 dB.



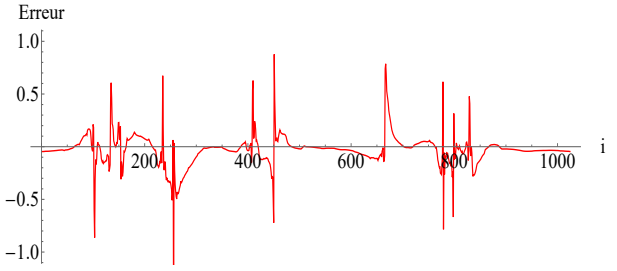
(b) Algorithme proposé.



(c) Erreur de l'estimé, MSE $= 4.840 \times 10^{-2}$.



(d) TIWT, seuil dur.



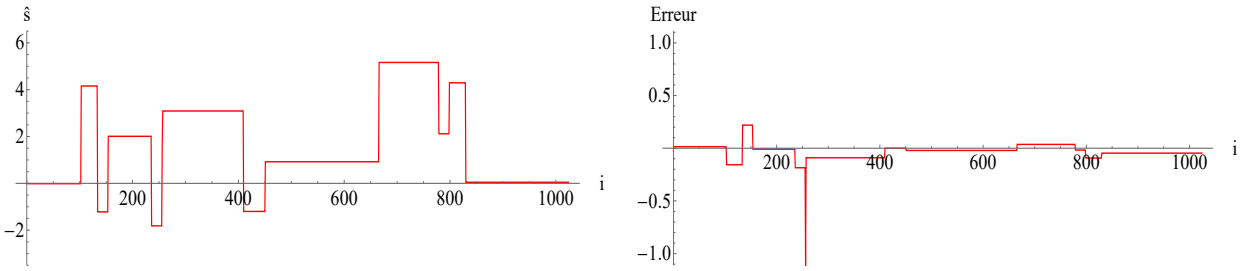
(e) Erreur de l'estimé, MSE $= 2.309 \times 10^{-2}$.

Figure 5.3 Comparaison des algorithmes sur le signal "Blocks" pour un bruit ajouté de variance plus grande.

Version oracle de l'algorithme proposé

On peut illustrer le phénomène des erreurs lors de l'analyse statistique préliminaire discuté plus haut en introduisant une version "oracle" de l'algorithme proposé. Dans cette version, lors de l'analyse statistique préliminaire, la matrice D est construite à partir du signal pur s et non à partir du signal bruité y. Ainsi, on ne fait pas d'erreurs dans l'analyse statistique préliminaire et on détecte toutes les discontinuités du signal pur. On montre dans la figure

5.4 les résultats de cette version oracle de notre l'algorithme sur le même signal bruité de la figure 5.3a.



(a) Signal estimé.

(b) Erreur de l'estimé, MSE $= 0.584 \times 10^{-2}$.

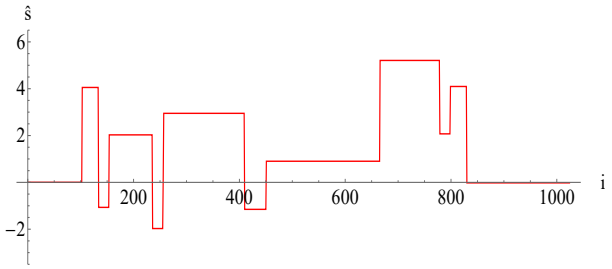
Figure 5.4 Résultats de la version oracle de notre algorithme sur le signal "Blocks".

L'estimé produit par la version oracle de notre algorithme, figure 5.4a, est très similaire au signal pur. Les paliers sont parfaitement constants, et les transitions entre ceux-ci sont nettes. Quantitativement, le MSE de l'estimé est significativement plus petit que celui de l'algorithme par ondelettes avec le seuil dur. Seul un changement de palier, à $i \approx 250$, n'est pas parfaitement localisé dans l'estimé ce qui provoque le pic dans le vecteur erreur visible dans la figure 5.4b. Comme l'information relative aux caractéristiques locales du signal s a été extraite sans erreur lors de l'analyse statistique préliminaire, la matrice D construite lors de cette analyse est bien adaptée au signal pur s. La résolution du problème de minimisation (3.16) avec cette matrice D produit une solution très proche du signal pur s. Le fait que la version oracle de l'algorithme proposé donne de bons résultats est encourageant en ce sens qu'on peut penser à des améliorations de l'analyse statistique préliminaire qui permettraient d'avoir des résultats s'approchant de ceux de la version oracle. On pourrait par exemple faire des tests statistiques plus complexes sur les différentes parties du signal bruité y afin que l'analyse soit plus robuste au bruit. Une autre idée serait de lancer une seconde procédure de débruitage sur le signal estimé par notre algorithme. En effet, même si certains changements de paliers sont arrondis dans l'estimé de la figure 5.3b, la pente entre ces paliers est très prononcée. On pourrait tenter de détecter la position de ces fortes pentes et les classifier comme des discontinuités.

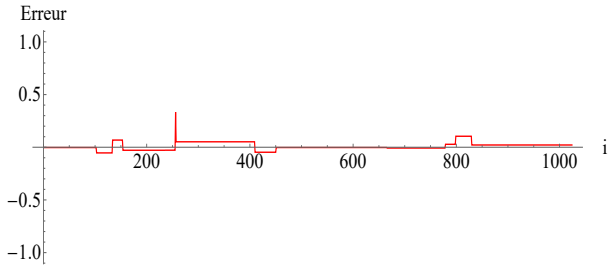
Caractéristiques de la solution pour les cas $\lambda = 0$ et $\lambda < 0$

L'estimé produit par notre algorithme de débruitage est la solution d'un problème d'optimisation avec contrainte. On a résolu de manière analytique ce problème dans la section 3.3. Deux cas de figure sont possibles lors de la résolution du problème d'optimisation. Le multi-

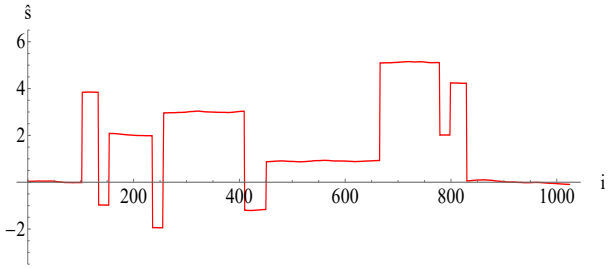
plicateur de Lagrange, noté λ, peut être nul ou strictement négatif. La solution du problème d'optimisation diffère lorsqu'elle est obtenue à partir de l'un ou l'autre de ces deux cas, voir section 3.3.3. On montre dans la figure 5.5 des exemples du signal estimé par notre algorithme lorsque le problème d'optimisation a été résolu avec le cas $\lambda = 0$ et $\lambda < 0$. Dans les deux cas, le signal pur est le signal "Blocks" et le bruit ajouté est ajusté pour obtenir un SNR de 16 dB.



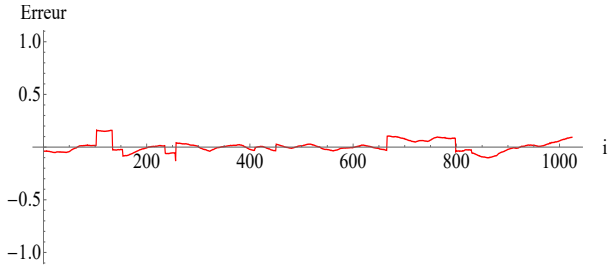
(a) Solution obtenue avec le cas $\lambda = 0$



(b) Erreur de l'estimé, MSE $= 1.299 \times 10^{-3}$.



(c) Solution obtenue avec le cas $\lambda < 0$



(d) Erreur de l'estimé, MSE $= 2.693 \times 10^{-3}$.

Figure 5.5 Différences de la solution pour les cas $\lambda = 0$ et $\lambda < 0$, signal "Blocks" 16 dB.

Dans cet exemple, l'estimé obtenu avec le cas $\lambda = 0$ obtient un MSE plus bas. Lorsque la solution est obtenue avec le cas $\lambda = 0$, cette solution est la projection du signal bruité y sur le noyau de D représentant notre modèle composite du signal s. Dans cet exemple, les éléments du noyau de D sont des signaux constants par morceaux. Ainsi, le signal estimé avec le cas $\lambda = 0$ est constant par morceaux. Chaque palier est parfaitement constant, les erreurs viennent du fait que la hauteur et la position de ces paliers ne sont pas parfaitement estimées, voir figure 5.5b. Lorsque la solution est obtenue avec le cas $\lambda < 0$, celle-ci n'est pas nécessairement constante par morceaux. On remarque alors que chaque palier du signal estimé n'est pas parfaitement constant, voir figure 5.5d.

Il n'y a pas de choix à faire lors de la résolution du problème d'optimisation, le fait d'obtenir la solution avec le cas $\lambda = 0$ ou $\lambda < 0$ dépend du signal bruité. Les solutions obtenues avec l'un ou l'autre de ces deux cas ont cependant des caractéristiques différentes.

Temps de calcul

Le tableau 5.3 montre le temps de calcul moyen sur 100 réalisations pour produire un estimé du signal "Blocks" avec un bruit ajouté de $\sigma = 0.3$ comme dans la figure 5.1b. L'algorithme proposé est plus lent que les quatre autres algorithmes. L'optique de ce projet de maîtrise était de développer un nouvel algorithme de débruitage basé sur une analyse statistique des signaux. L'accent a surtout été porté sur le développement de la méthode en tant que telle plutôt que sur une version rapide de celle-ci. L'optimisation du code utilisé pour produire l'estimé $\hat{s}$ est une des voies d'amélioration de l'algorithme proposé.

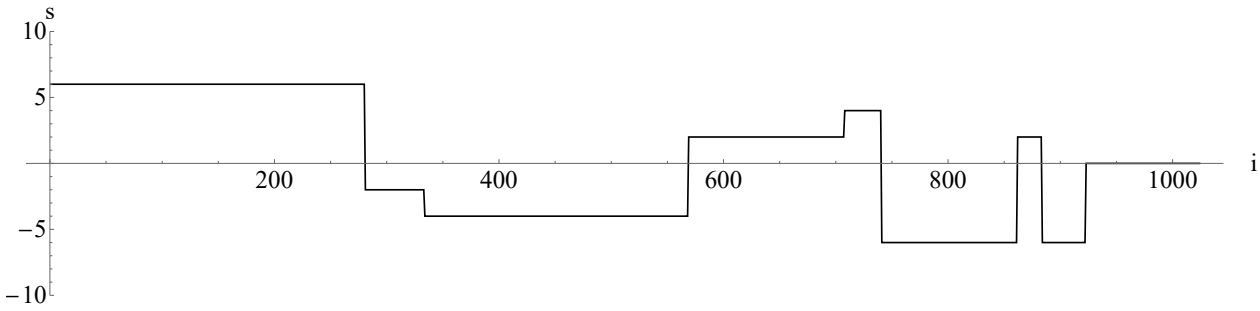
Tableau 5.3 Temps de calcul moyen sur 100 réalisations du signal "Blocks", $\sigma = 0.3$.

Méthode	Temps de calcul moyen (s)
TIWT, seuil doux	0.662 ± 0.008
TIWT, seuil dur	0.664 ± 0.008
VT, ϕ_{L1}	$\mathbf{0.0452 \pm 0.0004}$
VT, ϕ_{arctan}	0.0633 ± 0.0008
Algorithme proposé	2.80 ± 0.04

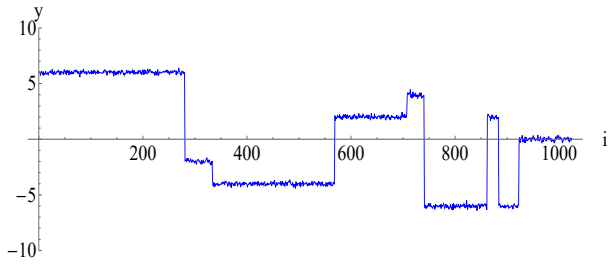
5.1.5 Résultats pour des signaux aléatoires

On effectue maintenant une analyse statistique des performances des différents algorithmes en fonction du bruit ajouté sur des signaux constants par morceaux aléatoires. On génère des signaux constants par morceaux aléatoirement de la manière suivante. Chaque signal pur s est de taille $n = 1024$. La longueur des paliers du signal est une variable aléatoire entière uniformément distribuée entre 3 et 300. La hauteur des paliers est une variable aléatoire entière uniformément distribuée dans l'ensemble $\{-6, -4, -2, 0, 2, 4, 6\}$. On présente un exemple d'un signal s généré par cette procédure dans la figure 5.6a. On fait varier le SNR de 30 dB à 10 dB par des pas de 2 dB. On montre une réalisation y du signal bruité ayant un SNR de 30 dB dans la figure 5.6b et un SNR de 10 dB dans la figure 5.6c. L'écart-type du bruit ajouté est ajusté pour obtenir un SNR constant selon la variance de chaque signal.

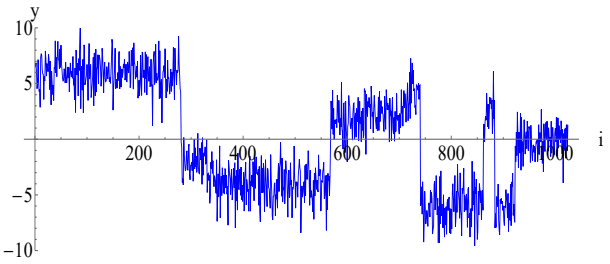
On génère 100 signaux constants par morceaux selon cette procédure. Pour chaque valeur de SNR, les algorithmes produisent un estimé $\hat{s}$ pour les 100 signaux. On effectue ensuite la moyenne des MSE obtenus sur ces 100 signaux pour chaque algorithme et pour chaque valeur de SNR. On présente dans la figure 5.7 les résultats de cette expérience. Les résultats en termes de MSE de la méthode par ondelettes avec le seuil doux sont significativement moins bons que les quatre autres algorithmes. Ceux-ci ne sont pas présentés dans la figure 5.7.



(a) Signal pur.



(b) Signal bruité, $\sigma \approx 0.143$, SNR = 30 dB.



(c) Signal bruité, $\sigma \approx 1.432$, SNR = 10 dB.

Figure 5.6 Exemple d'un signal constant par morceaux généré aléatoirement.

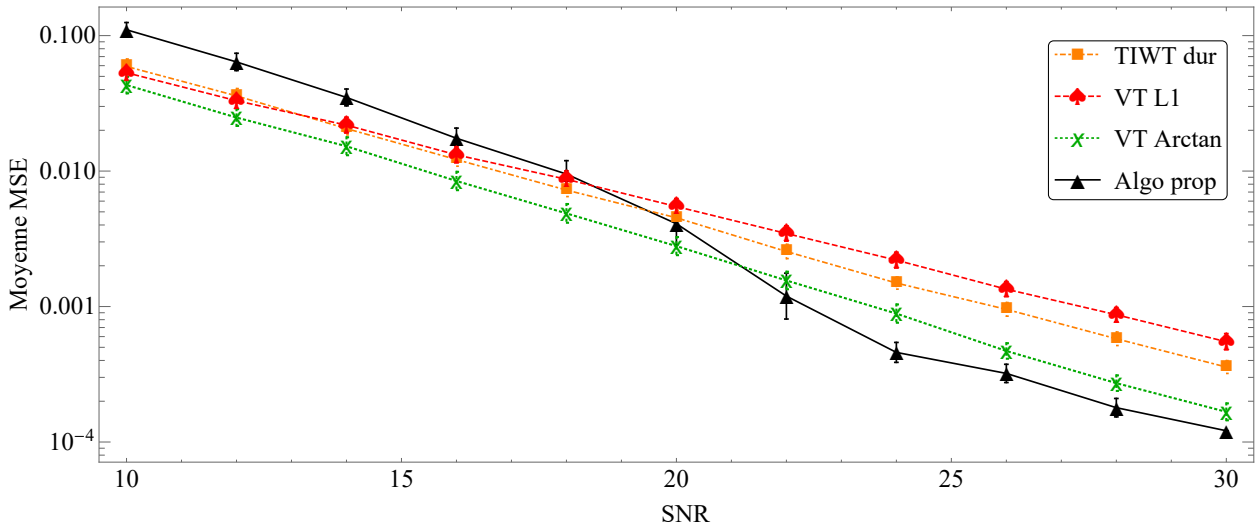


Figure 5.7 Comparaison des résultats pour des signaux constants par morceaux générés aléatoirement [4].

4. Les barres d'erreurs correspondent encore une fois à deux fois l'écart-type empirique de la moyenne.

On constate dans la figure 5.7 que pour les niveaux de bruit les plus bas, c'est-à-dire pour des SNR entre 24 et 30 dB, l'algorithme proposé obtient les meilleurs résultats en termes de MSE par rapport aux trois autres algorithmes. En particulier, pour un SNR de 24 dB, l'erreur moyenne de notre algorithme est plus petite d'un facteur 1.4 par rapport à l'algorithme de variation totale avec ϕ_{arctan}, d'un facteur 1.8 par rapport à l'algorithme par ondelettes avec seuil dur et d'un facteur 2.2 par rapport à l'algorithme de variation totale avec ϕ_{L1}. Lorsque le bruit est plus grand, notre algorithme performe moins bien comparé aux autres algorithmes. Ce phénomène peut être expliqué de la même manière que dans le cas du signal "Blocks" ayant un SNR de 10 dB de la figure 5.3a : l'analyse statistique préliminaire est moins efficace pour détecter les caractéristiques locales des signaux dans ce cas. Pour un SNR de 10 dB, l'erreur de notre algorithme est plus grande d'un facteur 1.5 en moyenne par rapport aux trois autres algorithmes. On montre dans la figure 5.8 les résultats de la version oracle de notre algorithme, qu'on compare avec ceux de la version originale de notre algorithme ainsi que ceux de l'algorithme par variation totale avec la fonction ϕ_{arctan}.

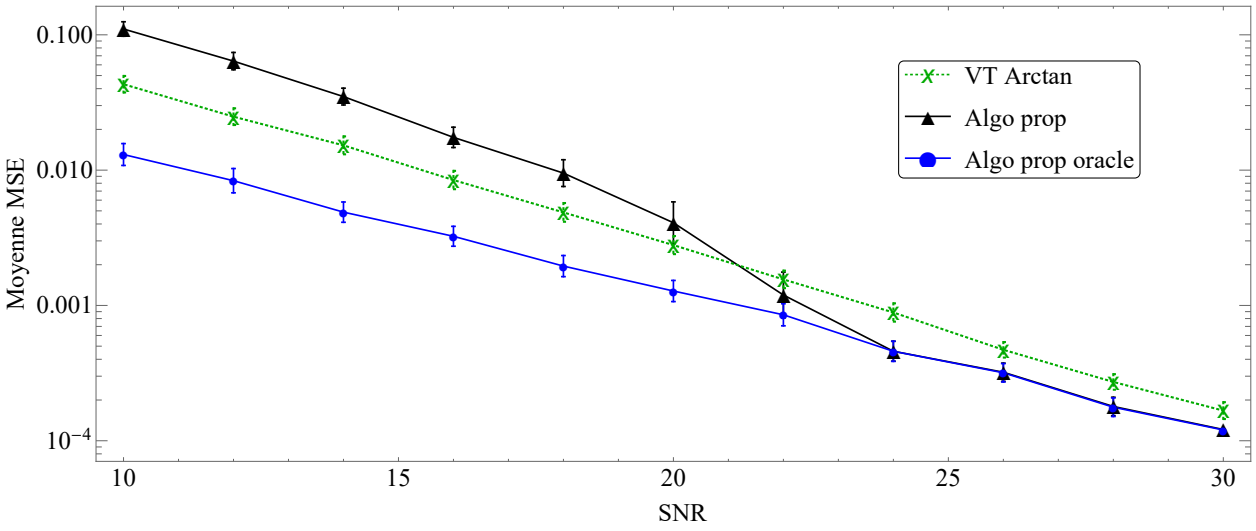


Figure 5.8 Comparaison des résultats avec la version oracle de l'algorithme pour des signaux constants par morceaux générés aléatoirement.

L'erreur de la version oracle de notre algorithme est en moyenne 1.5 fois plus petite que celle de la méthode par variation totale. On ne constate pas, dans la figue 5.8, la détérioration des résultats de notre algorithme pour des valeurs de bruit plus grand pour la version oracle de notre algorithme. Encore une fois, les bons résultats de la cette version de notre algorithme permet d'espérer de meilleurs résultats pour la version originale, moyennant une amélioration de la détection des caractéristiques locales des signaux lors de l'analyse statistique préliminaire pour des niveaux bruit plus grands.

5.2 Signaux linéaires par morceaux

On traite maintenant du cas où on sait à priori que le signal pur s est un signal linéaire par morceaux. On suppose également que chaque morceau linéaire contient au moins cinq points.

5.2.1 Choix des paramètres pour l'algorithme proposé

On a construit dans la section 3.1.1 des vecteurs ayant la propriété d'être orthogonaux aux parties linéaires d'un signal. C'est donc ces vecteurs que nous utiliserons dans l'analyse statistique préliminaire de y dans le cas linéaire par morceaux. On définit dans cette section un vecteur v_k comme suit :

$$v_k := (1, \underbrace{0, ..., 0}_{k-1}, -2, \underbrace{0, ..., 0}_{k-1}, 1)^T \in \mathbb{R}^{2k+1}. \tag{5.11}$$

Avec cette définition, on a que $v_1 = (1, -2, 1)^T$. Soit l'ensemble $\{v_1, v_2, ..., v_M\}$ de M vecteurs v_k définis comme en (5.11). On doit choisir le paramètre M correspondant au nombre de vecteurs v_k qu'on utilisera dans l'analyse statistique préliminaire. Comme on suppose dans les expériences suivantes que les morceaux linéaires des signaux ont au moins 5 points, on prendra dans la suite $M = 2$ (le vecteur v_2 est de taille 5). Il est possible de faire un traitement des résultats de l'analyse statistique préliminaire analogue à celui présenté dans l'algorithme 3 de la section 3.3.5 pour les vecteurs v_k présentés ici. Cependant, ce traitement n'a pas permis d'améliorer la qualité de l'estimé. Par conséquent, nous n'effectuerons pas ce traitement dans le cas linéaire par morceaux.

Pour la même raison que dans le cas des signaux constants par morceaux, on propose de garder le seuil de rejet α à 0.05.

5.2.2 Algorithmes comparés

On compare les résultats de l'algorithme proposé pour des signaux linéaires par morceaux avec deux variantes d'algorithmes de débruitage par seuillage des coefficients d'ondelettes. Les algorithmes de variation totale ne sont pas bien adaptés pour le débruitage de signaux qui ne sont pas constants par morceaux, bien que des tentatives aient été faites afin de développer des méthodes plus générales permettant de gérer un plus grand éventail de signaux (Bredies et al., 2010).

Algorithmes par transformation en bases d'ondelettes

Pour des signaux linéaires par morceaux, on peut choisir une autre base d'ondelettes que la base de Haar mieux adaptée à ce type de signaux. On propose dans ce cas les ondelettes de Daubechies à 2 moments nuls (*vanishing moments*). Cette base d'ondelettes est notée db2, à ne pas confondre avec notre notation dB pour décibel. Le choix de cette base d'ondelettes se justifie par le fait que les fonctions qui la composent dans sa version continue sont orthogonales aux polynômes de degré inférieur ou égal à 1. Cette base est donc bien adaptée pour représenter des fonctions linéaires par morceaux.

L'algorithme de débruitage procède de la même manière que pour les signaux constants par morceaux, c'est-à-dire qu'on modifie selon les fonctions de seuillage dur (5.5) et doux (5.6) les composantes $\breve{y}$ du signal bruité y dans la base d'ondelettes db2. On utilise encore une fois dix niveaux de décomposition et le même seuil δ_{Univ} est utilisé. L'estimé final est une moyenne de n estimés produits à partir des permutation circulaires de y (méthode TIWT).

5.2.3 Résumé des paramètres

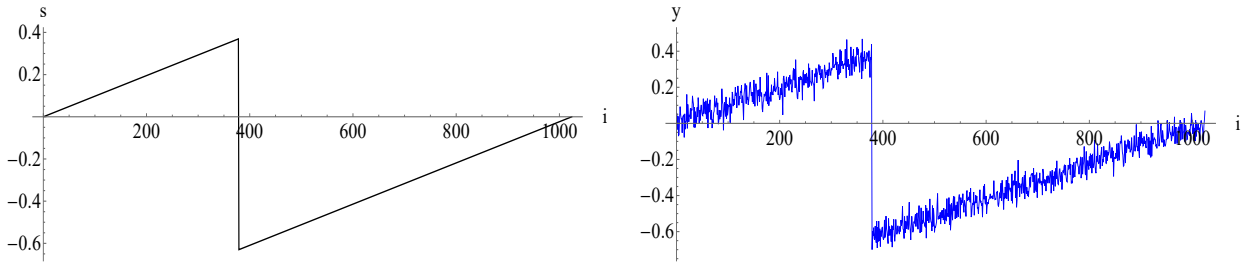
Le tableau 5.4 présente un résumé des paramètres utilisés dans les trois algorithmes testés

Tableau 5.4 Paramètres utilisés dans les algorithmes testés pour des signaux linéaires par morceaux.

Méthode	Paramètres
TIWT, seuil doux	Famille : db2 Niveaux de décomposition : 10 $\delta_{\text{Univ}} = \sigma\sqrt{2\log n}$ Nombre de permutations : n
TIWT, seuil dur	Famille : db2 Niveaux de décomposition : 10 $\delta_{\text{Univ}} = \sigma\sqrt{2\log n}$ Nombre de permutations : n
Algorithme proposé	$\{v_1, v_2\}$ comme définis en (5.11) $\alpha = 0.05$

5.2.4 Résultats pour un signal test

On présente dans la figure 5.9 le signal linéaire par morceaux pur et bruité déjà présenté dans l'introduction. On peut obtenir ce signal à partir de la même fonction MakeSignal de WaveLab en spécifiant "Ramp". Le signal est de taille $n = 1024$.



(a) Signal pur.

(b) Signal bruité y, $\sigma = 0.05$, SNR = 15.2 dB.

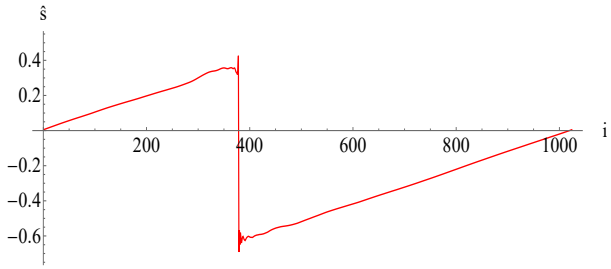
Figure 5.9 Signal linéaire par morceaux "Ramp" pur et bruité.

On montre dans la figure 5.10 l'estimé $\hat{s}$ produit par chaque algorithme à partir du signal bruité de la figure 5.9b.

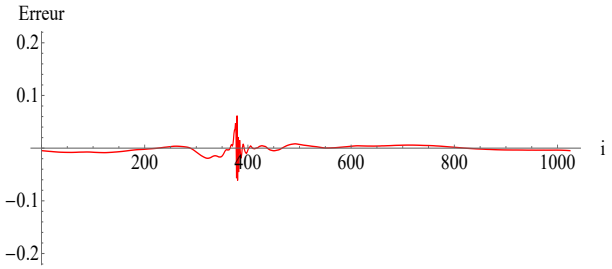
L'estimé $\hat{s}$ produit par l'algorithme proposé est le meilleur en termes de MSE. Dans le cas de l'estimé produit par la méthode d'ondelettes avec le seuil doux, présenté dans la figure 5.10c, la discontinuité entre les deux parties linéaires du signal est arrondie. Cela se traduit par une grande erreur au niveau de cette discontinuité (figure 5.10d). Dans le cas de l'estimé produit par la méthode d'ondelettes avec seuil dur, présenté dans la figure 5.10a, des artefacts oscillants sont visibles au niveau de la discontinuité. Cela crée une erreur plus grande à ce niveau (figure 5.10b). Notre algorithme ne souffre pas d'erreur plus grande au niveau de la discontinuité du signal (figure 5.10f). Les deux parties linéaires du signal pur sont très bien estimées (figure 5.10e).

On présente dans le tableau 5.5 la moyenne des MSE obtenus pour chaque algorithme sur 100 réalisations de signaux bruités y ayant des SNR de 10 dB, 16 dB et 20 dB à partir du signal pur "Ramp" de la figure 5.9a. L'écart-type du bruit σ correspondant à ces valeurs de SNR est respectivement approximativement de 0.091, 0.046 et 0.029. Des exemples du signal "Ramp" dégradé par ces trois niveaux de bruit peuvent être trouvés dans l'annexe B.

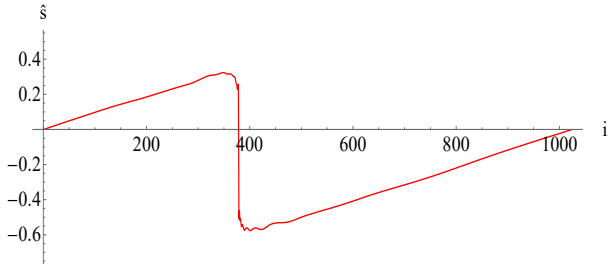
L'algorithme proposé donne les meilleurs résultats en moyenne en termes de MSE pour le signal bruité à 16 et à 20 dB. L'erreur de notre algorithme est réduite d'un facteur 1.3 et d'un facteur 1.5 respectivement pour ces deux niveaux de bruit par rapport à l'algorithme par seuillage des coefficients d'ondelettes avec le seuil dur. Pour un SNR de 10 dB, l'algorithme



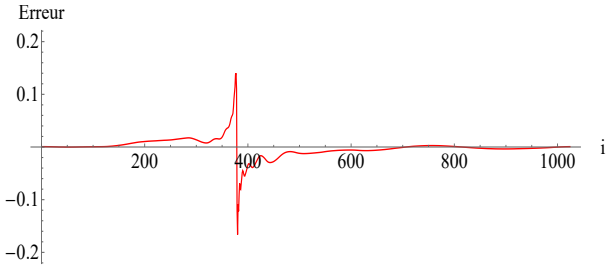
(a) TIWT, seuil dur.



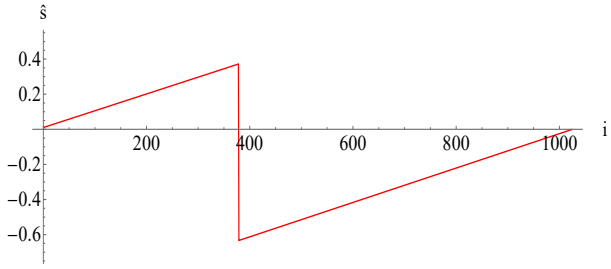
(b) Erreur de l'estimé, MSE $= 5.315 \times 10^{-5}$.



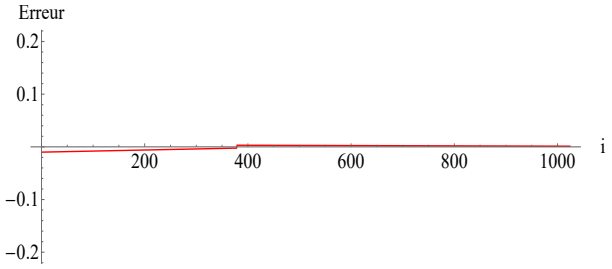
(c) TIWT, seuil doux.



(d) Erreur de l'estimé, MSE $= 33.918 \times 10^{-5}$.



(e) Algorithme proposé.



(f) Erreur de l'estimé, MSE $= 1.917 \times 10^{-5}$.

Figure 5.10 Comparaison des algorithmes sur le signal "Ramp".

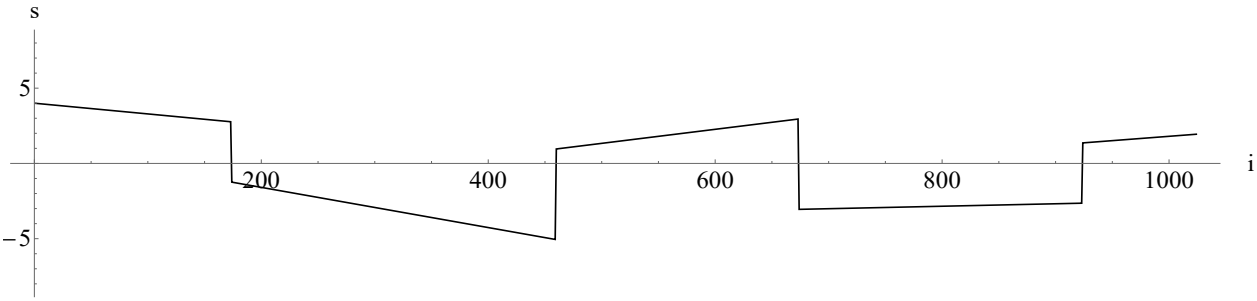
Tableau 5.5 MSE moyen sur 100 réalisations du bruit pour le signal "Ramp".

Méthode	SNR = 10 dB ($\times 10^{-4}$)	SNR = 16 dB ($\times 10^{-5}$)	SNR = 20 dB ($\times 10^{-5}$)
TIWT, seuil dur	1.6 ± 0.1	4.4 ± 0.2	1.9 ± 0.1
TIWT, seuil doux	12.9 ± 0.4	37 ± 1	15.3 ± 0.4
Algorithme proposé	1.3 ± 0.4	$\mathbf{2.6 \pm 0.6}$	$\mathbf{0.9 \pm 0.2}$

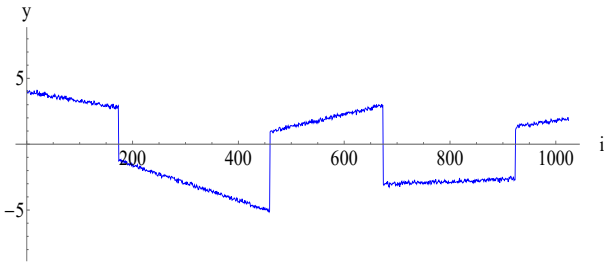
proposé obtient un MSE comparable avec celui de l'algorithme par ondelettes avec seuil dur. Pour les trois niveaux de bruit, l'algorithme proposé obtient un MSE significativement plus petit que l'algorithme par seuillage doux.

5.2.5 Résultats pour des signaux aléatoires

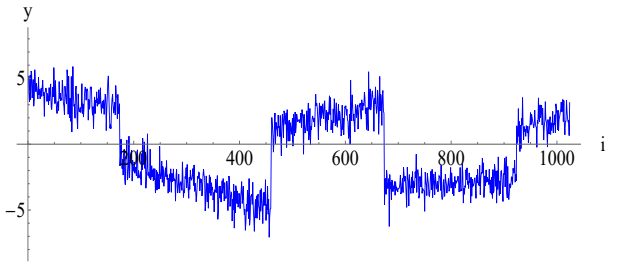
On effectue une analyse statistique des résultats des différents algorithmes en fonction du bruit ajouté sur des signaux linéaires par morceaux aléatoires. On génère des signaux linéaires par morceaux aléatoirement de la manière suivante. Chaque signal pur s est de taille $n = 1024$. La longueur des morceaux linéaires des signaux est une variable aléatoire entière uniformément distribuée entre 5 et 300. La pente de ces morceaux linéaires est aléatoire et suit une distribution $\mathcal{N}(0, 0.01)$. Entre chaque morceaux, les signaux font un saut dont la valeur est une variable aléatoire uniformément distribuée dans l'ensemble $\{-6, -4, -2, 2, 4, 6\}$. On présente un exemple d'un signal s généré par cette procédure dans la figure 5.11a. On fait varier le SNR de 30 dB à 10 dB par des pas de 2 dB. On montre une réalisation y d'un signal bruité ayant un SNR de 30 dB dans la figure 5.11b et un SNR de 10 dB dans la figure 5.11c.



(a) Signal pur.



(b) Signal bruité, $\sigma \approx 0.090$, SNR = 30 dB.



(c) Signal bruité, $\sigma \approx 0.894$, SNR = 10 dB.

Figure 5.11 Exemple d'un signal linéaire par morceaux généré aléatoirement.

On génère 100 signaux linéaires par morceaux selon la procédure décrite plus haut. La moyenne des MSE obtenus pour chaque algorithme et pour chaque valeur du SNR est présentée dans la figure 5.12. On présente également dans cette figure les résultats de la version oracle de notre algorithme, c'est-à-dire la version lors de laquelle l'analyse statistique préliminaire s'est faite à partir du signal pur s et non du signal bruité y.

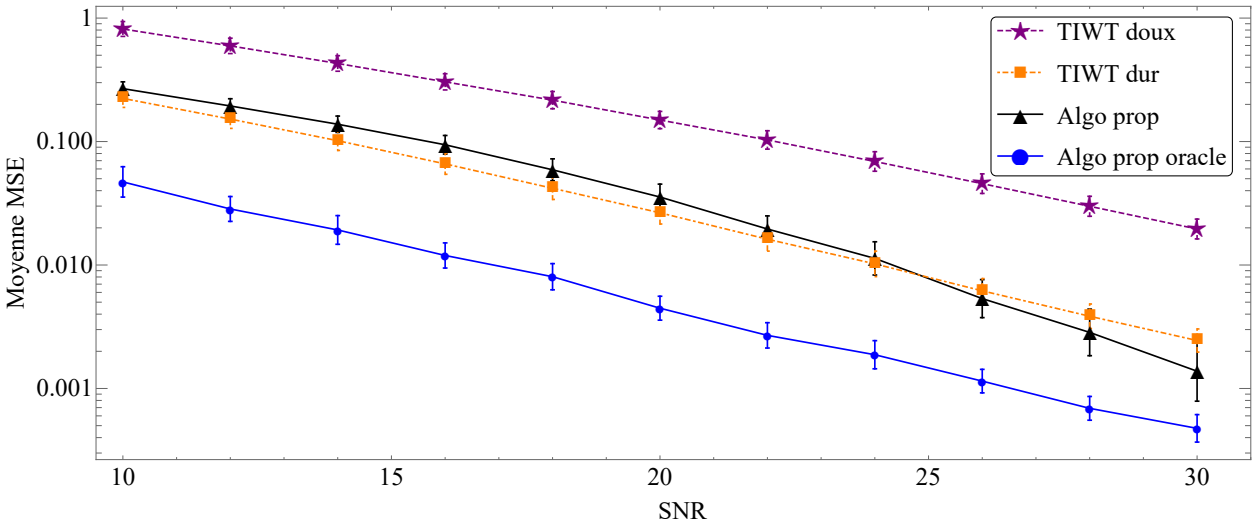


Figure 5.12 Comparaison des résultats avec les algorithmes de débruitage par ondelettes pour des signaux linéaires par morceaux générés aléatoirement.

On constate dans la figure 5.12 que le MSE moyen de notre algorithme est comparable à celui de l'algorithme par ondelettes avec seuil dur. Notre algorithme est meilleur pour tous les niveaux de bruit que l'algorithme par ondelettes avec seuil doux. Encore une fois, la version oracle obtient le MSE le plus bas pour tous les niveaux de bruit, et ne se détériore pas lorsque le bruit augmente. L'erreur de la version oracle de notre algorithme est en moyenne plus petite d'un facteur 2.3 que celle de l'algorithme par ondelettes avec le seuil dur.

5.3 Signaux lisses par morceaux

On traite maintenant du cas plus général où les signaux sont de signaux lisses par morceaux. Ces signaux sont obtenus à partir d'un échantillonnage équidistant de fonctions à valeurs dans $\mathbb{R}$ lisses par morceaux.

5.3.1 Choix des paramètres

On a construit dans la section 3.1.1 des vecteurs ayant la propriété d'être orthogonaux aux parties constantes, linéaires, quadratiques et cubiques d'un signal. On définit dans cette

section ces vecteurs de la manière suivante :

$$\begin{cases} v_1 := (-1,1)^T \in \mathbb{R}^2, \\ v_2 := (1,-2,1)^T \in \mathbb{R}^3, \\ v_3 := (1,-3,3,-1)^T \in \mathbb{R}^4, \\ v_4 := (1,-4,6,-4,1)^T \in \mathbb{R}^5. \end{cases} \tag{5.12}$$

Le choix des vecteurs v avec lesquels on effectue l'analyse statistique préliminaire est plus difficile pour des signaux lisses par morceaux plus généraux puisqu'on a moins d'information à priori. On peut par exemple choisir un sous-ensemble des vecteurs v_1 à v_4 définis en (5.12). On pourrait également ajouter des vecteurs de la forme $(-1,0...,0,1)^T$ ou $(1,0,..,0,-2,0,...,0,1)^T$ comme dans les deux dernières sections.

Les résultats présentés dans la suite sont obtenus en effectuant l'analyse statistique préliminaire avec v_1, v_2, v_3 et v_4 définis en (5.12). Encore une fois, le paramètre α du seuil de rejet de l'hypothèse d'orthogonalité est fixé à 0.05.

5.3.2 Algorithmes comparés

On compare les résultats de l'algorithme proposé pour des signaux lisses par morceaux encore une fois avec des variantes d'algorithmes de débruitage par seuillage des coefficients d'ondelettes.

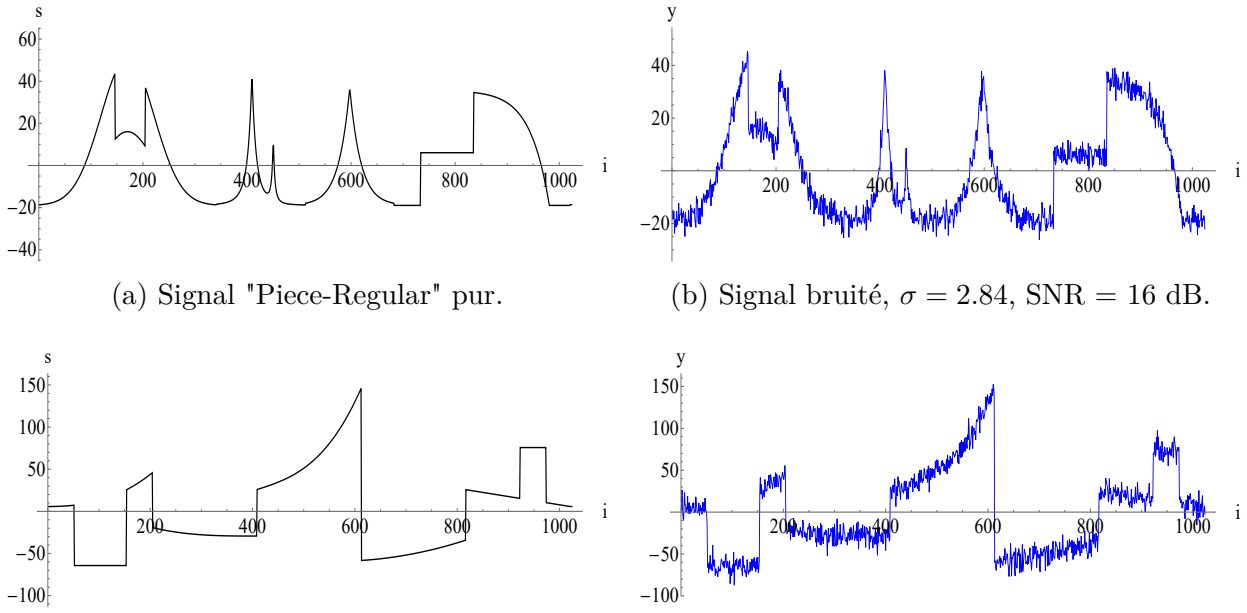
On testera dans la suite les ondelettes de Daubechies et de Symlet à quatre moments nuls, notées db4 et sym4 respectivement. Pour chacune de ces deux familles, on testera les fonctions de seuillage dur et doux. On prendra toujours dix niveaux de décomposition et le même seuil δ_{Univ} sur les composantes de y dans la base d'ondelettes sera utilisé. La méthode TIWT est toujours utilisée, avec n permutations.

5.3.3 Résultats

Deux signaux lisses par morceaux provenant de la fonction MakeSignal sont testés pour chaque algorithme. Il s'agit des signaux "Piece-Regular" et "Piece-Polynomial". Ces signaux sont de taille $n = 1024$. Un bruit additif est ajouté aux signaux purs afin d'obtenir un SNR de 16 dB dans les deux cas. On présente dans la figure 5.13 les versions pures et bruitées de ces deux signaux.

On montre dans la figure 5.14 l'estimé $\hat{s}$ produit à partir des signaux bruités des figures 5.13b et 5.13d pour l'algorithme proposé ainsi que pour la méthode par ondelettes ayant obtenu le

MSE le plus bas.



(a) Signal "Piece-Regular" pur.

(b) Signal bruité, $\sigma = 2.84$, SNR = 16 dB.

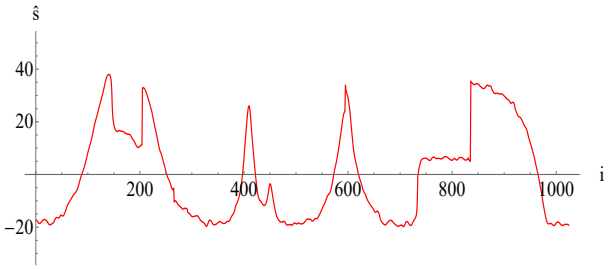
(c) Signal "Piece-Polynomial" pur.

(d) Signal bruité, $\sigma = 7.98$, SNR = 16 dB.

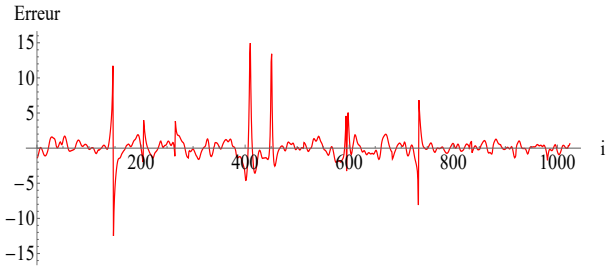
Figure 5.13 Signaux lisses par morceaux "Piece-Regular" et "Piece-Polynomial" purs et bruités.

Les estimés produits par l'algorithme proposé sont moins bons en terme de MSE que ceux produits par les méthodes d'ondelettes. Qualitativement, l'estimé souffre d'erreurs plus grandes au niveau des discontinuités (figures 5.14b et 5.14f). On note cependant que les estimés produits par les méthodes d'ondelettes souffrent également, dans une moindre mesure, d'erreurs aux niveaux des discontinuités des signaux. Ceci est encore une fois dû à l'effet de pseudo-Gibbs.

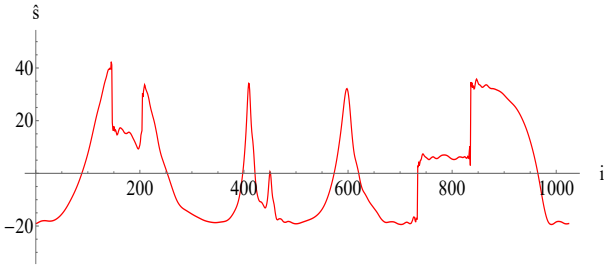
L'analyse statistique préliminaire telle que présentée ici est moins efficace dans le cas de signaux plus généraux pour préserver les discontinuités du signal. En effet, on présente dans la figure 5.15 les résultats obtenus à partir des deux mêmes signaux bruités avec la version oracle de l'algorithme proposé. Les variations abruptes entre les différentes parties du signal sont bien localisées dans chaque estimé (figures 5.15a et 5.15c). Ces estimés ne souffrent pas de défauts aux endroits des discontinuités des signaux comme pour les méthodes par ondelettes et l'erreur n'est pas plus grande au niveau de ceux-ci (figures 5.15b et 5.15d).



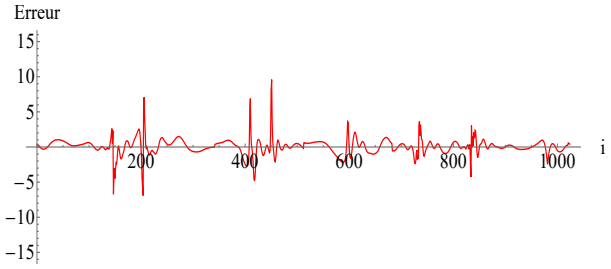
(a) Algorithme proposé.



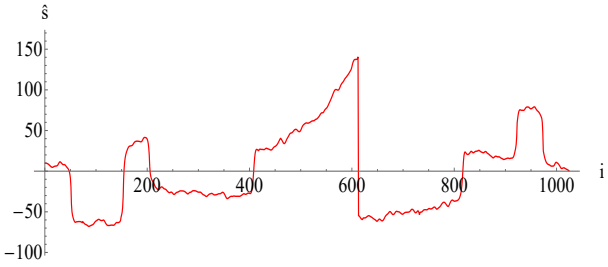
(b) Erreur de l'estimé, MSE = 2.745.



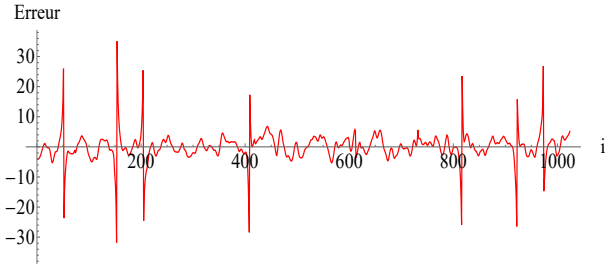
(c) TIWT db4, seuil dur.



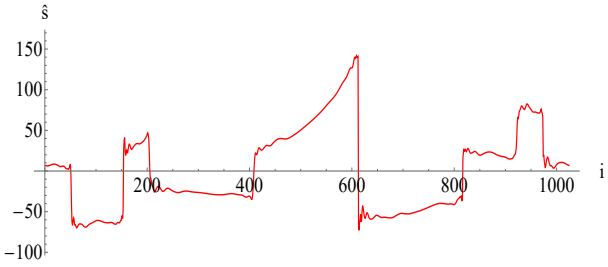
(d) Erreur de l'estimé, MSE = 1.365.



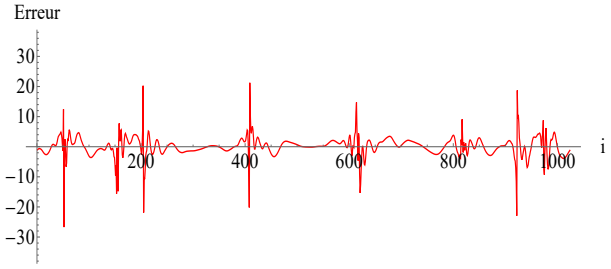
(e) Algorithme proposé.



(f) Erreur de l'estimé, MSE = 20.724.

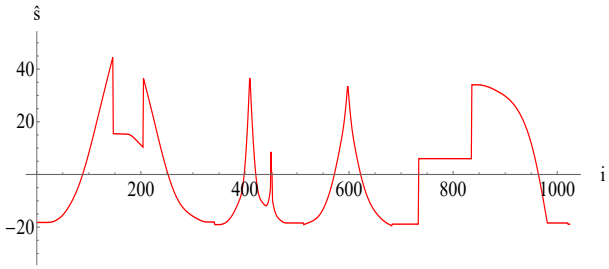


(g) TIWT db4, seuil dur.

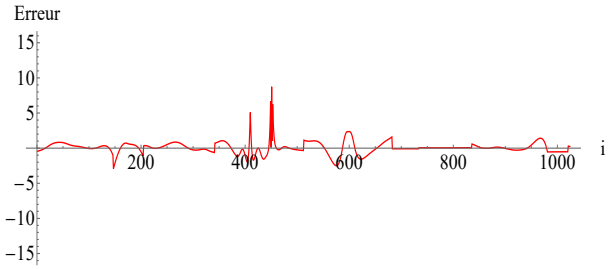


(h) Erreur de l'estimé, MSE = 10.798.

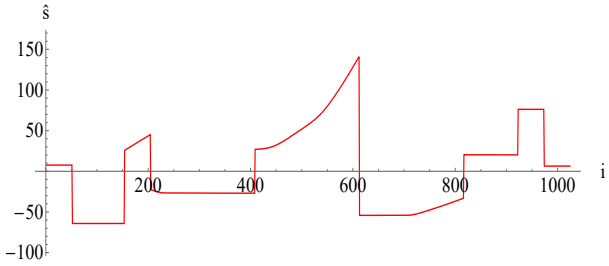
Figure 5.14 Comparaison des algorithmes sur les signaux "Piece-Regular" et "Piece-Polynomial".



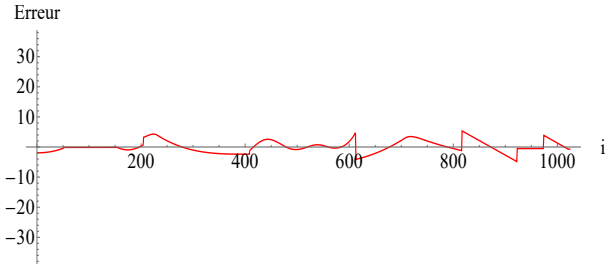
(a) Signal estimé, "Piece-Regular".



(b) Erreur de l'estimé, MSE = 0.755.



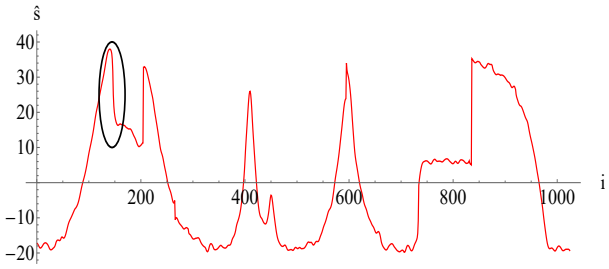
(c) Signal estimé, "Piece-Polynomial".



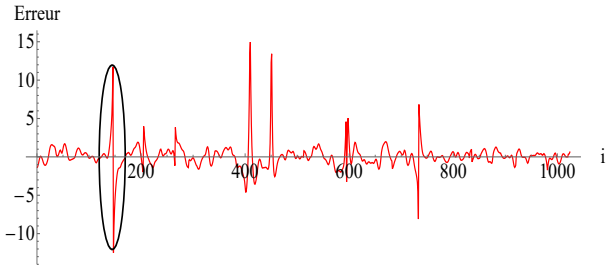
(d) Erreur de l'estimé, MSE = 3.667.

Figure 5.15 Résultats de la version oracle de notre algorithme sur les signaux lisses par morceaux.

On présente dans la figure 5.16 un exemple d'un endroit où une discontinuité du signal "Piece-Regular" a été arrondie par la version originale de notre algorithme. On a encerclé dans la figure 5.16a la discontinuité arrondie ainsi que l'erreur associée dans la figure 5.16b.



(a)



(b)

Figure 5.16 Exemple d'une discontinuité arrondie par l'algorithme proposé.

On présente dans la figure 5.17 la partie du signal "Piece-Regular" pure et bruitée correspondant à l'endroit encerclé dans la figure 5.16, c'est-à-dire entre les indices $i = 140$ et $i = 150$.

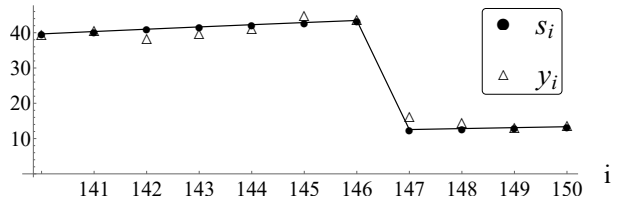


Figure 5.17 Signal "Piece-Regular" pur et bruité entre les indices 140 et 150.

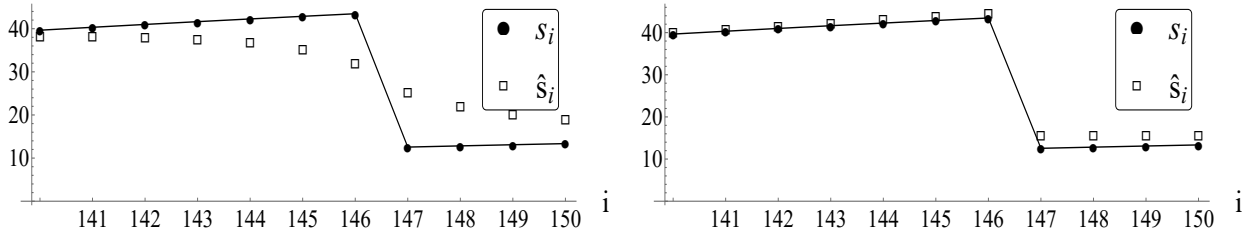
On montre dans le tableau 5.6 les résultats des tests d'orthogonalités avec les vecteurs v_1 à v_4 définis en (5.12) pour la partie du signal bruité y présentée dans la figure 5.17. Le seuil de rejet de l'hypothèse d'orthogonalité est toujours calculé selon la règle (3.11), avec $\delta_\alpha = 1.96$ puisqu'on prend $\alpha = 0.05$.

Tableau 5.6 Résultats des tests locaux pour une partie du signal "Piece-Regular".

i	$\lvert [H_{v_1} y]_i \rvert$	H_0	$\lvert [H_{v_2} y]_i \rvert$	H_0	$\lvert [H_{v_3} y]_i \rvert$	H_0	$\lvert [H_{v_4} y]_i \rvert$	H_0
	Seuil = 7.88		Seuil = 13.65		Seuil = 24.92		Seuil = 46.63	
140	0.951	Oui	3.063	Oui	6.561	Oui	10.104	Oui
141	2.113	Oui	3.498	Oui	3.542	Oui	6.054	Oui
142	1.385	Oui	0.044	Oui	2.512	Oui	10.077	Oui
143	1.341	Oui	2.468	Oui	7.564	Oui	13.471	Oui
144	3.809	Oui	5.096	Oui	21.035	Oui	72.822	Non
145	1.287	Oui	26.131	Non	51.787	Non	77.015	Non
146	27.418	Non	25.656	Non	25.28	Non	26.723	Oui
147	1.762	Oui	0.429	Oui	1.496	Oui	-	-
148	1.334	Oui	1.924	Oui	-	-	-	-
149	0.591	Oui	-	-	-	-	-	-

Pour les tests avec v_1 et v_2, il n'y a pas d'erreurs dans les hypothèses d'orthogonalité : on fait l'hypothèse que le signal pur s est orthogonal à v_1 et v_2 aux bons endroits. Pour le vecteur v_3, on accepte l'hypothèse d'orthogonalité entre $v_3 = (1, -3, 3, -1)^T$ et $(s_{144}, s_{145}, s_{146}, s_{147})^T$. On fait donc l'hypothèse que le signal pur s est quadratique à cet endroit, ce qui n'est pas le cas, voir figure 5.17. Pour le vecteur v_4, on accepte l'hypothèse d'orthogonalité entre $v_4 = (1, -4, 6, -4, 1)^T$ et $(s_{143}, s_{144}, s_{145}, s_{146}, s_{146})^T$ ainsi que $(s_{146}, s_{147}, s_{148}, s_{149}, s_{150})^T$. On fait donc l'hypothèse que le signal pur s est localement cubique à cet endroit, ce qui n'est encore une fois pas le cas, voir figure 5.17. Ces erreurs dans les hypothèses d'orthogonalités font en sorte que le signal estimé $\hat{s}$ est plus lisse que le signal pur s à cet endroit. En effet, on

pénalise les solutions qui ne sont pas localement quadratiques et cubiques aux endroits où on a considéré que le signal pur s l'était. On présente dans la figure 5.18 le signal estimé par notre algorithme dans sa version originale et sa version oracle entre les mêmes indices $i = 140$ et $i = 150$. On remarque dans la figure 5.18a que l'estimé produit par notre algorithme atténue la variation abrupte présente dans le signal pur. L'estimé produit par la version oracle de l'algorithme, figure 5.18b, ne fait pas cette erreur.



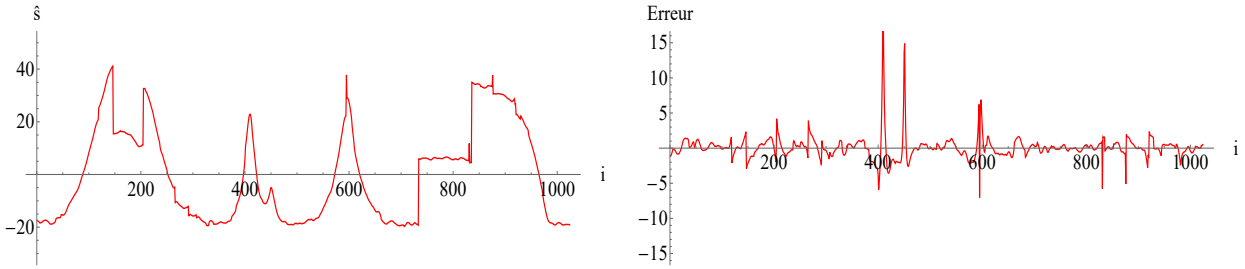
(a) Partie du signal estimé, version originale.

(b) Partie du signal estimé, version oracle.

Figure 5.18 Résultats de l'algorithme proposé sur une partie du signal "Piece-Regular".

Suite à cette analyse des résultats de notre algorithme sur une partie du signal lisse par morceaux "Piece-Regular", on peut suggérer de faire l'analyse statistique préliminaire seulement avec les vecteurs v_1 et v_2 de la définition (5.12). En effet, ce sont les tests locaux avec les vecteurs v_3 et v_4 qui semblent avoir créé les erreurs plus grandes au niveau des discontinuités du signal. On montre dans la figure 5.19 le résultat de notre algorithme en effectuant l'analyse statistique préliminaire seulement avec v_1 et v_2. Le résultat est comparable à l'estimé produit par notre algorithme avec les vecteurs v_1 à v_4 (figures 5.14a et 5.14b). Le MSE est légèrement plus petit dans ce dernier cas. On constate dans la figure 5.19b des erreurs entre les indices $i = 800$ et $i = 1000$. Le signal pur est quadratique entre ces indices. Ces erreurs sont dues au fait qu'on n'a pas testé l'orthogonalité au vecteur v_3 lors dans l'analyse statistique préliminaire.

Le choix des vecteurs v avec lesquels on effectue l'analyse statistique préliminaire est important et celui-ci influence le résultat de notre algorithme. On a proposé ici un choix naturel pour des signaux lisses par morceaux, c'est-à-dire les vecteurs v_1 à v_4 définis en (5.12). Cependant, d'autres choix pourraient être faits et il est possible que ces choix puissent améliorer l'estimé produit par notre algorithme.



(a) Signal estimé.

(b) Erreur de l'estimé, MSE = 2.771.

Figure 5.19 Résultat de l'algorithme proposé avec v_1 et v_2.

On présente dans le tableau 5.7 les moyennes des MSE de chaque algorithme sur 100 réalisations du bruit pour des SNR de 10, 16 et 20 dB pour les deux signaux tests 1-D "Piece-Regular" et "Piece-Polynomial". Des exemples de ces deux signaux dégradés ayant des SNR de 10 et 20 dB peuvent être trouvés dans l'annexe B. Les résultats de l'algorithme proposé sont obtenus en effectuant l'analyse statistique préliminaire avec les vecteurs v_1 à v_4. Dans le cas des méthodes par ondelettes, on prend la famille et la fonction de seuillage ayant obtenus le MSE moyen le plus bas. L'algorithme proposé obtient en moyenne une erreur plus grande d'un facteur 1.4 pour le signal "Piece-Regular" et d'un facteur 1.3 pour le signal "Piece-Polynomial" par rapport à l'algorithme par ondelettes. La version oracle de l'algorithme est celle qui donne les meilleurs résultats. L'erreur est diminuée d'un facteur 1.3 pour "Piece-Regular", 1.7 pour "Piece-Polynomial" par rapport à l'algorithme par ondelettes en moyenne.

Tableau 5.7 MSE moyen sur 100 réalisations du bruit pour les signaux "Piece-Regular" et "Piece-Polynomial".

Méthode	SNR = 10 dB	SNR = 16 dB	SNR = 20 dB
Signal "Piece-Regular"			
Algorithme proposé	6.7 ± 0.2	2.51 ± 0.06	1.19 ± 0.02
Algorithme proposé oracle	2.1 ± 0.2	0.69 ± 0.04	0.32 ± 0.02
db4, seuil dur	$\mathbf{4.8 \pm 0.1}$	$\mathbf{1.16 \pm 0.03}$	$\mathbf{0.48 \pm 0.01}$
Signal "Piece-Polynomial"			
Algorithme proposé	59 ± 2	19.9 ± 0.5	8.8 ± 0.2
Algorithme proposé oracle	13 ± 1	3.7 ± 0.3	1.4 ± 0.1
db4, seuil dur	$\mathbf{42 \pm 1}$	$\mathbf{11.2 \pm 0.2}$	$\mathbf{4.1 \pm 0.1}$

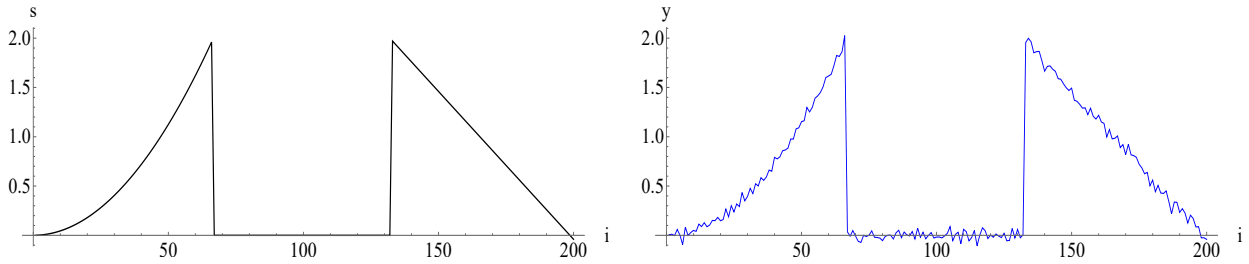
5.3.4 Remarque sur la pondération des contraintes

Il est possible de pondérer l'importance relative entre les différentes contraintes associées aux vecteurs v_k dans la fonction objectif du problème (3.16). Étant donnée une analyse statistique préliminaire effectuée sur un signal y avec un ensemble de vecteurs $\{v_1, ..., v_M\}$, on peut définir le modèle composite D comme étant

$$D := \begin{pmatrix} p_1 D_{v_1} \\ p_2 D_{v_2} \\ \vdots \\ p_M D_{v_M} \end{pmatrix}, \tag{5.13}$$

où $p_k \in \mathbb{R}$. On a toujours considéré $p_k = 1$ jusqu'à maintenant. En changeant ces pondérations, on peut donner plus ou moins d'importance à la contrainte associée à un vecteur v_k dans la fonction objectif.

On présente un exemple sur des signaux s de taille $n = 200$ comportant une partie constante, une partie linéaire et une partie quadratique. Un exemple d'un tel signal est présenté dans la figure 5.20a. On ajoute un bruit d'écart-type $\sigma = 0.04$ à ces signaux (figure 5.20b).



(a) Signal pur.

(b) Signal bruité, $\sigma = 0.04$, SNR = 23.97 dB.

Figure 5.20 Signal ayant une partie constante, une partie linéaire et une partie quadratique.

On génère 100 signaux semblables au signal de la figure 5.20a en faisant varier aléatoirement de manière uniforme l'indice de la première discontinuité entre $i = 20$ et $i = 80$ et l'indice de la seconde discontinuité entre $i = 120$ et $i = 160$. On utilise les vecteurs $v_1 = (-1, 1)^T$, $v_2 = (1, -2, 1)^T$ et $v_3 = (1, -3, 3, -1)^T$ dans l'analyse statistique préliminaire. On fixe p_1 et p_2 à 1 dans le modèle composite D défini en (5.13) et on fait varier p_3 de 0 à 2 par des pas de 0.05. Dans la figure 5.21, on présente le MSE moyen pour chaque valeur de p_3 sur les 100 signaux auxquels un bruit d'écart-type 0.04 a été ajouté. On remarque dans la figure 5.21 que les résultats sont significativement moins bons pour des valeurs de p_3 entre 0 et 0.5, puis

sont comparables entre $p_3 = 0.5$ et $p_3 = 2$. Des minimums locaux sont atteints en $p_3 = 0.3$, 1.05 et 1.55. Le fait qu'un grand nombre de valeurs de poids p_k donne des résultats similaires en termes de MSE et que certaines valeurs isolées semblent être significativement meilleures (ici en $p_3 = 1.55$) est une caractéristique de notre algorithme que nous avons remarquée pour plusieurs types de signaux. Nous avons constaté empiriquement que la position des minimum locaux du MSE était très sensible à la famille de signaux considérée, en ce sens que de petits changements des signaux (par exemple la position ou le nombre de discontinuités) entraînent typiquement de grands changements des minimum locaux. Cette sensibilité des minimums locaux aux signaux, que nous avons observée dans plusieurs expériences numériques impliquant des signaux de complexité variable, nous a conduit à penser que l'optimisation des poids p_k était un problème difficile. L'effet des p_k dans (5.13) sur la solution $\hat{s}$ produite par notre algorithme et la façon de les optimiser étant donné un certain type de signal demeure une voie à explorer.

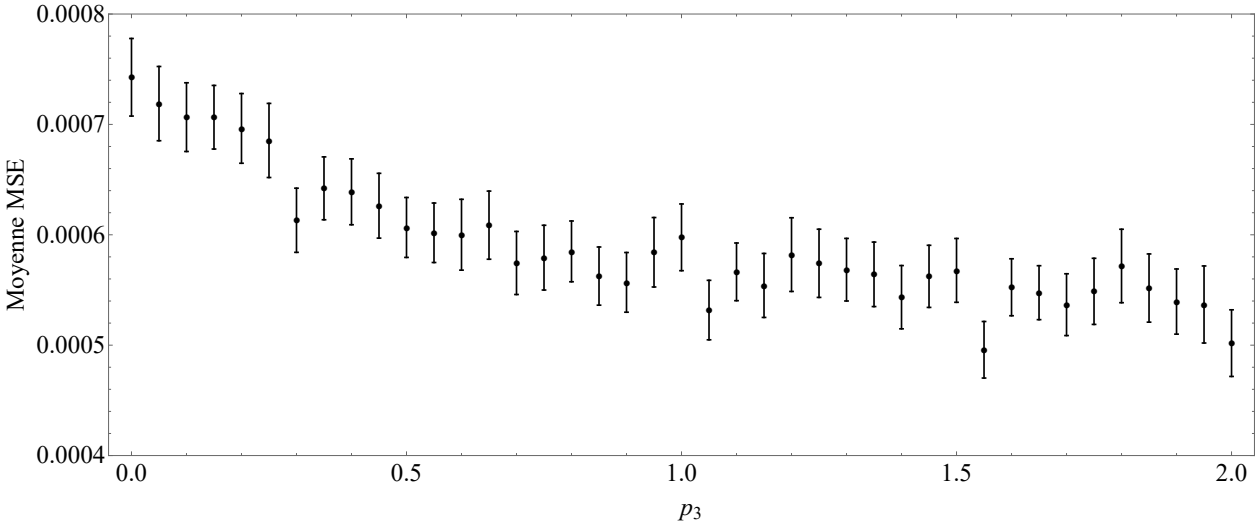


Figure 5.21 Exemple de l'effet de la pondération des contraintes sur le MSE de notre méthode.

5.4 Discussion sur les résultats de la version 1-D de l'algorithme

L'algorithme de débruitage proposé dans ce mémoire introduit un nouveau cadre de travail pour le problème de débruitage de signaux numériques. Ce cadre permet une certaine flexibilité puisqu'on peut choisir comment diriger l'analyse statistique préliminaire en choisissant les vecteurs v avec lesquels on effectue celle-ci. Lorsqu'on possède des connaissances sur le types de signaux à débruiter, on peut choisir les vecteurs v en conséquences. On a vu dans les sections 5.1 et 5.2 deux exemples de cette approche pour des signaux constants et linéaires

par morceaux. Dans ces deux cas, l'algorithme proposé obtient des résultats compétitifs avec les méthodes de débruitage par variation totale et par seuillage des coefficients d'ondelettes. Cette flexibilité a cependant un inconvénient. Le choix des vecteurs v avec lesquels on effectue l'analyse statistique préliminaire n'est pas aisé lorsqu'on a peu d'information sur les signaux à débruiter, par exemple pour des signaux lisses par morceaux comme ceux présentés dans la section 5.3. Dans ce cas, les résultats obtenus avec notre algorithme sont moins bons qualitativement. Certaines variations abruptes dans les signaux purs sont atténuées dans le signal estimé.

Théoriquement, lorsque l'analyse statistique préliminaire permet la construction d'une matrice D bien adaptée à un signal s donné, les résultats sont qualitativement et quantitativement très bons, comme montré grâce à la version oracle de l'algorithme (voir la figure 5.15 et le tableau 5.7). Il est donc permis d'espérer de meilleurs résultats grâce à des améliorations futures de l'algorithme.

CHAPITRE 6 RÉSULTATS POUR DES SIGNAUX 2-D

On présente dans ce chapitre les résultats de la version 2-D de notre algorithme, c'est-à-dire pour le débruitage d'images. Ces résultats seront comparés avec d'autres algorithmes de débruitage d'images de la littérature.

On comparera encore les performances de chaque algorithme grâce au MSE. Dans le cas 2-D, s et y sont des matrices $\mathbb{R}^{l\times l}$. On utilise dans la définition (5.1) la norme de Frobenius et $n = l \times l$.

Une autre méthode permettant de comparer la performance de différents algorithmes de débruitage d'images est ce qu'on appelle l'analyse du résidu d'un estimé $\hat{s}$. Ce résidu est défini comme suit :

$$\text{résidu } = y - \hat{s} \ \in \mathbb{R}^{l\times l}. \tag{6.1}$$

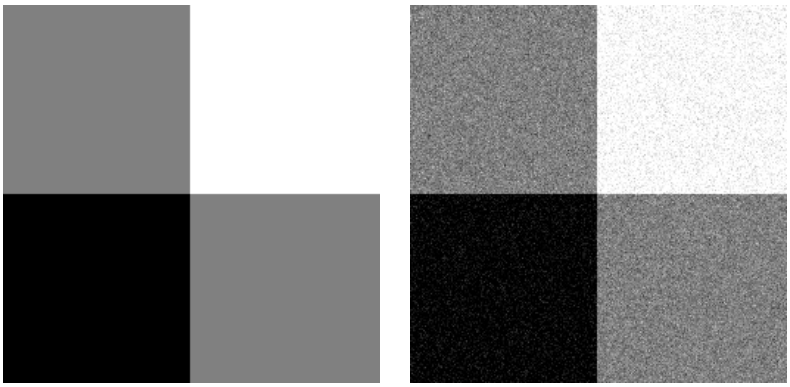
Si $\hat{s}$ est un estimé parfait de s, on a que $\hat{s} = s$. Dans ce cas, le résidu de cet estimé est

$$\begin{aligned} \text{résidu } &= y - \hat{s}, \\ &= (s + w) - s, \\ &= w. \end{aligned}$$

Ainsi, le résidu d'un estimé $\hat{s}$ devrait avoir la structure d'un bruit blanc. Si ce n'est pas le cas, alors on en déduira qu'on a introduit des erreurs en estimant l'image s avec $\hat{s}$. La méthode de l'analyse du résidu d'un estimé est introduite et détaillée dans l'article de Buades et al. (2005).

6.1 Image synthétique

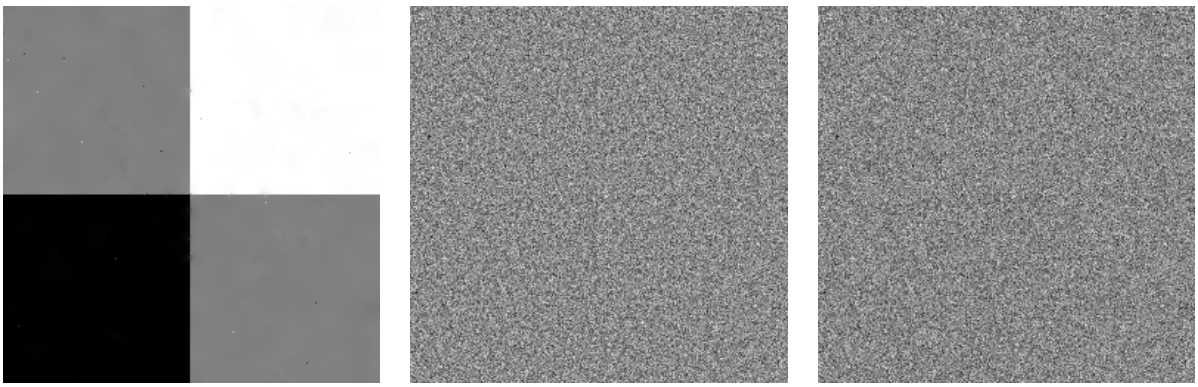
On teste tout d'abord notre algorithme sur une image synthétique constante par morceaux. Comme dans le cas 1-D, lorsqu'on a une connaissance à priori du type d'image à débruiter, on peut adapter notre algorithme en choisissant les vecteurs v avec lesquels on effectue l'analyse statistique préliminaire. On présente dans la figure 6.1 une image synthétique de dimension 256×256 constante par morceaux avec quatre morceaux. Les pixels des parties blanches et noires de l'image ont respectivement une valeur de 255 et 0, les pixels des parties grises ont une valeur de 128. On ajoute à cette image un bruit blanc d'écart-type $\sigma = 20$.



(a) Image pure $s \in \mathbb{R}^{256\times256}$. (b) Image bruitée y, $\sigma = 20$.

Figure 6.1 Image constante par morceaux pure et bruitée.

On teste la version 2-D de notre algorithme sur l'image bruitée de la figure 6.1b. On fait l'hypothèse que l'on sait que l'image pure sous-jacente est constante par morceaux. On effectue grâce à cette connaissance l'analyse statistique préliminaire de l'image bruitée avec le vecteur $v = (-1, 1)^T$ associé aux contraintes constantes. Le seuil α demeure à 0.05. La taille des fenêtres avec lesquelles on découpe l'image est fixée à $m = 16$ et le paramètre h pour la décroissance du noyau gaussien dans la pondération des différents estimés dans chaque fenêtre est fixé à $h = m/4 = 4$. On montre dans la figure 6.2a l'estimé $\hat{s}$ de l'image et dans la figure 6.2b le résidu de cet estimé. Dans la figure 6.2c, on présente le bruit blanc w [1].



(a) Image débruitée $\hat{s}$. (b) Résidu, MSE = 4.21. (c) Bruit ajouté w.

Figure 6.2 Résultats de l'algorithme pour une image constante par morceaux.

Visuellement, l'image débruitée produite par notre algorithme, présentée dans la figure 6.2a, est une bonne approximation de l'image pure (figure 6.1a). En regardant de près, on constate

1. Pour les images des résidus et du bruit, on recale les valeurs des pixels entre 0 et 255 pour la visualisation.

cependant que la valeur de quelques pixels isolés est mal estimée par notre algorithme (des pixels noirs ou blancs sont entre autres visibles dans les parties grises de l'image). Le résidu de l'estimé, présenté dans la figure 6.2b, est visuellement similaire au bruit blanc ajouté à l'image pure (figure 6.2c). Cette observation signifie qu'on a fait très peu d'erreurs en estimant l'image s par $\hat{s}$. L'espérance de la distance moyenne au carré entre l'image bruitée y et l'image pure s est donnée par $\sigma^2 = 400$. La distance moyenne au carré entre l'image estimée $\hat{s}$ et l'image pure s est donnée par MSE $= 4.21$. L'erreur moyenne est donc presque réduite d'un facteur 10 grâce à l'estimé $\hat{s}$ par rapport à l'image bruitée y ($\sqrt{\frac{400}{4.21}} \approx 10$).

6.2 Images naturelles

On montre dans cette section des résultats de la version 2-D de notre algorithme appliqué à des images naturelles. On compare ces résultats avec quelques algorithmes de débruitage classiques de la littérature. On teste les algorithmes sur cinq images tests provenant d'une banque d'images souvent utilisée dans le domaine du débruitage d'images. Ces cinq images sont présentées dans la figure 6.3.

6.2.1 Choix des paramètres

Pour des images naturelles, tout comme pour des signaux lisses par morceaux en une dimension, on propose dans notre algorithme d'utiliser les vecteurs v_1 à v_4 associés aux contraintes constantes, linéaires, quadratiques et cubiques définis en (5.12). On prendra encore une fois le seuil de rejet $\alpha = 0.05$. On utilise des fenêtres carrées de dimension $m = 16$ pour le découpage des images et le paramètre h du noyau gaussien est fixé à $m/4 = 4$.

6.2.2 Résultats

On compare les résultats de notre algorithme en termes de MSE avec d'autres algorithmes de débruitage d'images dans la littérature. Une présentation de ces algorithmes a été faite dans le chapitre 2. Les algorithmes comparés sont : la méthode de variation totale originale (VT) (Rudin et al., 1992), le filtre bilatéral (FB) (Tomasi et Manduchi, 1998), l'algorithme *nonlocal means* (NLM) (Buades et al., 2005), une version itérative de l'algorithme *nonlocal means* (NLM IT) (Kervrann et Boulanger, 2006) et l'algorithme *Block Matching 3-D* (BM3D) (Dabov et al., 2007). On trouve dans le tableau 6.1 la valeur du MSE pour chaque algorithme et pour chaque image de la figure 6.3 auxquelles un bruit d'écart-type $\sigma = 20$ a été ajouté.



(a) Lena 512×512.



(b) Barbara 512×512.



(c) Boats 512×512.



(d) House 256×256.



(e) Peppers 256×256.

Figure 6.3 Images tests non bruitées.

Ces valeurs sont celles reportées par Kervrann et Boulanger (2006) [2] sauf pour l'algorithme *Block Matching 3-D*. Pour ce dernier algorithme les valeurs du MSE sont celles trouvées dans l'article original de Dabov et al. (2007).

Tableau 6.1 MSE des algorithmes de débruitage pour les images tests bruitées, $\sigma = 20$.

Image	Lena	Barbara	Boats	House	Peppers
Algorithme proposé	56.28	132.52	81.97	59.64	94.54
VT	58.22	127.67	81.49	51.30	91.64
FB	61.65	129.15	93.77	64.88	84.16
NLM	50.59	75.00	87.92	45.61	80.92
NLM IT	35.41	59.71	63.25	33.35	56.77
BM3D	**32.22**	**43.86**	**53.10**	**27.29**	**48.31**

2. Tableau I page 2875. Les algorithmes sont comparés dans ce tableau selon la valeur du *peak signal-to-noise ratio* (PSNR). Cependant, on peut extraire la valeur du MSE à partir du PSNR : $\text{PSNR} = 10 \log_{10} \frac{255^2}{MSE}$.

Notre algorithme de débruitage n'obtient pas de meilleurs résultats en termes de MSE que l'état-de-l'art pour des images naturelles, c'est-à-dire l'algorithme B3MD. L'erreur est en moyenne plus grande d'un facteur 1.4 par rapport à celui-ci. Bien qu'en moyenne un peu plus élevé, le MSE de notre algorithme est comparable à ceux des algorithmes VT, FB et NLM. Dans le cas de l'image Lena on obtient des valeurs de MSE plus basses que les algorithmes VT et FB. Le temps de calcul est en moyenne de 10 minutes pour les images 512×512 (Lena, Barbara, Boats) et de 2.5 minutes pour les images 256×256 (House, Peppers).

On présente dans les figures 6.4 à 6.8 les résultats de notre algorithme appliqué aux cinq images tests. On discute ici des résultats obtenus pour l'image Lena, présentés dans la figure 6.4. On peut faire des remarques analogues pour les quatre autres images. L'image estimée par notre algorithme, présentée dans la figure 6.4c, a visuellement réduit le bruit présent dans l'image bruitée de la figure 6.4b. Le MSE obtenu étant d'environ 56, l'erreur moyenne entre $\hat{s}$ et s est réduite d'un facteur 2.8 par rapport à l'espérance de l'erreur entre y et s. Cependant, on peut voir dans le résidu de l'estimé (figure 6.4d) un fantôme de l'image originale. Cette observation signifie que l'algorithme a commis des erreurs lors de l'estimation de l'image. Les contours ne sont malheureusement pas bien préservés dans l'estimé $\hat{s}$. Dans la section 5.3, on a discuté des difficultés rencontrées pour préserver les discontinuités de signaux 1-D lisses par morceaux avec notre algorithme. Les discontinuités d'un signal 2-D peuvent être associées aux contours de l'image. L'image estimée par la version 2-D de notre algorithme est sujette au même type d'erreurs que dans le cas 1-D.

Cependant, le fait que notre algorithme donne de moins bons résultats pour des images naturelles n'est pas seulement dû, comme pour les signaux 1-D lisses par morceaux présentés dans la section 5.3, à des erreurs lors de l'analyse statistique préliminaire du signal. En effet, on peut considérer que les cinq images testées plus haut ne respectent pas l'hypothèse faite sur les signaux purs pour le développement de notre algorithme, à savoir que ceux-ci doivent être lisses par morceaux. On présente dans la figure 6.9a les valeurs des pixels sur la première ligne de l'image Lena[3] (figure 6.4a). Même aux endroits où l'image pure semble homogène, c'est-à-dire par exemple sur le mur du fond de l'image Lena, entre les indices $i = 100$ et $i = 300$ de la figure 6.9a, la valeur des pixels varie toujours un peu. Ainsi, même les parties constantes de l'image pure ne sont pas réellement constantes, et l'image pure est elle-même un peu bruitée initialement. On cherche, lors de l'analyse statistique préliminaire du signal bruité, à déterminer les endroits où le signal pur est plausiblement constant, linéaire, quadratique, etc. Si le signal pur ne possède pas ces caractéristiques, on doit remettre en question la façon de procéder dans notre algorithme. Tel que présenté dans ce mémoire, l'algorithme est construit

3. La première ligne d'une image est celle du haut, par convention.

à partir de l'hypothèse que les signaux sont lisses par morceaux. Des modifications devraient être apportées à l'algorithme pour pouvoir donner de meilleurs résultats dans le cas d'images naturelles qui ne respectent pas cette hypothèse.



(a) Image non bruitée.



(b) Image bruitée, $\sigma = 20$.



(c) Image estimée.

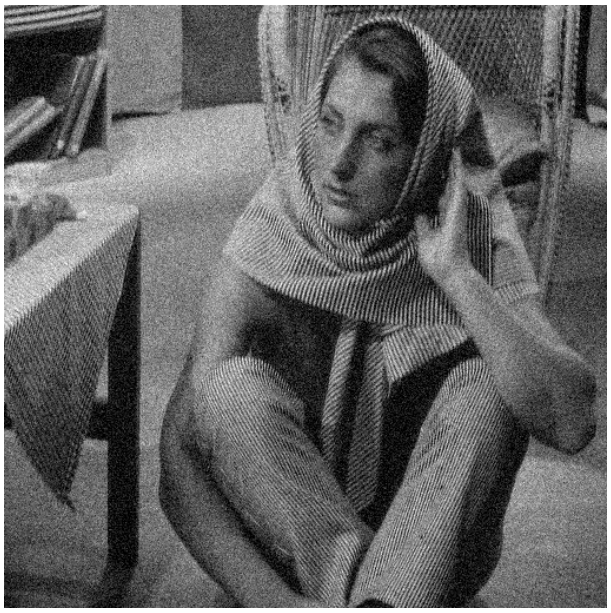


(d) Résidu, MSE = 56.28.

Figure 6.4 Résultats de l'algorithme proposé pour l'image Lena.



(a) Image non bruitée.



(b) Image bruitée, $\sigma = 20$.



(c) Image estimée.



(d) Résidu, MSE = 132.52.

Figure 6.5 Résultats de l'algorithme proposé pour l'image Barbara.



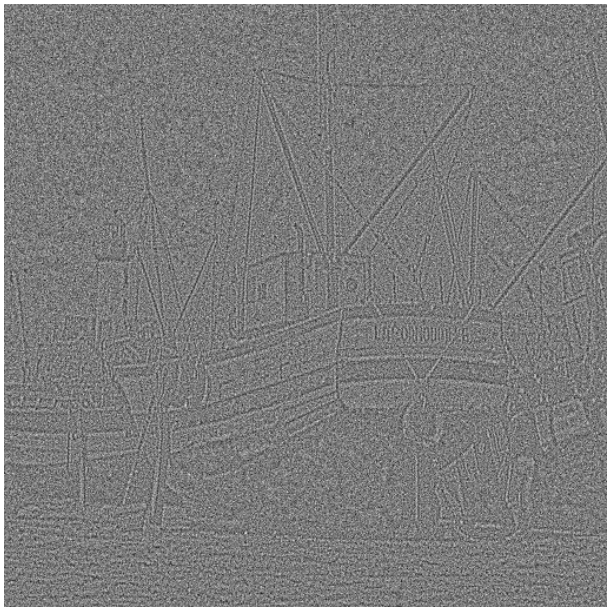
(a) Image non bruitée.



(b) Image bruitée, $\sigma = 20$.



(c) Image estimée.



(d) Résidu, MSE = 81.97.

Figure 6.6 Résultats de l'algorithme proposé pour l'image Boats.



(a) Image non bruitée.



(b) Image bruitée, $\sigma = 20$.



(c) Image estimée.

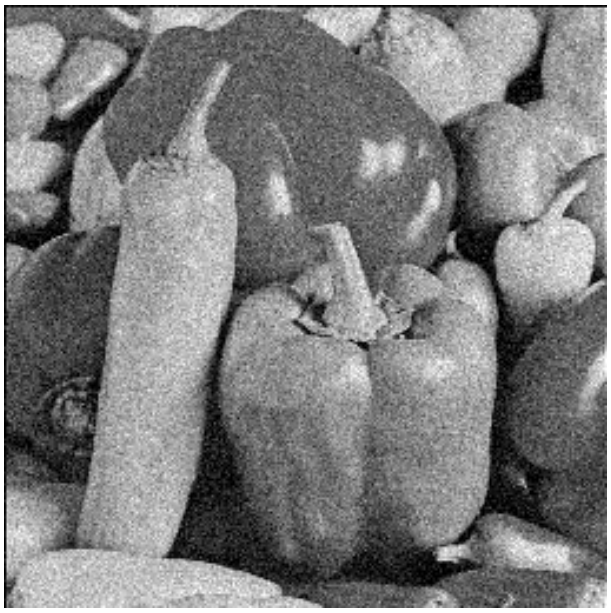


(d) Résidu, MSE = 59.64.

Figure 6.7 Résultats de l'algorithme proposé pour l'image House.



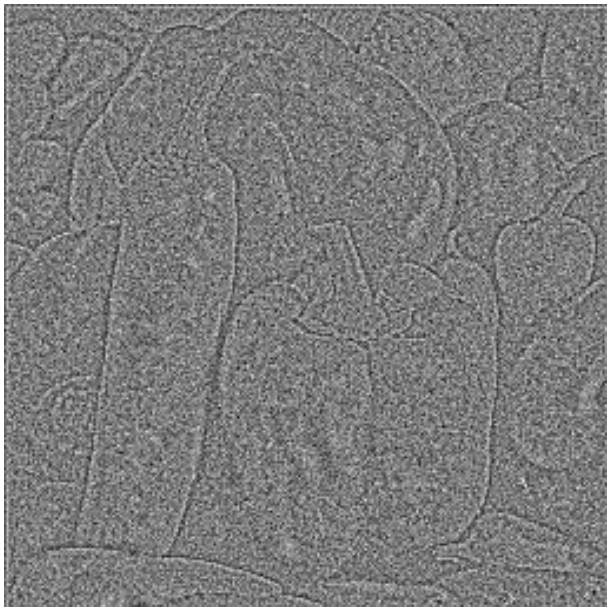
(a) Image non bruitée.



(b) Image bruitée, $\sigma = 20$.

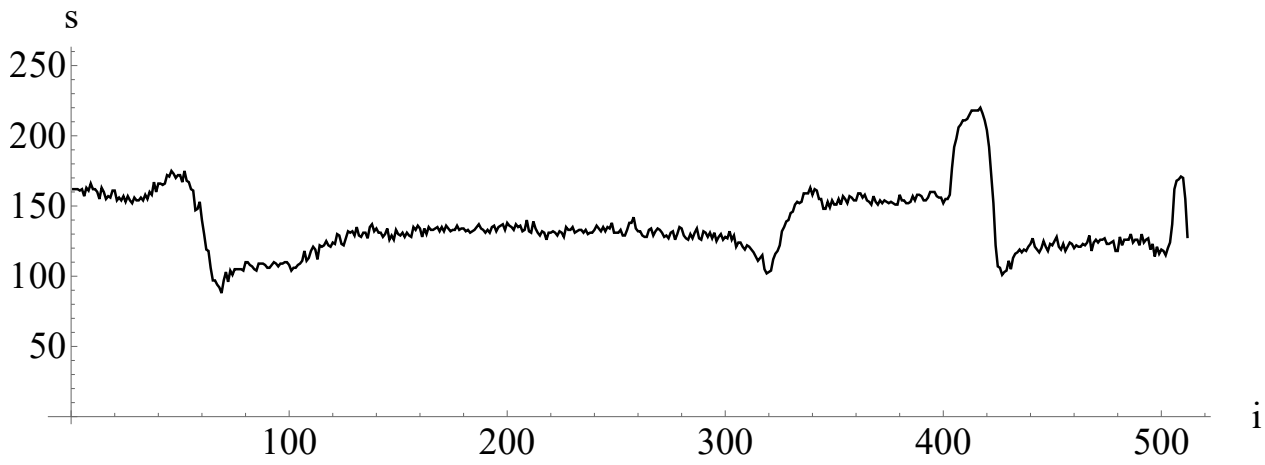


(c) Image estimée.

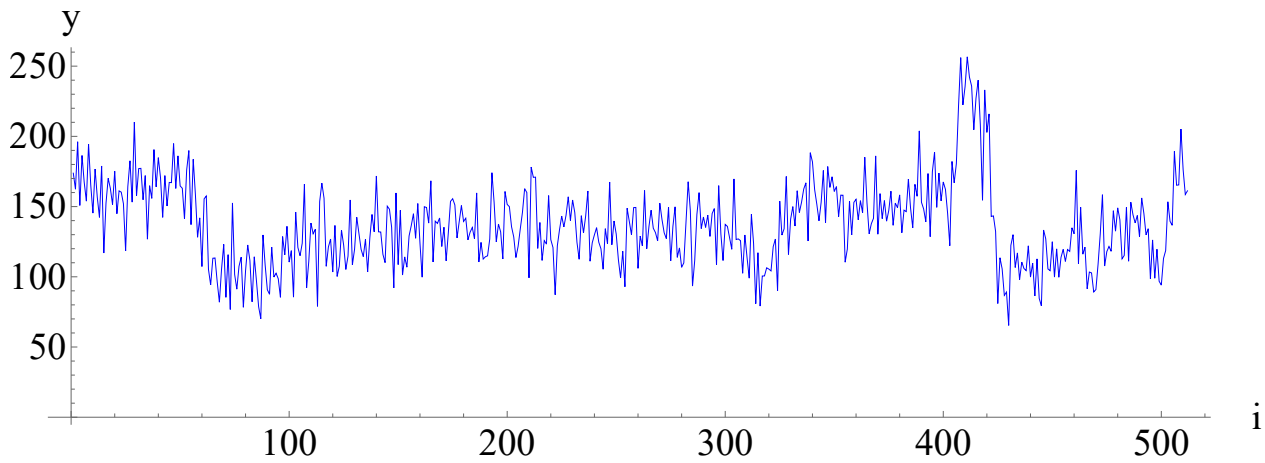


(d) Résidu, MSE = 94.54.

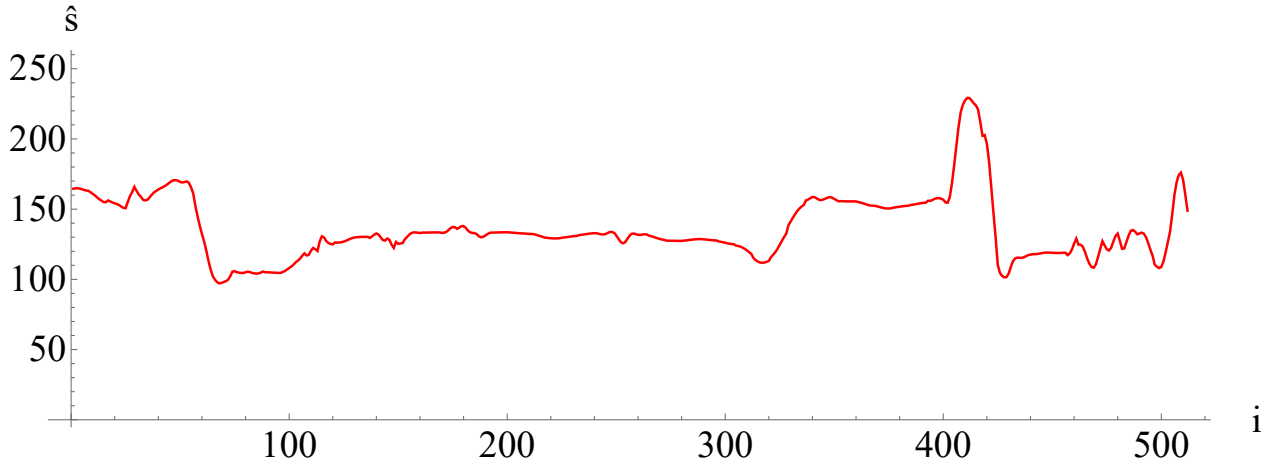
Figure 6.8 Résultats de l'algorithme proposé pour l'image Peppers.



(a) Signal pur.



(b) Signal bruité, $\sigma = 20$.



(c) Signal estimé.

Figure 6.9 Première ligne de l'image Lena.

On présente dans la figure 6.9b la première ligne de l'image Lena bruitée de la figure 6.4b. On présente également dans la figure 6.9c la première ligne de l'image estimée par notre algorithme de la figure 6.4c. En comparant la figure 6.9c et la figure 6.9b, on constate encore une fois que notre estimé a permis de diminuer le bruit. Cependant, on remarque aussi que notre algorithme a atténué certaines variations abruptes présentes dans le signal pure en comparant les figures 6.9c et 6.9a. Par exemple, aux indices $i \approx 60$ et $i \approx 320$, le signal pur de la figure 6.9a possède des creux qui sont moins prononcés dans le signal estimé de la figure 6.9c. C'est ce type d'erreurs qui créer le fantôme de l'image originale dans le résidu de l'estimé (figure 6.4d). Les transitions abruptes dans une image ne sont pas nécessairement faites sur deux pixels consécutifs. En effet, même si un contour d'une image peut paraître net à vue d'oeil, il est possible que le changement dans les valeurs des pixels se fasse sur plus de deux pixels consécutifs. Lorsque le bruit est élevé, comme dans la figure 6.9b, les tests statistiques locaux sur 2 à 5 points sont moins efficaces pour détecter ces transitions dans le signal pur. Une avenue intéressante d'amélioration pour notre algorithme serait que la taille des parties du signal sur lesquelles on effectue les tests locaux puisse varier. Pour des niveaux de bruit plus grand, des tests statistiques sur plus de points seraient davantage robuste au bruit et permettrait éventuellement de mieux détecter les transitions abruptes dans le signal pur.

CHAPITRE 7 CONCLUSION

7.1 Synthèse des travaux

On a présenté dans ce mémoire un nouveau cadre de travail pour le débruitage de signaux numériques. Celui-ci est basé sur une analyse statistique locale des signaux bruités y. Cette analyse a pour but de modéliser les caractéristiques locales d'un signal afin de préserver celles-ci lors du processus de débruitage. L'analyse statistique est conceptuellement simple, elle est effectuée au moyen de produits scalaires entre des vecteurs, qu'on a notés v, et toutes les parties du signal de même taille que ces vecteurs v. Selon la valeur de ces produits scalaires, des hypothèses sur les caractéristiques locales du signal pur sont faites. On a appelé ces hypothèses des modèles locaux du signal. Les modèles locaux sont rassemblés dans un modèle dit composite et sont stockés dans une matrice notée D. Ce modèle composite est intégré dans un problème d'optimisation avec contrainte impliquant tout le signal. La solution de ce problème d'optimisation constitue l'estimé $\hat{s}$ du signal pur s. Le problème d'optimisation pénalise les solutions s'écartant des modèles locaux obtenus lors de l'analyse statistique préliminaire. L'algorithme est donc adaptatif, le problème d'optimisation dépend du signal d'entrée. On résout le problème d'optimisation de manière analytique. L'algorithme a d'abord été présenté pour des signaux 1-D, on a ensuite généralisé celui-ci pour des signaux 2-D.

On a comparé les performances de l'algorithme proposé dans ce mémoire avec d'autres algorithmes de débruitage dans la littérature. Pour des signaux 1-D simples, à savoir constants ou linéaires par morceaux, il est possible de diriger l'analyse statistique préliminaire afin que l'algorithme proposé donne de très bons résultats, compétitifs avec d'autres méthodes de débruitage, à savoir des méthode de variation totale et par seuillage des coefficients dans une base d'ondelettes. Pour des signaux lisses par morceaux plus généraux, la façon de diriger l'analyse statistique préliminaire, entre autres le choix des vecteurs v, est un problème difficile. L'estimé produit avec les choix que nous avons proposés dans ce mémoire arrondit certaines transitions abruptes présentes dans le signal pur.

Dans le cas 2-D, pour des images naturelles, les choix qu'on doit faire pour diriger l'analyse statistique préliminaire sont également difficiles. De plus, les images naturelles ont une structure plus complexe que les signaux 1-D synthétiques sur lesquels l'algorithme a été testé initialement. En effet, celles-ci ne sont pas des signaux lisses par morceaux. De plus, les discontinuités dans une image ne sont pas aussi nettes que dans le cas des signaux 1-D synthétiques testés. L'estimé produit par notre algorithme pour des images naturelles souffre d'erreurs au niveau des contours de l'image.

7.2 Améliorations futures

Le cadre de travail pour le problème de débruitage introduit dans ce mémoire ouvre la porte à un grand nombre d'améliorations potentielles. En effet, l'estimé pourra toujours être défini comme la solution du problème d'optimisation (3.16), peu importe la matrice D. On peut alors se poser la question : Quelle serait la *meilleure* matrice D pour un signal donné ? La réponse à cette question pourrait diriger l'analyse statistique préliminaire afin de tenter d'obtenir une matrice D optimale.

On peut penser à d'autres types de tests locaux lors de l'analyse statistique préliminaire. Par exemple, on pourrait tester localement différents modèles sur un signal par l'entremise d'un test du χ^2. On choisirait alors le modèle s'adaptant le mieux localement au signal et cette information pourrait être traduite de la même manière dans la matrice D. On pourrait grâce à cette méthode augmenter le nombre de points sur lesquels on teste les modèles.

La matrice D est construite à partir d'une multitude de tests statistiques locaux avec un seuil de rejet α fixé à 5 %. On peut se demander, suite à ces tests, quelle est l'erreur globale commise dans la construction de la matrice D. Autrement dit, quel niveau de confiance peut-on avoir dans la matrice D construite suite à un très grand nombre de tests à 5 % ? Si ce niveau de confiance est faible, il serait intéressant de changer la façon d'effectuer les tests locaux afin de garantir un niveau de confiance globale bas sur l'ensemble des tests statistiques locaux, c'est-à-dire sur l'ensemble de la matrice D. L'idée de faire des tests statistiques sur un plus grand nombre de points, mentionnée plus haut, diminuerait le nombre de tests effectués et permettrait donc éventuellement de diminuer l'erreur sur l'ensemble des tests.

Le fait que la version oracle de l'algorithme donne de bons résultats rend l'amélioration de l'analyse statistique préliminaire une voie d'amélioration prometteuse.

On a fait comme hypothèse dans ce mémoire que la variance du bruit σ^2 était connue. En réalité, étant donné un signal bruité y, cette variance est inconnue. On pourrait ajouter à notre algorithme une méthode afin d'estimer la variance du bruit directement sur le signal y.

Dans le cas de signaux naturels qui ne sont pas des signaux lisses par morceaux dans leur version non bruitée, comme les images naturelles présentées dans la section 6.2, on peut également penser à des améliorations qui permettraient à notre algorithme de donner de meilleurs résultats. Un pré-traitement du signal pourrait par exemple être fait afin de diminuer le bruit intrinsèque au signal. On lancerait alors notre algorithme sur cette nouvelle version du signal.

L'amélioration du code utilisé dans ce mémoire diminuerait le temps de calcul. Une attention moins grande a été portée à cette composante dans le présent mémoire. L'implémentation de la méthode dans un langage de bas niveau comme C ou C^{++} rendrait l'algorithme plus

rapide. De plus, il existe des méthodes numériques efficaces pour résoudre les problèmes d'optimisation comme celui présenté dans ce mémoire, à savoir la minimisation d'une forme quadratique dans une sphère à haute dimension. Hager (2001) présente une méthode appelée *sequential subspace method* (SSM) pour résoudre ce type de problèmes. L'implémentation d'une telle méthode dans notre algorithme rendrait également celui-ci plus rapide.

En somme, le présent mémoire a jeté les bases théoriques d'un nouvel algorithme de débruitage de signaux numériques. À la lumière des bons résultats obtenus pour des signaux simples ainsi que pour la version oracle de cet algorithme, il est envisageable grâce à des améliorations futures de rendre celui-ci plus performant.

RÉFÉRENCES

M. S. Bazaraa, H. D. Sherali, et C. M. Shetty, *Nonlinear programming : theory and algorithms*, 3e éd. Hoboken, NJ : J. Wiley and Sons, 2006, ch. 4, pp. 190–191.

S. Beheshti et M. A. Dahleh, "A new information-theoretic approach to signal denoising and best basis selection", *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3613–3624, Oct. 2005. DOI : 10.1109/TSP.2005.855075

K. Bredies, K. Kunisch, et T. Pock, "Total generalized variation", *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010. DOI : 10.1137/090769521

A. Buades, B. Coll, et J. Morel, "A review of image denoising algorithms, with a new one", *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

H. C. Burger, C. J. Schuler, et S. Harmeling, "Image denoising : Can plain neural networks compete with bm3d ?" dans *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, États-Unis, Juin 2012, pp. 2392–2399. DOI : 10.1109/CVPR.2012.6247952

A. Chambolle, "An algorithm for total variation minimization and applications", *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, Jan. 2004. DOI : 10.1023/B:JMIV.0000011325.36760.1e

A. Chambolle et P.-L. Lions, "Image recovery via total variation minimization and related problems", *Numerische Mathematik*, vol. 76, no. 2, pp. 167–188, Avr. 1997. DOI : 10.1007/s002110050258

P. Chatterjee et P. Milanfar, "Is denoising dead ?" *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 895–911, Avr. 2010. DOI : 10.1109/TIP.2009.2037087

H. A. Chipman, E. D. Kolaczyk, et R. E. McCulloch, "Adaptive bayesian wavelet shrinkage", *Journal of the American Statistical Association*, vol. 92, no. 440, pp. 1413–1421, Déc. 1997.

R. Coifman et D. Donoho, *Wavelets and Statistics*. New York : Springer Verlag, 1995, ch. Translation-invariant de-noising, pp. 125–150.

K. Dabov, A. Foi, V. Katkovnik, et K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering", *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp.

2080–2095, Août 2007. DOI : 10.1109/TIP.2007.901238

D. L. Donoho et I. Johnstone, “Ideal spatial adaptation by wavelet shrinkage”, *Biometrika*, vol. 81, no. 3, pp. 425–455, Juil. 1994.

——, “Adapting to unknown smoothness via wavelet shrinkage”, *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

M. Elad et M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries”, *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, Déc. 2006. DOI : 10.1109/TIP.2006.881969

A. Foi, V. Katkovnik, et K. Egiazarian, “Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images”, *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, Mai 2007. DOI : 10.1109/TIP.2007.891788

S. H. Friedberg, A. J. Insel, et L. E. Spence, *Linear algebra*. Englewood Cliffs, N.J. : Prentice Hall, 1979, ch. 7, p. 432.

R. C. Gonzalez et R. E. Woods, *Digital image processing*, 2e éd. Upper Saddle River, N.J. : Prentice Hall, 2002.

W. W. Hager, “Minimizing a quadratic over a sphere”, *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 188–21, 2001.

P. Jain et V. Tyagi, “A survey of edge-preserving image denoising methods”, *Information Systems Frontiers*, vol. 18, no. 1, pp. 159–170, Fév. 2016. DOI : 10.1007/s10796-014-9527-0

H. Jarlett. (2017, Déc.) Un stockage de données record. https://home.cern/fr/about/updates/2017/12/breaking-data-records-bit-bit. Consulté le 14 avril 2018.

C. Kervrann et J. Boulanger, “Optimal spatial adaptation for patch-based image denoising”, *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006. DOI : 10.1109/TIP.2006.877529

H.-Y. Lin, S.-Y. Liang, Y.-L. Ho, Y.-H. Lin, et H.-P. Ma, “Discrete-wavelet-transform-based noise removal and feature extraction for ecg signals”, *IRBM*, vol. 35, no. 6, pp. 351–361, 2014.

M. Little et N. Jones, “Generalized methods and solvers for noise removal from piecewise constant signals : Part I-background theory”, *Proc. Roy. Soc. A*, vol. 467, pp. 3088–3114, 2011.

C. Louchet et L. Moisan, “Total variation as a local filter”, *SIAM Journal on Imaging Sciences*, vol. 4, no. 2, pp. 651–694, 2011. DOI : 10.1137/100785855

J. Mairal, F. Bach, J. Ponce, G. Sapiro, et A. Zisserman, “Non-local sparse models for image restoration”, dans *2009 IEEE 12th International Conference on Computer Vision*, Kyoto, Japon, Sep. 2009, pp. 2272–2279. DOI : 10.1109/ICCV.2009.5459452

S. G. Mallat, *A wavelet tour of signal processing*, 2e éd. San Diego : Academic Press, 1999.

G. Nason, “Wavelet shrinkage using cross-validation”, *Journal of the Royal Statistical Society Series b- Methodological*, vol. 58, no. 2, pp. 463–479, 1996.

S. Ramani, T. Blu, et M. Unser, “Monte-carlo sure : A black-box optimization of regularization parameters for general denoising algorithms”, *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008. DOI : 10.1109/TIP.2008.2001404

L. Rudin, S. Osher, et E. Fatemi, “Nonlinear total variation based noise removal algorithms”, *Physica D*, vol. 60, pp. 259–268, 1992.

I. W. Selesnick, A. Parekh, et . Bayram, “Convex 1-d total variation denoising with non-convex regularization”, *IEEE Signal Processing Letters*, vol. 22, no. 2, pp. 141–144, Fév. 2015. DOI : 10.1109/LSP.2014.2349356

D. Strong et T. Chan, “Edge-preserving and scale-dependent properties of total variation regularization”, *Inverse Problems*, vol. 19, no. 6, pp. S165–S187, 2003.

C. Sutour, C. A. Deledalle, et J. F. Aujol, “Adaptive regularization of the nl-means : Application to image and video denoising”, *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3506–3521, Août 2014. DOI : 10.1109/TIP.2014.2329448

H. Takeda, S. Farsiu, et P. Milanfar, “Kernel regression for image processing and reconstruction”, *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, Fév. 2007. DOI : 10.1109/TIP.2006.888330

A. M. Thompson, J. C. Brown, J. W. Kay, et D. M. Titterington, “A study of methods of choosing the smoothing parameter in image restoration by regularization”, *IEEE Tran-*

sactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 4, pp. 326–339, Avr. 1991.

A. N. Tikhonov et V. Y. Arsenin, *Solutions of ill-posed problems*. Washington, D.C. : John Wiley & Sons, New York : V. H. Winston & Sons, 1977.

C. Tomasi et R. Manduchi, "Bilateral filtering for gray and color images", dans *Sixth International Conference on Computer Vision*, Bombay, Inde, Jan. 1998, pp. 839–846. DOI : 10.1109/ICCV.1998.710815

C. R. Vogel et M. E. Oman, "Iterative methods for total variation denoising", *SIAM Journal on Scientific Computing*, vol. 17, no. 1, pp. 227–12, Jan. 1996.

Y. Whang et H. Zhou, "Total variation wavelet-based medical image denoising", *International Journal of Biomedical Imaging*, vol. 2006, pp. 1–6, 2006.

Y. Zhao, J. G. Liu, B. Zhang, W. Hong, et Y. R. Wu, "Adaptive total variation regularization based sar image despeckling and despeckling evaluation index", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2765–2774, Mai 2015. DOI : 10.1109/TGRS.2014.2364525

ANNEXE A CODES

On peut trouver le code Mathematica utilisé pour l'algorithme proposé ainsi que pour les méthodes par ondelettes dans le fichier nommé CodeProjetMaitriseMorin.nb à partir du lien suivant :

`https://github.com/MaitriseMorin/Code-Debruitage-signaux-numeriques`

Le code Matlab pour les méthodes de variation totale peut également être trouvé à partir de ce lien. On présente ce code ici.

Fonction MakeSignal modifiée. Seuls les signaux 1-D synthétiques utilisés dans ce mémoire peuvent être créés à partir de la fonction ci-dessous.

```
function sig = MakeSignal(Name,n)
% MakeSignal -- Make artificial signal
% Usage
% sig = MakeSignal(Name,n)
% Inputs
% Name string: 'Bumps', 'Blocks',
% 'Ramp', 'Piece-Regular' (Piece-Wise Smooth),
% 'Piece-Polynomial' (Piece-Wise 3rd degree polynomial)
% n desired signal length
% Outputs
% sig 1-d signal
%
% References
% Various articles of D.L. Donoho and I.M. Johnstone
%
% Modifiee par Gregoire Morin 2018
%
if nargin > 1,
t = (1:n) ./n;
end
if strcmp(Name,'Bumps'),
pos = [ .1 .13 .15 .23 .25 .40 .44 .65 .76 .78 .81];
hgt = [ 4 5 3 4 5 4.2 2.1 4.3 3.1 5.1 4.2];
wth = [.005 .005 .006 .01 .01 .03 .01 .01 .005 .008 .005];
sig = zeros(size(t));
for j =1:length(pos)
sig = sig + hgt(j)./( 1 + abs((t - pos(j))./wth(j))).^4;
end
elseif strcmp(Name,'Blocks'),
pos = [ .1 .13 .15 .23 .25 .40 .44 .65 .76 .78 .81];
hgt = [4 (-5) 3 (-4) 5 (-4.2) 2.1 4.3 (-3.1) 2.1 (-4.2)];
sig = zeros(size(t));
for j=1:length(pos)
sig = sig + (1 + sign(t-pos(j))).*(hgt(j)/2) ;
end
elseif strcmp(Name,'Ramp'),
```

```
sig = t - (t >= .37);
elseif strcmp(Name,'Piece-Regular'),
sig1=-15*MakeSignal('Bumps',n);
t = (1:fix(n/12)) ./fix(n/12);
sig2=-exp(4*t);
t = (1:fix(n/7)) ./fix(n/7);
sig5=exp(4*t)-exp(4);
t = (1:fix(n/3)) ./fix(n/3);
sigma=6/40;
sig6=-70*exp(-((t-1/2).*(t-1/2))/(2*sigma^2));
sig(1:fix(n/7))= sig6(1:fix(n/7));
sig((fix(n/7)+1):fix(n/5))=0.5*sig6((fix(n/7)+1):fix(n/5));
sig((fix(n/5)+1):fix(n/3))=sig6((fix(n/5)+1):fix(n/3));
sig((fix(n/3)+1):fix(n/2))=sig1((fix(n/3)+1):fix(n/2));
sig((fix(n/2)+1):(fix(n/2)+fix(n/12)))=sig2;
sig((fix(n/2)+2*fix(n/12)):-1:(fix(n/2)+fix(n/12)+1))=sig2;
sig(fix(n/2)+2*fix(n/12)+fix(n/20)+1:(fix(n/2)+2*fix(n/12)+3*fix(n/20)))=...
-ones(1,fix(n/2)+2*fix(n/12)+3*fix(n/20)-fix(n/2)-2*fix(n/12)-fix(n/20))*25;
k=fix(n/2)+2*fix(n/12)+3*fix(n/20);
sig((k+1):(k+fix(n/7)))=sig5;
diff=n-5*fix(n/5);
sig(5*fix(n/5)+1:n)=sig(diff:-1:1);
% zero-mean
bias=sum(sig)/n;
sig=bias-sig;
elseif strcmp(Name,'Piece-Polynomial'),
t = (1:fix(n/5)) ./fix(n/5);
sig1=20*(t.^3+t.^2+4);
sig3=40*(2.*t.^3+t) + 100;
sig2=10.*t.^3 + 45;
sig4=16*t.^2+8.*t+16;
sig5=20*(t+4);
sig6(1:fix(n/10))=ones(1,fix(n/10));
sig6=sig6*20;
sig(1:fix(n/5))=sig1;
sig(2*fix(n/5):-1:(fix(n/5)+1))=sig2;
sig((2*fix(n/5)+1):3*fix(n/5))=sig3;
sig((3*fix(n/5)+1):4*fix(n/5))=sig4;
sig((4*fix(n/5)+1):5*fix(n/5))=sig5(fix(n/5):-1:1);
diff=n-5*fix(n/5);
sig(5*fix(n/5)+1:n)=sig(diff:-1:1);
sig((fix(n/20)+1):(fix(n/20)+fix(n/10)))=ones(1,fix(n/10))*10;
sig((n-fix(n/10)+1):(n+fix(n/20)-fix(n/10)))=ones(1,fix(n/20))*150;
% zero-mean
bias=sum(sig)/n;
sig=sig-bias;
else
disp(sprintf('MakeSignal:␣I␣don*t␣recognize␣<<%s>>',Name))
disp('Allowable␣Names␣are:')
disp('Bumps'),
disp('Blocks'),
disp('Ramp'),
disp('Piece-Regular');
disp('Piece-Polynomial');
end
```

```
% Originally made by David L. Donoho.
% Function has been enhanced.

% Part of Wavelab Version 850
% Built Tue Jan 3 13:20:39 EST 2006
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@stat.stanford.edu
```

Fonction fournie par Selesnick et al. (2015) pour le débruitage de signaux 1-D constant par morceaux avec des méthodes de variation totale.

```
function [x, cost] = TVD_ncvx(y, lam, pen, Nit)
% [x, cost] = TVD_ncvx(y, lam, pen, Nit)
% Total variation denoising with non-convex penalty.
%
% INPUT
% y - noisy signal
% lam - regularization parameter (lam > 0)
% pen - penalty ('log', 'atan', 'L1')
% Nit - number of iterations
%
% OUTPUT
% x - denoised signal
% cost - cost function history

% Reference:
% Convex 1-D Total Variation Denoising with Non-convex Regularization
% Ivan W. Selesnick, Ankit Parekh, and Ilker Bayram
% IEEE Signal Processing Letters, 2014

y = y(:); % Ensure column vector
cost = zeros(1, Nit); % Cost function history
N = length(y);

a = 1 / (4 * lam);

switch pen
case 'L1'

phi = @(x) abs(x);
dphi = @(x) sign(x);
wfun = @(x) abs(x);

case 'log'

phi = @(x) 1/a * log(1 + a*abs(x));
dphi = @(x) 1 ./(1 + a*abs(x)) .* sign(x);
wfun = @(x) abs(x) .* (1 + a*abs(x));

case 'atan'

phi = @(x) 2./(a*sqrt(3)) .* (atan((2*a.*abs(x)+1)/sqrt(3)) - pi/6);
wfun = @(x) abs(x) .* (1 + a.*abs(x) + a.^2.*abs(x).^2);
```

```
dphi = @(x) 1 ./(1 + a*abs(x) + a.^2.*abs(x).^2) .* sign(x);

end

e = ones(N-1, 1);
DDT = spdiags([-e 2*e -e], [-1 0 1], N-1, N-1); % D*D' (banded matrix)
D = @(x) diff(x); % D (operator)
DT = @(x) [-x(1); -diff(x); x(end)]; % D'

x = y; % Initialization
Dx = D(x);
Dy = D(y);

for k = 1:Nit
F = (1/lam) * spdiags(wfun(Dx), 0, N-1, N-1) + DDT; % F : Sparse matrix
    structure

x = y - DT(F\Dy); % Solve banded linear system
Dx = D(x);

cost(k) = 0.5*sum(abs(x-y).^2) + lam*sum(phi(Dx)); % Save cost function
    history
end
```

Exemple d'utilisation de la fonction TVD_ncvx.

```
%% TV denoising using convex and non-convex regularization
%
% Reference:
% Convex 1-D Total Variation Denoising with Non-convex Regularization
% Ivan W. Selesnick, Ankit Parekh, and Ilker Bayram
% IEEE Signal Processing Letters, 2014
%
% Code modifie par Gregoire Morin, 2018

%% Start

clear

%% Create data

N = 1024;
sigma = 0.3;
s = MakeSignal('Blocks', N)';
noise = sigma*randn(N,1);
y = s + noise;

%% Parameters

lam = 0.25 * sqrt(N) * sigma;
Nit = 100;

%% Variation totale originale (L1)

[x_L1, cost] = TVD_ncvx(y, lam, 'L1', Nit);
```

```
MSE_L1 = mean(abs(x_L1 - s).^2);


%% Variation totale avec la fonction arctan

[x_atan, cost_atan] = TVD_ncvx(y, lam, 'atan', Nit);
MSE_atan = mean(abs(x_atan - s).^2);

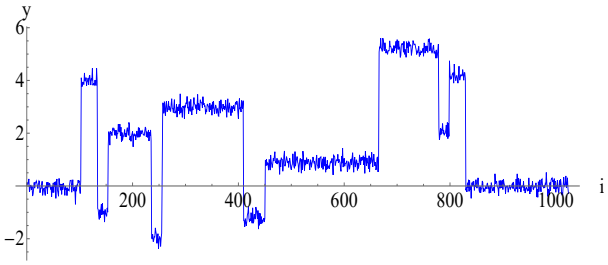
%% Graphiques

figure
plot(y,'b')
hold on
plot(s,'k')
legend('y','s')
title('Signal␣pur␣et␣bruite')

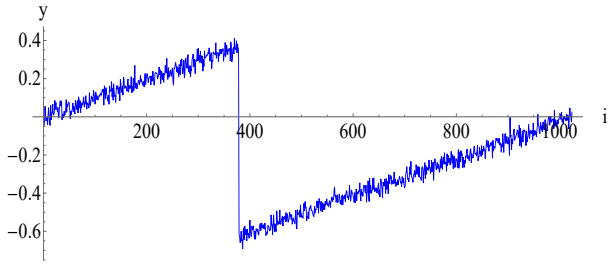
figure
plot(x_L1)
title('Estime␣VT␣L1')

figure
plot(x_atan)
title('Estime␣VT␣Arctan')
```

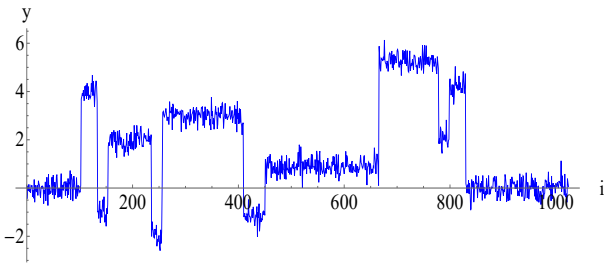
ANNEXE B SIGNAUX 1-D



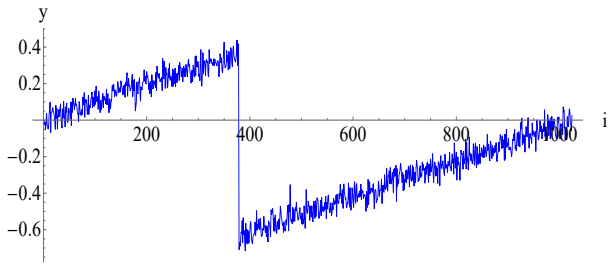
(a) Signal "Blocks", $\sigma = 0.192$, SNR = 20 dB.



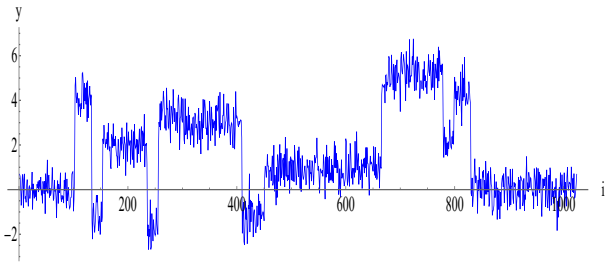
(b) Signal "Ramp", $\sigma = 0.029$, SNR = 20 dB.



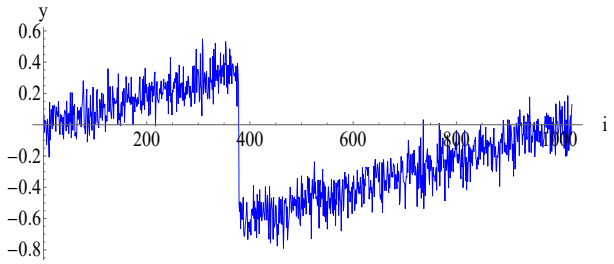
(c) Signal "Blocks", $\sigma = 0.303$, SNR = 16 dB.



(d) Signal "Ramp", $\sigma = 0.046$, SNR = 16 dB.

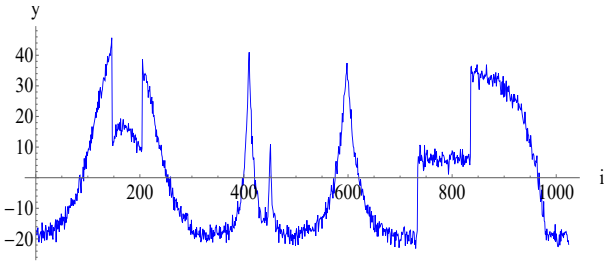


(e) Signal "Blocks", $\sigma = 0.606$, SNR = 10 dB.

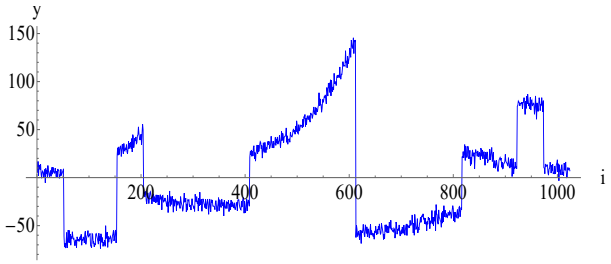


(f) Signal "Ramp",$\sigma = 0.091$, SNR = 10 dB.

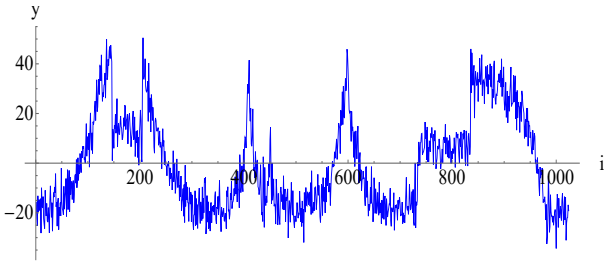
Figure B.1 Exemples des signaux "Blocks" et "Ramp" bruités.



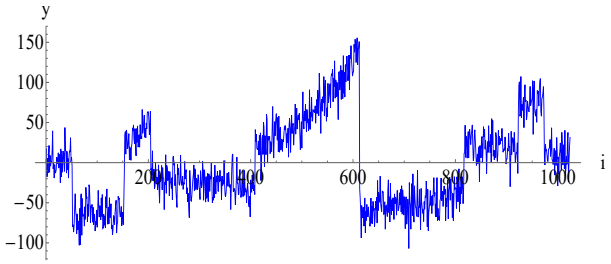
(a) Signal "Piece-Regular", $\sigma = 1.79$, SNR = 20 dB.



(b) Signal "Piece-Polynomial", $\sigma = 5.04$, SNR = 20 dB.



(c) Signal "Piece-Regular", $\sigma = 5.67$, SNR = 10 dB.



(d) Signal "Piece-Polynomial", $\sigma = 15.93$, SNR = 10 dB.

Figure B.2 Exemples des signaux "Piece-Regular" et "Piece-Polynomial" bruités.