



**HAL**  
open science

# Régularité et parcimonie pour les problèmes inverses en imagerie : algorithmes et comparaisons

Mikael Carlavan, Pierre Weiss, Laure Blanc-Féraud

## ► To cite this version:

Mikael Carlavan, Pierre Weiss, Laure Blanc-Féraud. Régularité et parcimonie pour les problèmes inverses en imagerie : algorithmes et comparaisons. *Traitement du Signal*, 2010. inria-00503050

**HAL Id: inria-00503050**

**<https://inria.hal.science/inria-00503050>**

Submitted on 16 Jul 2010

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

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

---

# Régularité et parcimonie pour les problèmes inverses en imagerie : algorithmes et comparaisons

Mikael Carlván\* — Pierre Weiss\*\* — Laure Blanc-Féraud\*

\* *Projet ARIANA INRIA/I3S/CNRS, 2004 route des Lucioles, 06902 Sophia Antipolis  
{mcarlava,blancf}@sophia.inria.fr*

\*\* *Institut de Mathématiques de Toulouse, Université Paul Sabatier, 118 route de  
Narbonne, 31062 Toulouse  
pierre.armand.weiss@gmail.com*

---

*RÉSUMÉ. Dans cet article, nous nous intéressons à la régularisation de problèmes inverses reposant sur des critères  $l^1$ . Nous séparons ces critères en deux catégories : ceux qui favorisent la régularisation des signaux (à variation totale bornée par exemple) et ceux qui expriment le fait qu'un signal admet une représentation parcimonieuse dans un dictionnaire. Dans une première partie, nous donnons quelques éléments de comparaisons théoriques et pratiques sur les deux a priori, pour aider le lecteur à choisir l'un ou l'autre en fonction de son problème. Pour cette étude, nous utilisons les transformées communément utilisées telles que la variation totale, les ondelettes redondantes ou les curvelets. Dans une deuxième partie, nous proposons un état des lieux des algorithmes de premier ordre adaptés à la minimisation de ces critères.*

*ABSTRACT. This article is a survey on regularization techniques for inverse problems based on  $l^1$  criteria. We split these criteria in two categories : those which promote regularity of the signal (e.g. total variation) and those which express the fact that a signal is sparse in some dictionary. In the first part of the paper, we give guidelines to choose a prior and propose a comparative study of these two priors on standard transforms such as total variation, redundant wavelets, and curvelets. In the second part of the paper, we give a sketch of different first order algorithms adapted to the minimization of these  $l^1$ -terms.*

*MOTS-CLÉS : Problèmes inverses, imagerie, critères  $l^1$ , régularité, parcimonie, algorithmes rapides, comparaisons expérimentales.*

*KEYWORDS: Inverse problems, imaging,  $l^1$ -criteria, regularity, sparsity, fast algorithms, experimental comparisons.*

---

## 1. Introduction

Ces dernières années ont vu un réel engouement de la communauté de traitement du signal et des images pour les techniques de régularisation reposant sur des critères  $l^1$ . Des termes comme parcimonie, échantillonnage compressif ou variation totale font régulièrement la une de conférences ou de revues spécialisées. De nombreux modèles et algorithmes d'optimisation sont proposés puis appliqués à la masse croissante des problèmes inverses rencontrés en ingénierie du signal.

Notre objectif dans cet article est de proposer un état des lieux de deux aspects de ces techniques :

1) l'aspect modélisation. On s'intéresse aux différences existant entre les modèles qu'on appellera de "régularité", dont la construction repose sur le fait que les signaux à reconstruire sont supposés réguliers (à variation bornée par exemple), et les modèles dits de "parcimonie" qui consistent à chercher des signaux qui s'expriment comme une combinaison linéaire d'un petit nombre d'atomes dans un dictionnaire adapté. Ces modèles sont largement utilisés pour la résolution de problèmes inverses.

2) l'aspect algorithmique. On s'intéresse à l'algorithmie nécessaire à la résolution rapide des problèmes d'optimisation convexe résultants. Nous ne présentons que des algorithmes de premier ordre, en dimension finie, tels que les descentes de (sous-) gradient, les techniques de forward-backward splitting et leur accélération multi-pas, les méthodes de type Douglas-Rachford splitting ou de directions alternées. Ces techniques sont aujourd'hui de plus en plus utilisées pour des problèmes inverses de traitement du signal et des images.

Le reste de ce document est organisé comme suit.

Dans une première partie, nous rappelons brièvement les principes qui motivent l'introduction de normes  $l^1$  pour la régularisation des problèmes inverses en traitement du signal et particulièrement de l'image. Nous proposons aussi une brève description des différences théoriques existant entre *a priori* de régularité et de parcimonie. Finalement, nous proposons une étude comparative expérimentale des deux *a priori* sur un problème de déconvolution. Ceci nous permet de comparer les performances de transformées utilisées communément pour la description des images, telles que les transformées en ondelettes redondantes, les curvelets ou la variation totale.

Dans une deuxième partie, nous décrivons de façon succincte plusieurs algorithmes d'optimisation adaptés à la minimisation des fonctionnelles convexes résultantes. Nous nous concentrons sur les techniques de premier ordre qui présentent l'avantage d'être rapides et de s'adapter à de nombreux scénarii. Notons qu'après soumission de cet article, nous avons découvert qu'un travail similaire a été réalisé récemment (Combettes *et al.*, 2010) avec un point de vue légèrement différent (espace hilbertien, pas de comparaison parcimonie/régularité). Le lecteur intéressé pourra trouver des informations et références complémentaires dans cet article.

## 2. Régularité et parcimonie pour les problèmes inverses

En imagerie numérique, de nombreux systèmes d'acquisition fournissent un ensemble de données  $z \in \mathbb{R}^m$  qui n'est pas directement interprétable comme une image ou une séquence d'images.  $z$  est en fait un ensemble de mesures indirectes et bruitées de l'objet d'intérêt. Dans la majorité des imageurs, le modèle mathématique de formation de ces données est le suivant :

$$z = \Phi u + b \quad [1]$$

où  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$  est un opérateur linéaire (échantillonnage, convolution, transformée de Radon,...),  $u \in \mathbb{R}^n$  est l'image que l'on souhaite reconstruire et  $b \in \mathbb{R}^m$  est un bruit additif.

Retrouver  $u$  à partir des mesures  $z$  est communément appelé problème inverse (Kirsch, 1996). Les premières techniques utilisées pour le résoudre consistaient à définir  $u$  comme  $\Phi^\dagger z$  où  $\Phi^\dagger$  est la pseudo-inverse de  $\Phi$ . Celle-ci peut être définie comme :

$$\Phi^\dagger z = \lim_{\tau \rightarrow 0} \arg \min_{u \in \mathbb{R}^n} \|\Phi u - z\|_2^2 + \tau \|u\|_2^2. \quad [2]$$

où  $\|\cdot\|_p$  est la norme  $l^p$  standard. L'inconvénient de cette approche est que l'opérateur  $\Phi$  a généralement un spectre compact, ce qui rend le problème inverse (1) mal-posé : l'opérateur  $\Phi^\dagger$  possède des valeurs singulières très élevées et  $\|\Phi^\dagger b\|$  peut être grand même lorsque  $\|b\|$  est faible.

En présence de bruit, de tels problèmes doivent être régularisés. Pour ce faire, les premières techniques utilisées consistaient à fixer  $\tau > 0$  et à résoudre le problème (2). Tikhonov a analysé ce type de techniques en incluant un opérateur linéaire  $G$  (typiquement un opérateur différentiel) dans le terme de régularisation :

$$\arg \min_{u \in \mathbb{R}^n} \|\Phi u - z\|_2^2 + \tau \|Gu\|_2^2. \quad [3]$$

Des techniques encore plus générales ont été développées. Elles consistent à résoudre des problèmes de type :

$$\arg \min_{u \in \mathbb{R}^n} \Upsilon(\Phi u - z) + \tau J(u) \quad [4]$$

où  $\Upsilon : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$  est une fonction dépendant de la statistique du bruit additif,  $J$  est un *a priori* sur l'image à reconstruire et  $\tau > 0$  est un paramètre de régularisation. Plus  $\tau$  est petit, plus l'image est proche des données, plus  $\tau$  est grand, plus la solution obtenue est régularisée. Le problème de minimisation du critère (4) peut être interprété dans l'approche bayésienne comme l'estimateur du maximum *a posteriori* (MAP) où le premier terme correspond à l'antilogarithme de la vraisemblance et le deuxième terme correspond au modèle sur l'objet cherché, c'est-à-dire le modèle *a priori* (Idier, 2001). Dans cet article nous nous intéressons au modèle de reconstruction (4) pour différents types d'attaches aux données  $\Upsilon$ , et à deux types d'*a priori* de régularisation  $J(\cdot)$  : les *a priori* de régularité et les *a priori* de parcimonie.

### 2.1. Choix de la fonction d'attache aux données $\Upsilon$

Suivant le type de bruit  $b$  en présence, le principe du MAP mène à différents choix pour la fonction  $\Upsilon$ . Les plus courants sont les suivants :

- Bruit blanc Gaussien :  $\Upsilon(\cdot) = \|\cdot\|_2^2$  (Geman *et al.*, 1984).
- Bruit blanc laplacien :  $\Upsilon(\cdot) = \|\cdot\|_1$  (Hamza *et al.*, 2001).
- Approximation du bruit impulsionnel :  $\Upsilon(\cdot) = \|\cdot\|_1$  (Nikolova, 2004).
- Bruit uniforme ou quantification :  $\Upsilon(\cdot) = \|\cdot\|_\infty$  (Gibson *et al.*, 2000; Weiss *et al.*, 2009).
- Bruit de Poisson :  $\Upsilon(x) = \sum_{i=1}^n x_i \log(x_i)$  (Lucy, 1974; Dupé *et al.*, 2009).

Souvent, l'estimation des paramètres est plus simple si  $\Upsilon$  est la fonction caractéristique d'un ensemble. Par exemple, si on est en présence d'un bruit blanc additif gaussien dont on connaît la variance  $\sigma^2$ , on a intérêt à résoudre le problème contraint suivant :

$$\arg \min_{u \in \mathbb{R}^n, \|\Phi u - z\|_2^2 \leq \sigma^2 n} J(u), \quad [5]$$

ainsi il n'est plus nécessaire d'estimer le paramètre  $\tau$ . Cet exemple revient à choisir :

$$\Upsilon(x) = \begin{cases} 0 & \text{si } \|x\|_2^2 \leq \sigma^2 n \\ +\infty & \text{sinon} \end{cases}. \quad [6]$$

Notons que dans tous les exemples mentionnés ci-dessus, la fonction  $\Upsilon$  est une fonction convexe fermée i.e. convexe semi-continue inférieurement (définitions au chapitre 3.1). On verra dans cet article que la convexité est une condition presque suffisante pour construire des algorithmes qui fournissent un minimum global en des temps raisonnables.

### 2.2. Choix de la fonction de régularisation $J(\cdot)$

La fonction  $x \mapsto \exp(-J(x))$  peut être interprétée comme une densité de probabilité sur l'espace des images. De très nombreux travaux consistent à rechercher des probabilités réalistes (Gaetan *et al.*, 2008). En traitement d'image, les *a priori* qui semblent connaître le plus grand succès actuellement sont certainement ceux reposant sur des critères  $l^1$  (Rudin *et al.*, 1992; Chambolle *et al.*, 1996; Donoho, 1995; Dupé *et al.*, 2009). Ces critères permettent en effet d'obtenir des résultats probants en des temps de calcul raisonnables et l'analyse des propriétés des minimiseurs est simplifiée par la convexité des différentes fonctions.

Ces critères  $l^1$  peuvent être séparés en deux grandes catégories, selon que le critère est écrit dans le domaine image ou dans le domaine transformé : les *a priori* de "régularité" et ceux de "parcimonie".

### 2.2.1. *A priori de régularité*

L'idée sous-jacente aux *a priori* de régularité est qu'une image est une fonction "lisse" ou "régulière" par morceaux. Par exemple, l'espace des images peut être modélisé par l'ensemble des fonctions  $C^\alpha$  par morceaux à bords  $C^\alpha$  (Le Pennec *et al.*, 2005). Il faut donc trouver des fonctions  $J$  qui favorisent ce type de régularité. Les meilleurs candidats existant dans la littérature sont des semi-normes de type variation totale ou Besov. Ils s'écrivent sous la forme :

$$J(u) = \|Du\|_1$$

où  $D : \mathbb{R}^n \rightarrow \mathbb{R}^p$  ( $p \geq n$ ) est une transformée linéaire. Ce type d'*a priori* suscite une littérature abondante dans le cas  $D = \nabla$ , le gradient discret,  $J$  correspondant alors à la variation totale (Rudin *et al.*, 1992). Un *a priori* beaucoup moins utilisé pour la résolution de problèmes inverses, mais qui fournit - nous le verrons plus tard - des résultats expérimentaux de bonne qualité pour des images naturelles, consiste à choisir une transformée en ondelettes redondantes. Dans ce cas,  $\|Du\|_1$  correspond à une semi-norme de Besov (Chambolle *et al.*, 1996) discrétisée. En utilisant un *a priori* de régularité, le problème (4) devient alors :

$$\text{Trouver } u \in \arg \min_{u \in \mathbb{R}^n} \Upsilon(\Phi u - z) + \tau \|Du\|_1. \quad [7]$$

Par la suite,  $U$  désignera l'ensemble des solutions de (7). L'utilisation de transformées en ondelettes redondantes avec un *a priori* de régularité est étonnamment peu répandue dans la littérature en traitement d'image (Bect *et al.*, 2004; Figueiredo *et al.*, 2003; Daubechies *et al.*, 2004). Celles-ci sont beaucoup plus souvent utilisées avec un *a priori* de parcimonie, décrit dans le paragraphe suivant.

### 2.2.2. *A priori de parcimonie*

L'idée sous-jacente aux *a priori* de parcimonie est qu'une image peut s'exprimer sous la forme d'une combinaison linéaire d'un petit nombre d'atomes élémentaires  $\{\psi_i\}_{i=1}^p$ ,  $\psi_i \in \mathbb{R}^n$ . Mathématiquement, ceci s'écrit :

$$u = \sum_{i=1}^p v_i \psi_i \quad [8]$$

avec  $p \geq n$  mais  $\#\{v_i, v_i \neq 0\} \ll n$  ( $\#$  est le cardinal d'un ensemble). L'équation (8) peut être réécrite sous la forme condensée  $u = Rv$  où  $R = [\psi_1, \psi_2, \dots, \psi_p]$

et  $v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_p \end{pmatrix}$ .  $R$  peut être interprété comme un opérateur de reconstruction ou de

synthèse tandis que dans le paragraphe précédent  $D$  désignait un opérateur de décomposition ou d'analyse. Pour pouvoir représenter tous les éléments de  $\mathbb{R}^n$ , nous supposons par la suite que le rang de  $R$  est égal à  $n$ .

Une méthode naturelle pour favoriser la parcimonie consiste à résoudre le problème suivant :

$$\arg \min_{v \in \mathbb{R}^p} \Upsilon(\Phi Rv - z) + \tau \|v\|_0 \quad [9]$$

où  $\|v\|_0$  est le nombre de composantes non nulles de  $v$ . Ce problème est combinatoire (NP-complet) et sa résolution rapide reste un problème ouvert. En pratique, la norme  $l^0$  est remplacée par son enveloppe convexe qui est la norme  $l^1$  (Chen *et al.*, 1998). En utilisant cette relaxation, le problème (4) devient alors :

$$\text{Trouver } v \in \arg \min_{v \in \mathbb{R}^p} \Upsilon(\Phi Rv - z) + \tau \|v\|_1. \quad [10]$$

Par la suite  $V$  désignera l'ensemble des solutions de (10). Notons que lorsque  $\Upsilon(\cdot) = \|\cdot\|_2^2$ , il existe des conditions qui assurent l'égalité des solutions du problème (9) et du problème (10) (Fuchs, 2005).

### 2.2.3. Liens théoriques élémentaires entre les deux a priori

Bien que les deux types d'*a priori* présentés soient largement utilisés dans la littérature, il n'existe quasiment aucun travail présentant de comparaisons entre les problèmes (7) et (10), à l'exception notable de (Elad *et al.*, 2007; Selesnick *et al.*, 2009). L'objectif de ce paragraphe est de présenter quelques liens théoriques simples entre les deux modèles. Nous rappellerons aussi le résultat théorique principal obtenu par les auteurs de (Elad *et al.*, 2007). Ceux-ci se placent dans le cadre  $\Upsilon(\cdot) = \|\cdot\|_2^2$ . Nous présenterons des résultats expérimentaux au paragraphe 2.2.4.

#### Cas 1 : $D$ est une base

Un changement de variable dans (10) montre que lorsque  $D$  est une base, en choisissant  $R = D^{-1}$ , nous obtenons  $V = RU$  où  $V$  est l'ensemble des solutions de (10) et  $U$  est l'ensemble des solutions de (7). Ceci signifie que les deux problèmes sont équivalents, au sens où ils mènent à des solutions identiques pour  $R = D^{-1}$ .

#### Cas 2 : $R = D^\dagger$

De façon plus générale, en faisant le changement de variable  $v = Du$  dans (7), et en remarquant que  $D^\dagger v = u$  (car  $D^\dagger D = I$ ), nous obtenons :

$$U = D^\dagger \arg \min_{v \in \text{Im}(D)} \Upsilon(\Phi D^\dagger v - z) + \tau \|v\|_1 \quad [11]$$

En choisissant  $R = D^\dagger$ , le problème (7) est donc identique au problème (10) à l'exception de la contrainte affine  $v \in \text{Im}(D)$ . On voit ainsi que les solutions du problème (7) sont contraintes à un espace de dimension plus faible que celles de (10).

Ceci montre qu'en utilisant le modèle de régularité (7), on ne cherche pas forcément des solutions parcimonieuses. Par exemple, en choisissant  $D = \begin{bmatrix} \text{Id} \\ F \end{bmatrix}$  où  $F$

est la transformée de Fourier discrète,  $Im(D)$  ne contient aucun éléments parcimonieux d'après le principe de Heisenberg. Ainsi la résolution de (11) n'a pas de raison particulière de mener à des solutions creuses.

**Cas 3 :  $\Upsilon(\cdot) = \|\cdot\|_2^2$  (Elad *et al.*, 2007)**

Finalement, rappelons le théorème principal obtenu dans (Elad *et al.*, 2007) : *pour toute transformée  $D$  de rang plein, il existe une transformée  $R$  telle que pour tout  $x^0$ ,  $U = RV$ . L'inverse n'est pas vrai.* Ce dernier théorème indique que tout modèle de régularité peut être remplacé par un modèle de parcimonie avec une transformée  $R$  bien choisie. Il nous permet de conclure que les modèles de parcimonie sont plus généraux et devraient permettre d'obtenir de meilleurs résultats de reconstruction pour une transformée  $R$  bien adaptée aux images. En pratique, le choix d'une telle transformée est difficile et nous verrons dans la section expérimentale que les transformées utilisées actuellement donnent de meilleurs résultats de reconstruction pour des *a priori* de régularité.

#### 2.2.4. Comparaisons expérimentales des deux modèles

Nous avons effectué des comparaisons expérimentales des deux *a priori* sur un problème de déconvolution d'image avec bruit blanc additif gaussien. Ce scénario correspond au Cas 3 décrit dans le paragraphe précédent. D'après les résultats de (Elad *et al.*, 2007), on sait qu'avec une transformée  $R$  bien choisie, on devrait obtenir de meilleurs résultats avec le modèle de parcimonie qu'avec le modèle de régularité.

Pour cette comparaison, nous avons utilisé les transformées linéaires utilisées fréquemment dans la littérature telles que la variation totale (TV) (Rudin *et al.*, 1992), les curvelets (Candès *et al.*, 2006), les ondelettes redondantes (RDWT) de type Daubechies DB8 (Daubechies, 1992), les ondelettes redondantes complexes "dual-tree" (DTCW) (Selesnick *et al.*, 2005) ainsi que des dictionnaires composés de ces différentes transformées. Notre objectif est de déterminer quel modèle fournit les meilleurs résultats en pratique et quelles transformées doivent être utilisées.

Pour comparer les modèles, nous avons convolué plusieurs images, dont l'image Barbara avec des bruits additifs gaussiens de variances  $\sigma_b^2$  égales à 0.001, 0.0025, 0.005, 0.0075 et 0.01. Nous avons utilisé un noyau de convolution gaussien d'une taille de  $7 \times 7$  pixels. A titre d'exemple, les PSNR de l'image déconvoluée sont donnés sur le tableau 1.

Cet exemple et les expériences que nous avons menées sur des images à contenus différents, nous ont permis d'arriver aux conclusions suivantes :

– **Parcimonie ou régularité ?** Une amélioration sensible peut être obtenue avec un *a priori* de parcimonie si on est capable de trouver la transformée adaptée à l'image. Visuellement, la généralité des *a priori* de parcimonie sur ceux de régularité permet de retrouver plus de détails mais également plus d'éléments non désirés comme des artéfacts (voir la figure 1). Ceci semble expliquer les faibles performances de cet *a priori* lorsque le bruit devient trop important et les artéfacts deviennent majoritaires



par rapport à l'information apportée par l'ajout de détails. Dans certaines applications comme en imagerie biologique par exemple, on préférera restaurer moins de détails mais avoir peu d'artéfacts et donc utiliser un *a priori* de régularité. De façon grossière, on peut conclure que les *a priori* de parcimonie peuvent être préférés à faible niveau de bruit, tandis que les *a priori* de régularité peuvent être choisis à fort niveau de bruit.

– **Quelle transformée utiliser ?** Il est assez difficile de donner une réponse à cette question tant le résultat dépend du contenu de l'image. Néanmoins, d'après les simulations que nous avons réalisé sur une base composée de quelques images naturelles (ne présentant pas particulièrement de motifs favorables à une transformée), nous avons constaté à plusieurs reprises que les ondelettes redondantes donnent de bons résultats lorsque le bruit est faible. Cependant, elles sont rapidement dépassées par des transformées un peu plus élaborées, comme les ondelettes complexes dual-tree ou les curvelets, lorsque le bruit devient important. Dans cette configuration, les ondelettes complexes dual-tree permettent d'obtenir des résultats légèrement supérieurs à ce que donnent les curvelets. Nous avons aussi remarqué qu'il est intéressant de former des dictionnaires composés de ces différentes transformées. Le dictionnaire composé des ondelettes redondantes et des curvelets permet également d'obtenir des résultats sensiblement supérieurs à ceux donnés par les ondelettes complexes dual-tree. Pour illustrer ces propos, des résultats de déconvolution sur l'image test *Barbara* sont donnés tableau 1 et figure 1. Nous préparons en ce moment un rapport dans lequel des comparaisons détaillées sur des modèles plus variés seront effectuées.

Observations	25.90	23.46	21.23	19.71	18.49
TV	27.10	26.33	25.38	24.77	24.53
DTCW (R)	27.50	26.70	25.77	<b>25.25</b>	<b>25.16</b>
Curvelets (R)	27.40	26.58	25.49	25.02	24.87
RDWT (R)	27.69	26.47	25.79	24.78	24.45
RDWT + Curvelets (R)	27.58	26.65	25.63	25.02	24.78
DTCW + Curvelets (R)	27.44	26.65	25.71	25.21	25.12
RDWT + DTCW (R)	27.77	26.70	25.72	25.09	24.86
DTCW (P)	27.73	26.78	<b>25.83</b>	25.24	25.15
Curvelets (P)	27.50	26.55	25.47	24.95	24.78
RDWT (P)	27.60	26.20	25.09	24.33	23.91
RDWT + Curvelets (P)	27.66	26.56	25.43	24.80	24.50
DTCW + Curvelets (P)	27.77	26.81	25.74	25.14	24.96
RDWT + DTCW (P)	<b>27.97</b>	<b>26.84</b>	25.58	24.75	24.33

**Tableau 1.** PSNR en dB des différentes régularisations utilisées sur l'image *Barbara*. (P) désigne un *a priori* de parcimonie tandis que (R) désigne un *a priori* de régularité.



**Figure 1.** Résultats sur l'image Barbara, image observée ( $PSNR = 25.90$  dB). Seuls sont présentés les résultats avec les transformations classiques (TV, DTCW, RDWT et Curvelets) et un résultat obtenu par dictionnaire (Curvelets + RDWT). (P) désigne un a priori de parcimonie tandis que (R) désigne un a priori de régularité. De gauche à droite et de haut en bas : image d'origine, résultat obtenu avec la TV, DTCW (R), DTCW (P), Curvelets (R), Curvelets (P), RDWT (R), RDWT (P), Curvelets + RDWT (R) et Curvelets + RDWT (P).

### 3. Algorithmes de minimisation

Les problèmes d'optimisation convexe (7) et (10) ont plusieurs particularités qui rendent leur résolution difficile :

- Ils sont non-différentiables ou leur gradient n'est pas Lipschitz (propriété abondamment utilisée dans les algorithmes que nous présenterons).
- Ils vivent dans des espaces de très grande dimension (une image numérique est rarement composée de moins de  $10^6$  pixels et peut dépasser  $10^9$  voxels).
- Leur solution n'est pas unique en général.
- Il faut souvent obtenir des solutions en des temps courts, voire en temps réel (vidéo, imagerie médicale,...).

La dimension des problèmes fait qu'il est actuellement impossible d'obtenir des solutions de précision machine en des temps raisonnables. Lorsque l'on résout des problèmes si grands, il faut donc se contenter d'approximations suffisamment précises pour le système visuel humain. De telles précisions peuvent être atteintes en des temps courts par des techniques de premier ordre, n'utilisant que des informations de type gradient ou sous-gradient à chaque itération. Ces techniques présentent l'avantage d'avoir un faible coût par itération et nécessitent relativement peu d'espace mémoire. Dans cet article, nous présenterons donc uniquement ces méthodes et n'évoquerons pas les techniques d'ordre plus élevé, dérivées de la méthode de Newton telles que les points intérieurs (Nesterov *et al.*, 1994) ou le Second Order Cone Programming (Boyd *et al.*, 2004). Même avec cette restriction, il existe encore de très nombreuses méthodes et l'objectif de ce paragraphe n'est pas de les présenter de façon exhaustive, mais simplement d'aider le lecteur à se diriger vers l'une ou l'autre classe suivant le problème considéré et à construire une méthode adaptée à un problème donné. Nous ne donnons que les grandes lignes des algorithmes et un examen plus approfondi de la technique retenue sera probablement utile et le lecteur pourra se référer aux articles cités.

Les opérateurs linéaires  $R$ ,  $D$  et  $\Phi$  représentent souvent une source de difficulté importante pour la résolution des problèmes (7) et (10). Par exemple, nous verrons plus tard que la présence de l'opérateur  $D$  dans (7) ne permet pas en général de calculer de façon explicite la projection associée au terme de régularisation. Pour contourner cette difficulté, le problème peut être transformé en ajoutant des contraintes linéaires et en augmentant la taille de l'espace d'optimisation. Cette remarque nous a amené à classer les différents algorithmes suivant deux classes : ceux pour lesquels on travaille directement dans l'espace où est posé le problème et ceux qui reposent sur un ajout de contraintes linéaires et une augmentation de la dimension du problème.

Le reste de cette partie est organisé comme suit : nous rappelons d'abord quelques résultats d'analyse convexe, nous présentons ensuite des algorithmes qui résolvent directement le problème initial, puis nous présentons quelques techniques de résolution par transformation du problème en ajoutant des contraintes linéaires.

### 3.1. Notations et rappels d'optimisation convexe

Dans ce paragraphe, nous rappelons les résultats d'analyse convexe indispensables à la compréhension des différents algorithmes.

On considèrera systématiquement des problèmes d'optimisation dans des espaces de dimension finie.  $\langle \cdot, \cdot \rangle$  est le produit scalaire usuel.  $\| \cdot \|_2$  est la norme associée.  $\| \cdot \|_p$  est la norme  $l^p$  standard. La norme  $\| \cdot \|_2$  définit la métrique et la topologie de  $\mathbb{R}^n$ . Soit  $X \subseteq \mathbb{R}^n$  un ensemble.  $\text{ri}(X)$  est l'intérieur relatif de  $X$ . L'ensemble vide est noté  $\emptyset$ .

Id représente l'application identité. Soit  $A : \mathbb{R}^n \rightarrow \mathbb{R}^p$  une application linéaire.  $\|A\| = \max_{x \in \mathbb{R}^n, \|x\|_2 \leq 1} \|Ax\|_2$  représente la plus grande valeur singulière de  $A$ .  $A^\dagger$  est la pseudo-inverse de  $A$ . Elle a déjà été définie en (2). Rappelons le résultat suivant :  $(\text{Id} - A^\dagger A)$  est l'opérateur de projection sur  $\ker(A)$  (Golub *et al.*, 1996, p. 257). Soit  $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$  une application. Celle-ci est dite  $L$ -Lipschitz si  $\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \|g(x) - g(y)\|_2 \leq L\|x - y\|_2$ .

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  une fonction convexe. Cette fonction est dite fermée si les ensembles  $\{x \in \mathbb{R}^n, f(x) \leq \alpha\}$  sont fermés pour tout  $\alpha$ . Le sous-différentiel de  $f$  au point  $x$ , noté  $\partial f(x)$  est défini par :

$$\partial f(x) = \{\eta \in \mathbb{R}^n, f(x+h) \geq f(x) + \langle \eta, h \rangle \forall h \in \mathbb{R}^n\}. \quad [12]$$

Si  $f$  est différentiable en  $x$  alors  $\partial f(x) = \{\nabla f(x)\}$ .

Une fonction  $f$  est dite fortement convexe de paramètre  $\nu$  si elle est convexe et si :

$$f(x+h) \geq f(x) + \langle \eta, h \rangle + \frac{\nu}{2} \|h\|_2^2, \forall x \in \mathbb{R}^n, \forall \eta \in \partial f(x). \quad [13]$$

Par exemple la fonction  $x \mapsto \frac{1}{2} \|x\|_2^2$  est fortement convexe de paramètre de forte convexité égal à 1.

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  une fonction convexe, fermée. La transformée de Fenchel de  $f$  est notée  $f^*$ . Elle est définie par :

$$f^*(y) = \max_{x \in \mathbb{R}^n} \langle x, y \rangle - f(x). \quad [14]$$

C'est une fonction convexe fermée. Si  $f$  est une fonction fortement convexe de paramètre  $\nu > 0$ , alors  $f^*$  est une fonction différentiable à gradient  $\frac{1}{\nu}$ -Lipschitz.

L'opérateur proximal associé à  $f$ , noté  $\text{prox}_f$  est défini par :

$$\text{prox}_f (x^0) = \arg \min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2} \|x - x^0\|_2^2. \quad [15]$$

Pour obtenir plus d'informations sur cet opérateur, nous renvoyons le lecteur aux travaux de Combettes (Combettes *et al.*, 2007). Nous rappelons uniquement trois résultats dont nous nous servirons par la suite. Soit  $X \subseteq \mathbb{R}^n$  un ensemble convexe, fermé

et  $f(x) = \begin{cases} 0 & \text{si } x \in X \\ +\infty & \text{sinon} \end{cases}$ . Alors  $\text{prox}_f = \Pi_X$ , le projecteur euclidien sur l'ensemble  $X$ . L'opérateur proximal généralise donc la projection. Si  $f(x) = \tau\|x\|_1$ , alors  $\text{prox}_f$  est l'opérateur connu sous le nom de seuillage doux. Un calcul élémentaire donne :

$$\text{prox}_{\tau\|\cdot\|_1}(x^0) = \text{shrink}_\tau(x^0) = \text{sign}(x^0) \max(|x^0| - \tau, 0) \quad [16]$$

Finalement, si  $R : \mathbb{R}^p \rightarrow \mathbb{R}^n$  est une trame ajustée normalisée, c'est-à-dire une application linéaire telle que  $RR^* = \text{Id}^1$ , alors on peut montrer le résultat suivant (Combettes *et al.*, 2007) :

$$\text{prox}_{f \circ R}(x) = x + R^* \circ (\text{prox}_f - \text{Id}) \circ R(x). \quad [17]$$

### 3.2. Algorithmes d'optimisation sans transformation

Comme nous l'avons précisé, certains algorithmes consistent à transformer le problème en augmentant la dimension du problème et en ajoutant des contraintes linéaires. Ces algorithmes sont présentés en partie 3.3. Dans cette première partie, nous présentons des algorithmes qui ne font pas intervenir de variables supplémentaires au problème de départ. Dans cette classe d'algorithmes, les techniques principalement utilisées sont les suivantes :

- Les descentes de sous-gradient ou les descentes de sous-gradient projeté.
- Les techniques appelées “forward-backward splitting” qui peuvent être vues simplement comme une extension de la descente de gradient projeté.
- Les techniques d'accélération de convergence de Nesterov.
- Les techniques de lissage de fonctions non lisses.
- Les techniques de type “Douglas-Rachford”.

Nous les présentons succinctement dans la suite. Certaines de ces techniques ont également été synthétisées récemment dans (Aujol, 2009).

#### 3.2.1. Sous gradient projeté

A notre connaissance, la première méthode convergente pour résoudre les problèmes d'optimisation convexe non différentiable a été la technique de descente de sous-gradient projeté (Polyak, 1987). Cette technique s'applique au problème général suivant :

$$\text{Trouver } x \in \arg \min_{x \in X \subseteq \mathbb{R}^n} f(x) \quad [18]$$

où  $f : X \rightarrow \mathbb{R}$  est une fonction convexe, fermée, Lipschitz admettant un ensemble de minimiseurs  $X^* \neq \emptyset$ . Ce cadre inclut les problèmes (7) et (10) pour peu que  $\Upsilon$  soit Lipschitz. Il suffit en effet de choisir  $f(x) = \Upsilon(\Phi u - z) + \tau\|Du\|_1$  et  $X = \mathbb{R}^n$ .

1. c'est le cas de la transformée en curvelets par exemple ou d'une union de bases orthonormées

L'algorithme du sous-gradient projeté s'écrit comme suit :

---

**Algorithme 1:** Descente de sous-gradient projeté.

---

**Entrées :** Le nombre d'itérations  $N$  ;  
Le point de départ  $x^0 \in X$  ;

**Sorties :**  $x^N$ , une estimée d'un élément de  $X^*$ .

**début**

**pour**  $k$  allant de 0 à  $N - 1$  **faire**  
    |      $x^{k+1} = \Pi_X \left( x^k - t^k \frac{\eta^k}{\|\eta^k\|_2} \right)$ , avec  $\eta^k \in \partial f(x^k)$ .  
    |     **fin**  
**fin**

---

Le problème de cet algorithme est qu'il faut trouver une suite  $\{t^k\}$  qui assure la convergence - si possible rapide - de l'algorithme. Une technique simple consiste à choisir :

$$t^k = \frac{1}{\sqrt{k}}. \quad [19]$$

Ce choix assure que  $d(x^k, X^*) \rightarrow 0$  où  $d$  est la fonction distance d'un point à un ensemble. La convergence de  $d(x^k, X^*)$  vers 0 peut malheureusement être très lente car le pas de descente  $t^k$  tend vers 0. D'autres choix ont été proposés, menant à de meilleures propriétés théoriques (voir par exemple (Nesterov, 2009)). Cependant, en théorie, comme d'après notre expérience, les techniques de descentes de sous-gradients ne permettent d'obtenir que des précisions très modérées.

Elles peuvent être intéressantes dans des applications où la réduction du temps de calcul est fondamental : une dizaine d'itérations avec des pas  $t^k$  précalculés permet d'obtenir des approximations grossières. Pour des applications où la précision est importante, elles doivent être évitées.

### 3.2.2. Forward-Backward splitting

La technique appelée "Forward-Backward splitting" (Lions *et al.*, 1979; Combettes *et al.*, 2005) (dont l'iterative thresholding est un cas particulier) est adaptée au problème suivant :

$$\text{Trouver } x \in \arg \min_{x \in \mathbb{R}^n} f_1(x) + f_2(x) \quad [20]$$

où  $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction convexe, fermée, différentiable à gradient  $L$ -Lipschitz et  $f_2 : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  est une fonction convexe fermée telle que  $\text{dom}(f_2) \neq \emptyset$ . On admet que le problème (20) admet un ensemble de minimiseurs  $X^*$ .

L'algorithme "Forward-Backward splitting" prend la forme suivante :

---

**Algorithme 2:** Algorithme de forward-backward splitting

---

**Entrées :** Le nombre d'itérations  $N$  ;  
Le point de départ  $x^0 \in \text{dom}(f_2)$  ;

**Sorties :**  $x^N$ , une estimée d'un élément de  $X^*$ .

**début**

**pour**  $k$  allant de 0 à  $N - 1$  **faire**  
         $x^{k+1} = \text{prox}_{t^k f_2} ( x^k - t^k \nabla f_1(x^k) )$  .  
    **fin**  
**fin**

---

Pour obtenir la convergence de  $\{x^k\}$  vers  $X^*$ , il suffit de choisir un pas  $t^k \in ]\frac{1}{L}, \frac{2}{L}]$  où  $L$  est la constante de Lipschitz du gradient de  $f_1$  (Combettes *et al.*, 2005).

L'exemple typique de problème que l'on peut résoudre avec cette technique est celui de la reconstruction parcimonieuse avec attache aux données  $l^2$  :

$$\text{Trouver } v \in \arg \min_{v \in \mathbb{R}^p} \frac{1}{2} \|\Phi R v - z\|_2^2 + \tau \|v\|_1. \quad [21]$$

Dans ce cas, en choisissant  $f_1(x) = \frac{1}{2} \|\Phi R v - x_0\|_2^2$  et  $f_2(x) = \tau \|v\|_1$ , on obtient l'itération suivante, connue sous le nom de "iterative thresholding" :

$$v^{k+1} = \text{shrink}_{\tau/L} \left( v^k - \frac{1}{L} R^* \Phi^* (\Phi R v - z) \right) \quad [22]$$

où  $L = \|\Phi R\|^2$ .

Cette technique présente plusieurs avantages : elle a un coût faible par itération, nécessite peu de mémoire et a un taux de convergence bien meilleur que la descente de sous-gradient. Sa principale faiblesse réside dans le fait qu'elle ne s'applique qu'à une petite classe de problèmes : il faut que  $f_1$  soit différentiable et il faut être capable de calculer  $\text{prox}_{f_2}$ . Notons aussi que cet algorithme peut être nettement accéléré à moindre coût. Par exemple, la technique de Barzilai-Borwein permet de générer une suite  $\{t^k\}$  qui fournit de meilleurs résultats empiriques. Une autre stratégie, dont les fondements théoriques sont très solides et qui fonctionne bien en pratique a été proposée par Nesterov (Nesterov, 2007). Nous la détaillons ci-dessous.

### 3.2.3. Accélération de convergence "à la Nesterov"

Dans l'algorithme précédent, la direction de descente à l'itération  $k$  est définie simplement comme  $\nabla f_1(x^k)$ . L'idée de Nesterov est de définir la direction de descente comme un élément de tout le sous-espace vectoriel engendré par

$\{\nabla f_1(x^0), \nabla f_1(x^1), \dots, \nabla f_1(x^k)\}$ . Cette idée est très similaire à celle utilisée pour construire l'algorithme du gradient conjugué linéaire. Nous ne rentrons pas dans les détails de la construction de cet algorithme et renvoyons le lecteur intéressé à (Nesterov, 2009; Weiss, 2009). Cet algorithme s'applique exactement au même formalisme que dans la partie précédente et prend la forme suivante :

---

**Algorithme 3:** Algorithme accéléré de Nesterov

---

**Entrées :** Le nombre d'itérations  $N$  ;  
Le point de départ  $x^0 \in \text{dom}(f_2)$  ;

**Sorties :**  $x^N$ , une estimée d'un élément de  $X^*$ .

**début**

Poser  $A = 0$   
Poser  $g = 0$   
Poser  $t = \frac{2}{L}$   
**pour**  $k$  allant de 0 à  $N - 1$  **faire**  
     $a = \frac{t + \sqrt{t^2 + 4tA}}{2}$   
     $v = \text{prox}_{Af_2} ( x^0 - g )$   
     $y^k = \frac{Ax^k + av}{A + a}$   
     $x^{k+1} = \text{prox}_{f_2/L} ( y^k - \frac{1}{L} \nabla f_1(y^k) )$   
     $g = g + a \nabla f_2(x^{k+1})$   
     $A = A + a$

**fin**

**fin**

---

Théoriquement, cet algorithme assure une décroissance optimale de l'énergie, c'est à dire qu'aucune autre méthode de premier ordre ne peut dépasser le taux de convergence de cet algorithme (à une constante multiplicative près) sur cette classe de problèmes (Nesterov, 2004). Un point faible de cette méthode par rapport à l'algorithme classique de forward-backward splitting est qu'il nécessite plus de mémoire. Il faut en effet conserver les variables  $g$ ,  $y$ ,  $v$  et  $x$  alors que l'algorithme non accéléré ne conserve que  $x$ .

Comme l'algorithme de Nesterov est très similaire à la technique de forward-backward splitting, notre conseil au lecteur est de commencer par implémenter un algorithme de forward-backward, de vérifier qu'il converge, puis d'implémenter les quelques lignes supplémentaires pour obtenir l'algorithme multi-pas de Nesterov. La comparaison des performances sera alors immédiate. Notre expérience a toujours indiqué que l'algorithme multi-pas a une efficacité supérieure à la technique de forward-backward splitting, et nous la recommandons donc sauf si l'algorithme vise à être implémenté sur une architecture disposant de peu de mémoire.



### 3.2.4. Résolution par régularisation des singularités

Les deux algorithmes présentés dans les paragraphes précédents ne sont applicables que si une partie du problème est différentiable. Cette hypothèse est trop forte pour beaucoup d'applications. Par exemple, imaginons que l'on souhaite résoudre :

$$\text{Trouver } u \in \arg \min_{u \in \mathbb{R}^n} \|\Phi u - z\|_1 + \|Du\|_1. \quad [23]$$

Aucun des termes  $\|\Phi x - z\|_1$  et  $\|Dx\|_1$  n'est différentiable et on ne peut donc pas utiliser les techniques de type forward-backward splitting. Une solution originale et plus rapide que les descentes de sous-gradient a été proposée par Nesterov dans (Nesterov, 2005). Cette technique consiste à lisser les non-différentiabilités présentes dans le problème, puis à utiliser des techniques de descentes de gradient accélérées sur le problème régularisé. Nous donnons le principe de cette méthode sur le problème (23) et renvoyons le lecteur intéressé à (Nesterov, 2005; Weiss, 2009) pour une formalisation de ces algorithmes dans un cadre plus général et des variantes.

Le problème (23) peut être remplacé par :

$$\text{Trouver } u \in \arg \min_{u \in \mathbb{R}^n} \|\Phi u - z\|_{1,\epsilon} + \|Du\|_{1,\epsilon} \quad [24]$$

où  $\|x\|_{1,\epsilon} = \sum_{i=1}^n \sqrt{x_i^2 + \epsilon^2}$  et  $\epsilon > 0$ . On peut montrer que lorsque  $\epsilon$  tend vers 0, l'ensemble des solutions du problème (24) tend vers l'ensemble des solutions du problème (23). De plus, le problème (24) étant différentiable à gradient Lipschitz, on peut utiliser des techniques de descente de gradient accéléré pour le résoudre.

Cette technique présente plusieurs avantages :

- En un certain sens (Nemirovsky, 1992), elle a un taux de convergence “optimal”.
- Elle ne nécessite que le calcul de gradient à chaque itération, c'est-à-dire de produits matrice-vecteurs.
- Les solutions du problème (24) sont souvent meilleures en pratique que celles du problème (23). Ceci est dû au fait que les critères non différentiables utilisés abondamment dernièrement ne modélisent pas forcément de manière pertinente les images. Par exemple, il est bien connu que l'effet de staircasing obtenu en utilisant la variation totale est lié à la singularité en 0 de la norme  $l^1$  (Nikolova, 2004).

Les désavantages de cette technique sont les suivants :

- Dans certaines situations telles que l'échantillonnage compressif, régulariser les fonctions est maladroit car ce sont les non différentiabilités qui permettent de justifier le modèle.
- La constante de Lipschitz du gradient varie proportionnellement à  $\frac{1}{\epsilon}$ . Si l'on souhaite obtenir des solutions très précises, il faut choisir  $\epsilon$  très petit et la constante de Lipschitz explose. Ceci a pour effet de ralentir nettement l'algorithme car le pas de descente est petit.

### 3.2.5. Douglas-Rachford splitting

Une technique alternative proposée initialement dans (Lions *et al.*, 1979) puis introduite dans la communauté de traitement du signal par Pesquet et Combettes dans (Combettes *et al.*, 2007) est la technique dite de “Douglas-Rachford” (en référence à l’article (Douglas *et al.*, 1956)). Cette technique est adaptée au problème suivant :

$$\text{Trouver } x \in \arg \min_{x \in \mathbb{R}^n} f_1(x) + f_2(x). \quad [25]$$

où  $f_1 : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  et  $f_2 : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  sont deux fonctions convexes, fermées, telles que le problème (25) admette un ensemble de minimiseurs  $X^* \neq \emptyset$ . On suppose de plus que les  $\text{ri}(f_1) \cap \text{ri}(f_2) \neq \emptyset$ . Cette dernière hypothèse technique est vérifiée dans la grande majorité des cas. En utilisant la même présentation que (Combettes *et al.*, 2007), l’algorithme de Douglas-Rachford prend la forme suivante :

---

#### Algorithme 4: Algorithme de Douglas-Rachford.

---

**Entrées :** Le nombre d’itérations  $N$  ;  
 Le point de départ  $x^0 \in \text{dom}(f_2)$  ;  
 La suite de vecteurs  $\{\lambda^k\}_{k=0}^{N-1}$  ;  
 Le paramètre  $\gamma$  ;

**Sorties :**  $x^N$ , une estimée d’un élément de  $X^*$ .

**début**

**pour**  $k$  allant de 0 à  $N - 1$  **faire**

$$x^{k+\frac{1}{2}} = \text{prox}_{\gamma f_2} ( x^k )$$

$$x^{k+1} = x^k + \lambda^k \text{prox}_{\gamma f_1} ( 2x^{k+\frac{1}{2}} - x^k )$$

**fin**

**fin**

---

Il assure la convergence de  $\{x^k\}$  vers un point de  $X^*$  pour peu que  $\gamma \in ]0, +\infty[$  et que  $\lambda^k \in ]0, 2[$  avec  $\sum_{k=0}^{+\infty} \lambda^k (2 - \lambda^k) = +\infty$ . En pratique, le choix  $\lambda^k = 1$  est souvent fait. Notons finalement que cet algorithme peut être généralisé pour minimiser la somme d’un nombre arbitraire de fonctions  $f_1, f_2, f_3, \dots$ . Un algorithme est présenté par exemple dans (Combettes, 2009).

L’algorithme 4 est n’est utilisable que si les calculs de  $\text{prox}_{f_1}$  et de  $\text{prox}_{f_2}$  ont une solution explicite. Sinon, il faut avoir recours à des calculs itératifs pour calculer ces opérateurs proximaux. De tels exemples sont présentés dans (Chaux *et al.*, 2009) par exemple.

Pour illustrer son fonctionnement, nous proposons un exemple. Supposons que l’on souhaite résoudre (10). Il est alors naturel de poser  $f_1(x) = \Upsilon(\Phi R x - x_0)$  et  $f_2(x) = \tau \|x\|_1$ . Le calcul explicite de  $\text{prox}_{f_1}$  est impossible si on ne fait aucune

hypothèse sur  $\Phi$  et sur  $R$ . Un cas particulier permet tout de même de simplifier le calcul. Si  $\Phi = \text{Id}$  et  $R$  satisfait  $RR^* = \text{Id}$ , alors l'équation (17) donne :

$$\text{prox}_{f_1}(x) = x + R^* \circ (\text{prox}_\Upsilon - \text{Id}) \circ R(x). \quad [26]$$

Il suffit donc de savoir calculer  $\text{prox}_\Upsilon(x)$  pour pouvoir appliquer l'algorithme de Douglas-Rachford.

Nous verrons dans la partie suivante que cet algorithme peut être utilisé différemment en utilisant une séparation d'opérateurs linéaires.

### 3.3. Algorithmes reposant sur une transformation du problème

Comme nous l'avons vu précédemment, la présence des opérateurs linéaires  $R$ ,  $D$  et  $\Phi$  rendent les problèmes (7) et (10) particulièrement difficiles à résoudre. Par exemple, on ne sait pas en général calculer de manière explicite  $\text{prox}_{\|\cdot\|_1 \circ D}$  et on a alors souvent recours à des algorithmes itératifs pour réaliser ce calcul. Dans ce paragraphe, nous présentons plusieurs techniques pour séparer les difficultés liées aux opérateurs linéaires  $R$ ,  $D$  et  $\Phi$  et les difficultés liées aux fonctions  $\|\cdot\|_1$  et  $\Upsilon$ .

#### 3.3.1. Un premier exemple de séparation avec l'algorithme de Douglas-Rachford

Les techniques de "séparation d'opérateurs" reposent toutes sur le même principe : transformer le problème initial en un problème de dimension plus élevée en ajoutant des variables liées entre elles par des contraintes linéaires. Pour l'algorithme de Douglas-Rachford, nous illustrons cette idée à travers un exemple. On a vu précédemment que l'algorithme de Douglas-Rachford est difficile à appliquer au problème suivant :

$$\text{Trouver } u \in \arg \min_{u \in \mathbb{R}^n} \Upsilon(\Phi u - z) + \|Du\|_1 \quad [27]$$

si les opérateurs proximaux associés aux fonctions  $u \mapsto \Upsilon(\Phi u - z)$  et  $u \mapsto \|Du\|_1$  ne peuvent pas être calculés explicitement.

Il est clair que le problème (27) peut être réécrit de la façon suivante :

$$\begin{aligned} \text{Trouver } (u, v, w) \in \arg \min & \quad \Upsilon(w - z) + \|v\|_1 \\ \text{sujet à} & \quad u \in \mathbb{R}^n \\ & \quad v \in \mathbb{R}^p, v = Du \\ & \quad w \in \mathbb{R}^m, w = \Phi u \end{aligned} \quad [28]$$

Le problème initial posé dans  $\mathbb{R}^n$  devient ainsi un problème posé dans  $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m$ .

Pour écrire ce problème de façon plus compacte, on peut utiliser les notations suivantes :

$$x = \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \quad A = \begin{bmatrix} D & -\text{Id} & 0 \\ \Phi & 0 & -\text{Id} \end{bmatrix} \quad [29]$$

$$f_1(x) = \Upsilon(w - z) + \|v\|_1 \text{ et } f_2(x) = \begin{cases} 0 & \text{si } Ax = 0 \\ +\infty & \text{sinon} \end{cases} \quad [30]$$

Le problème (28) devient ainsi :

$$\text{Trouver } x \in \arg \min_{x \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m} f_1(x) + f_2(x) \quad [31]$$

et on est ramené au formalisme de l'algorithme de Douglas-Rachford (25). En effet,  $f_1$  et  $f_2$  sont deux fonctions convexes et satisfont les hypothèses posées en (25). Ce qui est intéressant dans cette nouvelle formulation est que les opérateurs  $\text{prox}_{f_1}$  et  $\text{prox}_{f_2}$  sont maintenant plus faciles à calculer. En effet :

$$\text{prox}_{f_1} \left( \begin{matrix} x \\ v \\ w \end{matrix} \right) = \begin{pmatrix} u \\ \text{prox}_{\Upsilon(\cdot - z)}(v) \\ \text{prox}_{\|\cdot\|_1}(w) \end{pmatrix} \quad [32]$$

et

$$\text{prox}_{f_2} \left( \begin{matrix} x \\ v \\ w \end{matrix} \right) = \arg \min_{x \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m, Ax = 0} \|x - z\|_2^2 \quad [33]$$

$$= (\text{Id} - A^\dagger A)z \quad [34]$$

peut être calculé à l'aide de la pseudo-inverse de  $A$ .

### 3.3.2. Dualité pour les problèmes fortement convexes

L'algorithme que nous allons présenter dans ce paragraphe peut aussi être considéré comme un algorithme de séparation. Il peut être appliqué uniquement lorsque le problème considéré est fortement convexe. Cet algorithme s'applique à la classe de problèmes suivante :

$$\text{Trouver } x \in \arg \min_{x \in \mathbb{R}^n} f_1(Ax) + f_2(x) \quad [35]$$

où  $f_1 : \mathbb{R}^p \rightarrow \mathbb{R} \cup \{+\infty\}$  est une fonction convexe, fermée,  $f_2 : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  est une fonction fortement convexe, fermée, de paramètre de forte convexité  $\nu > 0$  et  $A : \mathbb{R}^n \rightarrow \mathbb{R}^p$  est une transformée linéaire. On suppose de plus que  $\text{Ari}(f_1)$  et  $\text{ri}(f_2)$  ont une intersection non vide. Ces conditions sont suffisantes pour assurer l'existence et l'unicité de la solution de (35).

Tout comme dans l'exemple précédent, l'idée consiste à remplacer le problème (35) par un problème de plus grande dimension :

$$\text{Trouver } (x, u) \in \arg \min_{\substack{x \in \mathbb{R}^n \\ u \in \mathbb{R}^p, u = Ax}} f_1(u) + f_2(x) \quad [36]$$

Ce problème peut encore être réécrit sous la forme lagrangienne :

$$\begin{aligned} \text{Trouver } (x, u) \in \arg \min & \quad \max_{\lambda \in \mathbb{R}^p} (f_1(u) + f_2(x) + \langle \lambda, Ax - u \rangle) \quad . \quad [37] \\ \text{sujet à} & \quad x \in \mathbb{R}^n \\ & \quad u \in \mathbb{R}^p \end{aligned}$$

Les résultats de dualité de Fenchel-Rockafellar (Rockafellar, 1997) assurent que le min et le max peuvent être intervertis dans ce cas. Le problème dual s'écrit alors :

$$\begin{aligned} \text{Trouver } \lambda \in \arg \max & \quad \min_{u \in \mathbb{R}^p} (f_1(u) - \langle \lambda, u \rangle) + \min_{x \in \mathbb{R}^n} (f_2(x) + \langle A^* \lambda, x \rangle) \quad . \\ \text{sujet à} & \quad \lambda \in \mathbb{R}^p \end{aligned} \quad [38]$$

Soit encore, en utilisant la définition de la transformée de Fenchel :

$$\begin{aligned} \text{Trouver } \lambda \in \arg \min & \quad f_1^*(\lambda) + f_2^*(-A^* \lambda) \quad . \quad [39] \\ \text{sujet à} & \quad \lambda \in \mathbb{R}^p \end{aligned}$$

Le problème (39) est souvent appelé problème dual à (35). L'intérêt de cette formulation duale est que sous l'hypothèse de forte convexité de  $f_2$ ,  $f_2^*$  est différentiable à gradient  $\frac{1}{\nu}$ -Lipschitz. Le problème (39) est donc la somme d'une fonction différentiable et d'une fonction convexe et les algorithmes de forward-backward splitting ou de Nesterov peuvent être appliqués. De plus, la solution  $x^*$  de (35) est liée aux solutions  $\lambda^*$  de (39) par la relation suivante :  $x^* = \nabla f_2^*(-A^* \lambda)$ . Résoudre le problème dual permet donc de retrouver la solution primale.

Nous illustrons le principe de cet algorithme sur le problème suivant :

$$\begin{aligned} \text{Trouver } u \in \arg \min & \quad \frac{1}{2} \|u - z\|_2^2 + \|Du\|_1 \quad [40] \\ \text{sujet à} & \quad u \in \mathbb{R}^n \end{aligned}$$

La fonction  $u \mapsto \frac{1}{2} \|u - z\|_2^2$  est fortement convexe de paramètre de forte convexité 1. Le problème dual (39) s'écrit dans ce cas :

$$\begin{aligned} \text{Trouver } \lambda \in \arg \min & \quad \frac{1}{2} \|D^* \lambda\|_2^2 - \langle x^0, D^* \lambda \rangle \quad . \quad [41] \\ \text{sujet à} & \quad \lambda \in \mathbb{R}^p, \|\lambda\|_\infty \leq 1 \end{aligned}$$

Ce problème peut être résolu par une technique de type descente de gradient projeté :

$$\lambda^{k+1} = \Pi_{\{\lambda, \|\lambda\|_\infty \leq 1\}} (\lambda^k - t^k D(D^* \lambda^k - x^0)), \quad [42]$$

ou par les techniques accélérées de Nesterov (Weiss *et al.*, 2009). Le principe décrit ici est une généralisation de l'algorithme de Chambolle (Chambolle, 2004).

### 3.3.3. La technique des directions alternées

Le dernier algorithme que nous présentons est celui que nous avons utilisé pour réaliser nos comparaisons expérimentales. Il présente l'avantage d'être très général et

de mener à des temps de calculs très raisonnables. Cet algorithme est adapté à la classe de problèmes suivante :

$$\begin{aligned} \text{Trouver } (x, y) \in \arg \min & \quad f_1(x) + f_2(y) \\ \text{sujet à} & \quad Ax + By = a \\ & \quad x \in \mathbb{R}^n, y \in \mathbb{R}^m \end{aligned} \quad [43]$$

où :

–  $f_1 : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  et  $f_2 : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$  sont deux fonctions convexes et fermées.

–  $A \in \mathbb{R}^{l \times n}$  et  $B \in \mathbb{R}^{l \times m}$  sont deux applications linéaires.

–  $a \in \mathbb{R}^l$  est un vecteur donné.

L'algorithme des directions alternées repose fortement sur un outil appelé lagrangien augmenté. En attachant un multiplicateur de Lagrange  $\lambda \in \mathbb{R}^l$  à la contrainte linéaire (43), le lagrangien augmenté s'écrit :

$$\mathcal{L}(x, y, \lambda) := f_1(x) + f_2(y) + \langle \lambda, Ax + By - a \rangle + \frac{\beta}{2} \|Ax + By - a\|_2^2 \quad [44]$$

où  $\beta > 0$  est un paramètre qui permet de favoriser la contrainte linéaire  $Ax + By = a$  (Glowinski, 1984). L'idée de la technique de minimisation par directions alternées consiste à minimiser le lagrangien augmenté de façon alternative suivant  $x$ ,  $y$  puis  $\lambda$  de manière à trouver un point selle du lagrangien augmenté. Ce schéma s'écrit ainsi :

---

**Algorithme 5:** Technique des directions alternées pour résoudre (43)

---

**Entrées :** Le nombre maximal d'itérations  $N$  ;

Un point de départ  $x^0 \in \mathbb{R}^n$  ;

Un point de départ  $y^0 \in \mathbb{R}^m$  ;

Un point de départ  $\lambda^0 \in \mathbb{R}^l$  ;

La valeur des paramètres  $\gamma > 0$  et de  $\beta > 0$  ;

**Sorties :**  $(x^N, y^N)$ , une estimée d'un élément de  $X^* \times Y^*$ .

**début**

**pour**  $k$  allant de 0 à  $N - 1$  **faire**

**Etape 1.**  $x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \mathcal{L}(x, y^k, \lambda^k)$  .

**Etape 2.**  $y^{k+1} = \arg \min_{y \in \mathbb{R}^m} \mathcal{L}(x^{k+1}, y, \lambda^k)$  .

**Etape 3.**  $\lambda^{k+1} = \lambda^k + \beta\gamma(Ax^{k+1} + By^{k+1} - a)$ .

**fin**

**fin**

---

Le paramètre  $\gamma$  est un paramètre de relaxation qui doit appartenir à l'intervalle  $]0, \frac{\sqrt{5}+1}{2}[$  pour que  $\{(x^k, y^k)\}$  converge vers l'ensemble des minimiseurs (Glowinski,

1984). Le paramètre  $\beta$  permet de contrôler l'attache à la contrainte  $Ax + By = a$ . Plus il est grand, plus cette contrainte sera respectée au fil des itérations.

Nous illustrons le fonctionnement de ce schéma, l'appliquant au problème inverse (7) :

$$\text{Trouver } u \in \arg \min_{u \in \mathbb{R}^n} \Upsilon(\Phi u - z) + \tau \|Du\|_1 \quad [45]$$

Pour transformer le problème (45) en (43), on remarque d'abord qu'il est équivalent à :

$$\begin{aligned} \text{Trouver } (u, v, w) \in \arg \min & \quad \Upsilon(v - z) + \tau \|w\|_1 \\ \text{sujet à} & \quad u \in \mathbb{R}^n \\ & \quad v \in \mathbb{R}^m, v = \Phi u \\ & \quad w \in \mathbb{R}^p, w = Du \end{aligned} \quad [46]$$

En posant :

$$x = \begin{pmatrix} v \\ w \end{pmatrix} \in \mathbb{R}^m \times \mathbb{R}^p, \quad B = \begin{bmatrix} \Phi \\ D \end{bmatrix}, \quad A = -\text{Id}, \quad a = \begin{pmatrix} z \\ 0 \end{pmatrix} \quad [47]$$

$$f_1(x) = \Upsilon(v) + \tau \|w\|_1 \quad \text{et} \quad f_2(u) = 0 \quad [48]$$

le problème (45) est peut donc être réécrit :

$$\begin{aligned} \text{Trouver } (u, x) \in \arg \min & \quad f_1(x) + f_2(u) \\ \text{sujet à} & \quad Ax + Bu = a \\ & \quad x \in \mathbb{R}^m \times \mathbb{R}^p, u \in \mathbb{R}^n \end{aligned} \quad [49]$$

Avec ces notations, le lagrangien augmenté s'écrit :

$$\mathcal{L}(x, u, \lambda) := f_1(x) + \langle \lambda, Bu - x - a \rangle + \frac{\beta}{2} \|Bu - a - x\|_2^2 \quad [50]$$

La première étape de l'algorithme consiste à calculer :

$$\begin{aligned} x^{k+1} &= \arg \min_{x \in \mathbb{R}^m \times \mathbb{R}^p} f_1(x) + \langle \lambda^k, Bu^k - x - a \rangle + \frac{\beta}{2} \|Bu^k - a - x\|_2^2 \\ &= \arg \min_{x \in \mathbb{R}^m \times \mathbb{R}^p} \frac{1}{\beta} f_1(x) + \frac{1}{2} \|Bu^k - a + \frac{\lambda^k}{\beta} - x\|_2^2 \\ &= \text{prox}_{\frac{1}{\beta} f_1} \left( Bu^k - a + \frac{\lambda^k}{\beta} \right) \end{aligned} \quad [51]$$

avec :

$$\text{prox}_{\frac{1}{\beta} f_1} (x) = \begin{pmatrix} \text{prox}_{\frac{1}{\beta} \Upsilon} (v) \\ \text{shrink}_{\tau/\beta} (w) \end{pmatrix} \quad [52]$$

La deuxième étape de l'algorithme consiste à résoudre :

$$\begin{aligned} u^{k+1} &= \arg \min_{u \in \mathbb{R}^n} \langle \lambda^k, Bu - x^k - a \rangle + \frac{\beta}{2} \|Bu - a - x^k\|_2^2 \\ &= \arg \min_{u \in \mathbb{R}^n} \frac{\beta}{2} \|Bu + \frac{\lambda^k}{\beta} - a - x^k\|_2^2 . \end{aligned} \quad [53]$$

$u^{k+1}$  est donc la solution du système linéaire semi défini positif :

$$B^*Bu = B^* \left( a + x^k - \frac{\lambda^k}{\beta} \right). \quad [54]$$

On remarque ici qu'il s'agit d'inverser  $B^*B = \Phi^*\Phi + D^*D$  ce qui est en général mieux conditionné  $\Phi^*\Phi$ . Très souvent en imagerie, la transformée  $B^*B$  possède une structure particulière qui peut être exploitée numériquement pour résoudre (54). Les exemples typiques sont les matrices creuses ou les matrices circulantes. Cette remarque est utilisée dans (Ng *et al.*, 2009) par exemple pour obtenir des algorithmes rapides. Les systèmes peuvent aussi être résolus par des techniques standard de type gradient conjugué. Dans les expériences menées dans la partie expérimentale, pas plus de 6 itérations de gradient conjugué permettent d'obtenir une précision suffisante. Notons que les sous-problèmes 1 et 2 peuvent être résolus de manière inexacte, ce qui préserve la convergence de l'algorithme (He *et al.*, 2002) sous certaines conditions.

Finalement, l'algorithme permettant de résoudre (45) s'écrit :

---

**Algorithme 6:** Technique des directions alternées pour résoudre (45)

---

**Entrées :** Le nombre maximal d'itérations  $N$  ;

Un point de départ  $x^0 = \begin{pmatrix} v^0 \\ w^0 \end{pmatrix} \in \mathbb{R}^m \times \mathbb{R}^p$  ;

Un point de départ  $u^0 \in \mathbb{R}^n$  ;

Un point de départ  $\lambda^0 \in \mathbb{R}^l$  ;

Deux paramètres  $\gamma > 0$  et  $\beta > 0$  ;

**Sorties :**  $u^N$ , une estimée d'un élément de  $X^*$ .

**début**

**pour**  $k$  allant de 0 à  $N - 1$  **faire**

**Etape 1**  $x^{k+1} = \text{prox}_{\frac{1}{\beta}f_1} \left( Bu^k - a + \frac{\lambda^k}{\beta} \right)$

**Etape 2.**  $u^{k+1} = (B^*B)^{-1}B^* \left( a + x^k - \frac{\lambda^k}{\beta} \right)$

**Etape 3.**  $\lambda^{k+1} = \lambda^k + \beta\gamma(Ax^{k+1} + Bu^{k+1} - a)$ .

**fin**

**fin**

---



Cet algorithme est très flexible et permet de résoudre aussi bien des problèmes contraints que lagrangiens, avec tous types de bruits. La seule différence si on change le terme d'attache aux données est le terme  $\text{prox}_{\frac{1}{\beta}} \gamma ( w_1 )$  qui intervient à l'étape 1.

### 3.4. Tableau récapitulatif

Dans ce paragraphe, nous résumons les principaux aspects des algorithmes présentés dans cet article. Les techniques de minimisation présentées s'appliquent toutes au formalisme suivant :

$$\text{Trouver } \min_{x \in \mathbb{R}^n} f(x) = f_1(Ax) + f_2(Bx) \quad [55]$$

où  $f_1$  et  $f_2$  sont deux fonctions convexes fermées de domaine non vide et  $A$  et  $B$  sont deux transformées linéaires. Malheureusement, pour le moment, il ne semble pas qu'il existe un algorithme qui soit le mieux adapté (en terme de mémoire, temps de calcul et précision) à la minimisation de toutes les instances de (55) et nous espérons que cet article aidera le lecteur à orienter ses choix vers une des différentes méthodes existantes, en fonction de son problème. Ce choix doit dépendre fortement des caractéristiques du problème telles que la différentiabilité ou la différentiabilité partielle, la capacité de résoudre rapidement des systèmes linéaires, le calcul d'opérateurs proximaux...

Le tableau 2 résume les caractéristiques de chaque algorithme. Les colonnes  $f_1$  et  $f_2$  indiquent le type d'hypothèse nécessaire en plus de la convexité pour appliquer l'algorithme. Les colonnes  $\text{prox}_{f_1}$  et  $\text{prox}_{f_2}$  indiquent s'il faut être capable de calculer les opérateurs proximaux des fonctions  $f_1$  et  $f_2$ . La colonne **Pseudo-inv.** indique s'il faut être capable de calculer la pseudo inverse de  $A$  et/ou de  $B$ . La colonne **Sys. lin.** indique s'il est nécessaire de résoudre des systèmes linéaires. La colonne **Taux de conv.** indique les taux de convergences théoriques connus à ce jour pour les différentes méthodes. Un taux de convergence en  $O\left(\frac{1}{k}\right)$  indique que la suite  $\{x^k\}$  satisfait  $f(x^k) - f(x^*) \leq O\left(\frac{1}{k}\right)$  où  $x^*$  est une solution du problème de minimisation.

### 3.5. Discussion

Nous avons présenté de nombreux algorithmes et espérons avoir donné quelques pistes au lecteur pour lui permettre de résoudre son problème. Pour un même problème, il est probable que plusieurs algorithmes soient des candidats potentiels et le lecteur pourrait apprécier voir des comparaisons expérimentales. Nous n'avons pas mis de courbes comparatives de l'efficacité des méthodes, car ceci nous semble forcément incomplet.

En effet, d'un problème à une autre, la liste des candidats utilisables change fortement et le candidat idéal dépend de très nombreux paramètres (structures des transformées linéaires (inversibilité, Toeplitz, rang faible, creuse,...), structures des fonctions

	$f_1$	$f_2$	$\text{prox}_{f_1}$	$\text{prox}_{f_2}$	Pseudo-inv.	Sys. lin.	Taux de conv.
Sous-gradient projeté	Lipschitz	Lipschitz	✗	✗	✗	✗	$O\left(\frac{1}{\sqrt{k}}\right)$
Forward-Backward	$\nabla f_1$ Lipschitz	-	✗	✓	✗	✗	$O\left(\frac{1}{k}\right)$
Nesterov	$\nabla f_1$ Lipschitz	-	✗	✓	✗	✗	$O\left(\frac{1}{k^2}\right)$
Lissage	-	-	✓	✓	✗	✗	$O\left(\frac{1}{k}\right)$
Douglas-Rachford (1)	-	-	$\text{prox}_{f_1 \circ A}$	$\text{prox}_{f_1 \circ B}$	✗	✗	mal connu
Douglas-Rachford (2)	-	-	✓	✓	✓	✗	mal connu
Méthode duale	-	Fortement convexe	✓	✓	✗	✗	$O\left(\frac{1}{k^2}\right)$
Directions alternées	-	-	✓	✓	✗	✓	mal connu

**Tableau 2.** Quelques caractéristiques des différents algorithmes présentés dans cet article (✓ pour oui, ✗ pour non). Nous rappelons que  $f_1$  et  $f_2$  sont au minimum des fonctions convexes fermées.

(Lipschitz, fortement convexes, simples), structures des contraintes (pas de contrainte, contraintes simples, contraintes complexes), structure de l'ordinateur (calcul simple, FPGA, parallélisation CPU/GPU)).

Nous nous permettons tout de même une recommandation pour les problèmes classiques en imagerie : les techniques de directions alternées (qui peuvent être dérivées des algorithmes de type Douglas-Rachford), sont très flexibles et s'adaptent à de nombreux problèmes, pour peu que les transformées linéaires soient rapidement inversibles. Un lecteur encore peu averti souhaitant s'attaquer rapidement une grande classe de problèmes différents peut donc avoir intérêt à commencer à tester les techniques de directions alternées ou de Douglas-Rachford.

#### 4. Conclusion

Nous avons présenté dans cet article des méthodes de résolution de problèmes inverses régularisés sur des normes  $l^1$ . Différents modèles de régularisation reposant sur des normes  $l^1$  ont été décrits et nous avons tenté de les comparer théoriquement et expérimentalement. Il n'existe malheureusement pas de méthode générale, les performances tant en termes de temps de calcul que de qualité visuelle ou numérique dépendent du problème considéré. Le type de régularisation et le type d'algorithme doivent être adaptés à chaque problème et nous avons tenté de donner les informations nécessaires pour le choix de ces éléments.

## 5. Bibliographie

- Aujol J.-F., « Some first-order algorithms for total variation based image restoration », *Journal of Mathematical Imaging and Vision*, vol. 34, n° 3, p. 307-327, juillet, 2009.
- Bect J., Blanc-Féraud L., Aubert G., Chambolle A., « A  $l^1$ -unified variational framework for image restoration », *Proc. European Conference on Computer Vision*, Prague, République tchèque, p. 1-13, mai, 2004.
- Boyd S., Vandenberghe L., *Convex Optimization*, Cambridge University Press, mars, 2004.
- Candès E., Demanet L., Donoho D., Ying L., « Fast Discrete Curvelet Transforms », *Multiscale Modeling & Simulation*, vol. 5, n° 3, p. 861-899, 2006.
- Chambolle A., « An algorithm for total variation minimization and applications », *Journal of Mathematical Imaging and Vision*, vol. 20, n° 1, p. 89-97, 2004.
- Chambolle A., DeVore R. A., Lee N.-Y., Lucier B. J., « Nonlinear Wavelet Image Processing : Variational Problems, Compression, and Noise Removal through Wavelet Shrinkage », *IEEE Trans. Image Processing*, vol. 7, p. 319-335, 1996.
- Chaux C., Pesquet J.-C., Pustelnik N., « Nested iterative algorithms for convex constrained image recovery problems », *SIAM Journal on Imaging Sciences*, Juin, 2009.
- Chen S. S., Donoho D. L., Saunders M. A., « Atomic Decomposition by Basis Pursuit », *SIAM Journal on Scientific Computing*, vol. 20, n° 1, p. 33-61, 1998.
- Combettes P. L., « Iterative construction of the resolvent of a sum of maximal monotone operators », *Journal of Convex Analysis*, vol. 16, n° 4, p. 727-748, 2009.
- Combettes P. L., Pesquet J.-C., « A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery », *IEEE Journal of Selected Topics in Signal Processing*, décembre, 2007.
- Combettes P. L., Pesquet J.-C., *Proximal Splitting Methods in Signal Processing*, Springer-Verlag, New York, 2010.
- Combettes P. L., Wajs V. R., « Signal Recovery by Proximal Forward-Backward Splitting », *Multiscale Modeling & Simulation*, vol. 4, n° 4, p. 1168-1200, 2005.
- Daubechies I., *Ten lectures on wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.
- Daubechies I., Defrise M., Mol C. D., « An iterative thresholding algorithm for linear inverse problems with a sparsity constraint », *Communications on Pure and Applied Mathematics*, vol. 57, n° 11, p. 1413-1457, 2004.
- Donoho D., « De-noising by soft-thresholding », *IEEE transactions on information theory*, vol. 41, n° 3, p. 613-627, 1995.
- Douglas J., Rachford H., « On the numerical solution of heatconduction problems in two or three space variables », *Trans.Amer.Math.Soc.*, décembre, 1956.
- Dupé F., Fadili J., Starck J., « A proximal iteration for deconvolving Poisson noisy images using sparse representations », *IEEE Transactions on Image Processing*, vol. 18, n° 2, p. 310-321, 2009.
- Elad M., Milanfar P., Rubinstein R., « Analysis versus synthesis in signal priors », *Inverse Problems*, vol. 23, n° 3, p. 947-968, 2007.
- Figueiredo M. A. T., Nowak R. D., « An EM algorithm for wavelet-based image restoration », *IEEE Transactions on Image Processing*, vol. 12, n° 8, p. 906-916, août, 2003.

- Fuchs J., « Recovery of exact sparse representations in the presence of bounded noise », *IEEE Transactions on Information Theory*, vol. 51, n° 10, p. 3601-3608, 2005.
- Gaetan C., Guyon X. (eds), *Modélisation et statistique spatiales*, Springer, 2008.
- Geman S., Geman D., « Stochastic relaxation, Gibbs distributions, and the Bayesian relation of images », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, p. 721-741, 1984.
- Gibson J. D., Bovik A. (eds), *Handbook of Image and Video Processing*, Academic Press, Inc., Orlando, FL, USA, 2000.
- Glowinski R., *Numerical Methods for Nonlinear variation al Problems*, Springer-Verlag, 1984.
- Golub G., Van Loan C., *Matrix computations*, Johns Hopkins Univ Pr, 1996.
- Hamza B., Krim H., « Image denoising : A nonlinear robust statistical approach », *IEEE Transactions on Signal Processing*, vol. 49, n° 12, p. 3045-3054, 2001.
- He B., Liao L.-Z., Han D., Yang H., « A new inexact alternating directions method for monotone variational inequalities », *Journal on Mathematical Programming*, vol. 92, n° 1, p. 103-118, 2002.
- Idier J. (ed.), *Approche bayésienne pour les problèmes inverses*, Traité IC2, Série traitement du signal et de l'image, Hermès, Paris, novembre, 2001.
- Kirsch A., *An introduction to the mathematical theory of inverse problems*, Springer Verlag, 1996.
- Le Pennec E., Mallat S., « Sparse geometric image representations with bandelets », *IEEE Transactions on Image Processing*, vol. 14, n° 4, p. 423-438, 2005.
- Lions P., Mercier B., « Splitting algorithms for the sum of two nonlinear operators », *SIAM Journal on Numerical Analysis*, vol. 16, n° 6, p. 964-979, 1979.
- Lucy L., « An iterative technique for rectification of observed distributions », *The Astronomical Journal*, vol. 79, n° 6, p. 745-765, 1974.
- Nemirovsky A., « Information-based complexity of linear operator equations », *Journal of Complexity*, vol. 8, n° 2, p. 175, 1992.
- Nesterov Y., *Introductory lectures on convex optimization. A basic course*, Kluwer, 2004.
- Nesterov Y., « Smooth minimization of non smooth functions », *Mathematical Programming, Ser. B*, 2005.
- Nesterov Y., « Gradient methods for minimizing composite objective function », *CORE Discussion Paper 2007/76*, 2007.
- Nesterov Y., « Primal-dual subgradient methods for convex problems », *Mathematical Programming, Ser. B*, 2009.
- Nesterov Y., Nemirovsky A., *Interior-Point polynomial methods in convex programming*, vol. 13, SIAM Studies in Applied Mathematics, 1994.
- Ng M., Weiss P., Yuang X., « Solving Constrained Total-Variation Image Restoration and Reconstruction Problems via Alternating Direction Methods », *ICM Research Report*, 2009.
- Nikolova M., « A variational approach to remove outliers and impulse noise », *Journal of Mathematical Imaging and Vision*, vol. 20, n° 1, p. 99-120, 2004.
- Polyak B., *Introduction to optimization*, Optimization Software, Publications Division, New York, 1987.
- Rockafellar R., *Convex analysis*, Princeton university press, 1997.

- Rudin L. I., Osher S., Fatemi E., « Nonlinear total variation based noise removal algorithms », *Physica D*, vol. 60, p. 259-268, 1992.
- Selesnick I. W., Baraniuk R. G., Kingsbury N. G., « The Dual-Tree Complex Wavelet Transform », *IEEE Signal Processing Magazine*, vol. 22, n° 6, p. 123-151, Nov., 2005.
- Selesnick I. W., Figueiredo M. A. T., « Signal restoration with overcomplete wavelet transforms : comparison of analysis and synthesis priors », vol. 7446, SPIE, 2009.
- Weiss P., *Algorithmes rapides d'optimisation convexe. Application à la reconstruction d'images et à la détection de changements.*, Thèse de doctorat à l'université de Nice Sophia-Antipolis, 2009.
- Weiss P., Blanc-Féraud L., Aubert G., « Efficient schemes for total variation minimization under constraints in image processing », *SIAM journal on Scientific Computing*, vol. 31, n° 3, p. 2047-2080, 2009.

SERVICE ÉDITORIAL – HERMES-LAVOISIER  
14 rue de Provigny, F-94236 Cachan cedex  
Tél. : 01-47-40-67-67  
E-mail : [revues@lavoisier.fr](mailto:revues@lavoisier.fr)  
Serveur web : <http://www.revuesonline.com>