



HAL
open science

Un exploration numérique du traitement des signaux, des images et des surfaces

Gabriel Peyré

► **To cite this version:**

Gabriel Peyré. Un exploration numérique du traitement des signaux, des images et des surfaces. Matapli, 2011, 94, pp.41-64. hal-00558668

HAL Id: hal-00558668

<https://hal.science/hal-00558668>

Submitted on 23 Jan 2011

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.

Un exploration numérique du traitement des signaux, des images et des surfaces

Gabriel Peyré*

January 23, 2011

Abstract

L'apport des mathématiques appliquées en traitement du signal et des images est indéniable. Les mathématiques ont permis de faire émerger des concepts fondateurs tels que les équations aux dérivées partielles, les ondelettes ou l'estimation statistique. Dans ce domaine, les avancées théoriques vont de paire avec le développement de nouveaux algorithmes de calcul. Les *Numerical Tours of Signal Processing* (qu'on pourrait traduire en français par *une exploration numérique du traitement du signal*) constituent une collection de travaux pratiques pour appréhender à la fois les mathématiques et les algorithmes. Ils sont accessibles en ligne à l'adresse

www.numerical-tours.com

Ces travaux pratiques permettent de suivre étape par étape l'implémentation en Matlab ou en Scilab de nombreux algorithmes de traitement des signaux, des images et des surfaces. Ces TPs constituent une source importante d'information pour les chercheurs qui veulent apprendre des nouvelles méthodes parmi l'état de l'art du domaine. Mais ils sont aussi destinés à aider les étudiants (depuis la licence jusqu'au doctorat) pour apprendre les fondations théoriques et numériques du traitement du signal, des images et des surfaces.

1 Introduction

Le traitement moderne du signal (au sens large : signal 1D, image 2D, vidéo 2D+t, surface 3D, etc.) est confronté à une explosion de la complexité des données. Qu'il s'agisse de signaux audio, d'images couleur, de films ou de surfaces, la taille et la diversité de ces données rendent nécessaire le développement d'une science à la fois mathématique et numérique. Ces données doivent être restaurées, compressées ou analysées à l'aide d'algorithmes rapides qui exploitent des modèles mathématiques fins. Comprendre en profondeur ces modèles et les méthodes de traitement nécessite un échange permanent entre l'analyse mathématique et le calcul scientifique. Nous détaillons dans cet article une nouvelle plateforme en ligne, les *Numerical Tours of Signal Processing*, dont le but est d'aider l'utilisateur éclairé à maîtriser les techniques mathématiques et numériques du traitement du signal.

Cette plateforme présente plus d'une centaine de travaux pratiques (TPs) touchant à toutes les facettes des mathématiques du signal et de l'image. Elle sera utile pour les mathématiciens appliqués souhaitant comprendre plus en détail les problématiques du domaine. Elle permettra aussi aux ingénieurs en imagerie (médicale, sismique, astrophysique, etc.) de rendre plus concret certains concepts mathématiques. Chaque TP peut être appliqué sur n'importe quel signal, image ou surface, ce qui permet de confronter les méthodes aux données favorites de chaque discipline.

Apprendre les fondations théoriques du traitement du signal. L'utilisation de ressources en ligne en conjonction avec des logiciels de calcul scientifique comme Matlab¹ ou Scilab² est particulièrement adapté à l'apprentissage du traitement du signal [26]. Ces *Numerical Tours* sont cependant à mettre à part dans l'ensemble des ressources en ligne pour le traitement du signal. En

*CNRS et CEREMADE Université Paris-Dauphine, 75775 Paris Cedex 16 France, email : gabriel.peyre@ceremade.dauphine.fr

¹<http://www.mathworks.com/>

²<http://www.scilab.org/>

effet, ces TPs ont un contenu mathématique plus conséquent que la plus part des logiciels et sites web. L'accent est mis principalement sur l'échange nécessaire entre les méthodes mathématiques (et en particulier les formules et équations définissant les méthodes) et l'implémentation pratique sur machine. Cette base mathématique permet d'aller plus loin que la plus part des ressources classiques pour l'enseignement, qui utilisent principalement des notions élémentaires telles que le filtrage ou l'analyse de Fourier.

L'expérimentation numérique pour apprendre le traitement du signal. De l'avis de l'auteur, l'expérimentation numérique est très importante pour comprendre les outils théoriques et numériques développés récemment en traitement du signal. Un cours de Master typique en mathématiques appliquées au signal devrait alterner entre la description des outils, l'analyse théorique de leur performance, et l'expérimentation numérique sur des données synthétiques et réelles. Ces *Numerical Tours* ont été créés dans le but de rendre cette alternance entre théorie et pratique la plus simple possible à la fois pour l'enseignant et pour les étudiants. Ils permettent à l'enseignant d'avoir une plateforme d'expérimentation simple à utiliser. Ils permettent aux étudiants d'implémenter les méthodes et de comparer différentes techniques sur leurs propres données. Cette pratique leur permet également de mieux comprendre les concepts mathématiques, car les formules mathématiques se traduisent de façon très directe en des algorithmes rapides et élégants. Ceci leur permet de prendre conscience que le traitement numérique du signal est une science à part, qui nécessite une grande rigueur à la fois pour l'analyse théorique et pour l'expérimentation numérique [1].

Recherche et éducation reproductible. La recherche reproductible est un concept central du traitement des signaux et des images, voir par exemple [32, 17]. Un travail scientifique adhérent à la recherche reproductible offre, en même temps que la publication des travaux dans un journal, l'ensemble des logiciels permettant de reproduire les figures de l'article. Ces logiciels peuvent être également utilisés sur d'autres données. Nous pensons que cette recherche reproductible va de paire avec une éducation reproductible du traitement du signal, qui amène les étudiants à comprendre par eux même les concepts mathématiques et numériques en jeu, en les appliquant à leur propres données. Les TPs de ces *Numerical Tours* vont dans ce sens, en tentant de combler le fossé existant entre les concepts mathématiques de base (tels que Fourier, ondelettes ou EDPs) et les techniques plus avancées développées par les chercheurs.

2 Présentation des *Numerical Tours*

2.1 Côté utilisateurs ...

Les *Numerical Tours* constituent un ensemble de pages web, accessibles en ligne à l'adresse

`www.numerical-tours.com`

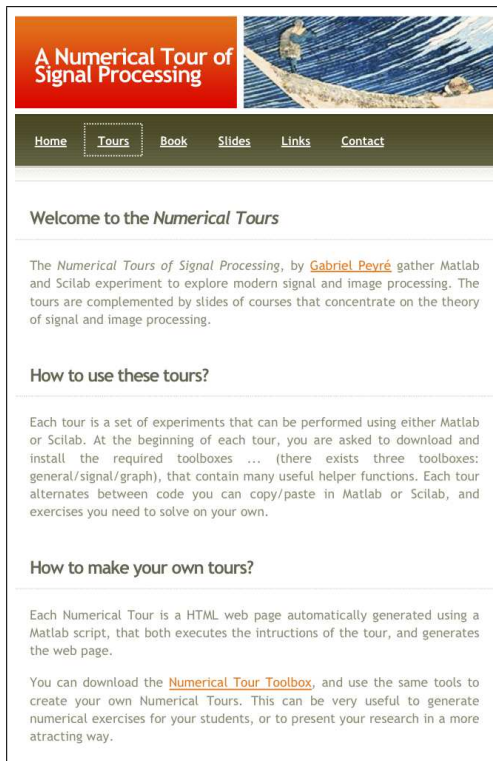
La figure 1, gauche, montre la page d'accueil des *Numerical Tours*.

Chaque page constitue un travail pratique (TP) indépendant. Un TP alterne entre des explications des méthodes mathématiques, avec des équations, des instructions Matlab/Scilab, et des figures montrant les résultats des instructions. Chaque TP est également entrecoupé d'exercices, où il est demandé à l'étudiant d'accomplir une tâche précise, dont le résultat est montré sous forme d'une figure. La figure 1, droite, montre le début d'un TP.

Il y a à l'heure actuelle plus de 100 TPs différents, mais ce nombre est en croissance constante. L'ensemble des TPs est regroupé en une vingtaine de sujets, tels que les ondelettes, l'infographie, le traitement du son, etc.

2.2 ...et côté développeurs

Chaque TP (c'est-à-dire la page web, avec le texte, les équations, les figures et les exercices) est entièrement créée à partir d'un seul fichier Matlab. C'est le seul fichier qui doit être édité et maintenu à jour par un auteur souhaitant contribuer aux *Numerical Tours*. Ce script contient à la fois les instructions Matlab/Scilab qui effectuent les calculs, des commentaires Matlab standards, et



Apply high and low filtering+subsampling in the horizontal direction (coordinate), to get coarse and details.

```
Coarse = subsampling(cconv(A,h,2),2);
Detail = subsampling(cconv(A,g,2),2);
```

Concatenate them in the horizontal direction to get the result.

```
A = cat3(2, Coarse, Detail );
```


Assign the transformed data.

```
MW(1:2^(j+1),1:2^(j+1)) = A;
```

Display the result of the horizontal transform.

```
clf;
imageplot(M,'Original image',1,2,1);
subplot(1,2,2);
plot_wavelet(MW,log2(n)-1); title('Transformed')
```

Original image



Transformed

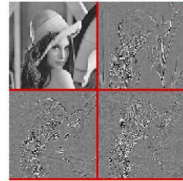


Figure 1: Gauche : page d'accueil des Numerical Tours. Droite : extrait d'un TP.

des commentaires spéciaux pour afficher du texte, des équations, des références bibliographiques, des liens HTML, etc.

Ce système d'écriture est à la fois simple à utiliser et à maintenir. Il garantit que les scripts Matlab que les étudiants doivent utiliser marchent et produisent les résultats affichés dans les figures. Avec un peu d'habitude, il est possible d'écrire un *Numerical Tours* complet en quelques minutes pour des concepts simples, et en quelques heures pour un sujet avancé.

La figure 2, gauche, montre le code source Matlab pour un TP très simple. Il contient un titre, un sous-titre, une équation, un exercice et une figure. On notera l'utilisation de doubles commentaires afin d'indiquer un texte à afficher, et des balises spéciales pour signaler le début et la fin d'un exercice.

Le script Matlab doit être compilé à l'aide d'outils développés spécialement pour les *Numerical Tours*. Ces outils forment un préprocesseur qui utilise la commande `publish` de Matlab afin de générer une page web.

3 Sujet couverts par les *Numerical Tours*

Nous allons maintenant détailler les principales notions mathématiques et numériques couvertes par les TPs des *Numerical Tours*. Toutes les figures sont extraites des *Numerical Tours* et peuvent donc être reproduites en suivant la description étape par étape de chaque TP.

3.1 Traitement en Fourier et en ondelettes

Un signal ou une image est manipulé numériquement comme un vecteur discrétisé $f \in \mathbb{R}^N$ où N est le nombre d'échantillons en 1D ou de pixels en 2D. De nombreuses méthodes de traitement sont effectuées à l'aide d'un changement de base. On considère alors les projections $\{\langle f, \psi_m \rangle\}_{m=0}^{N-1}$ où $\mathcal{B} = \{\psi_m\}_m$ est une base orthogonale de \mathbb{R}^N .

La base de Fourier suppose que les signaux ou les images sont périodiques. Elle est définie en 1D par

$$\forall 0 \leq m < N, \quad \forall 0 \leq x < N, \quad \psi_m(x) = e^{\frac{2i\pi}{N}xm}$$

Sample Numerical Tour

This tour defines and plot a discretized function.

Contents

- [Ploting a function](#)

Ploting a function

One can define the function

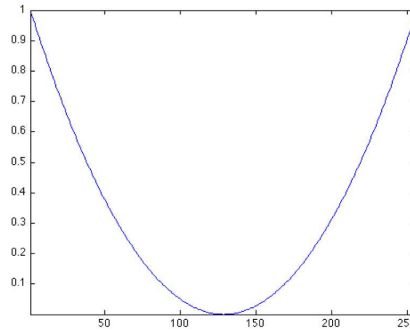
$$f : x \in [-1, 1] \rightarrow x^2 \in \mathbb{R}^+$$

in matlab as a discretized vector.

```
f = linspace(-1,1,256).^2;
```

Exercice 1: (the solution is `ex01.m`) As an exercise, display the function.

```
ex01;
```



```
%% Sample Numerical Tour
% This tour plots a discretized function.

%% Ploting a function

%%
% One can define the function
% \[f : x \in [-1,1] \rightarrow
% x^2 \in \mathbb{R}^+ \]
% in matlab as a discretized vector.

f = linspace(-1,1,256).^2;

% EXO
% As an exercise, display the function.
clf; h = plot(f); axis tight;
% EXO
```

Figure 2: *Gauche* : code Matlab. *Droite* : page web correspondante.

Le paramètre m indexe la fréquence d'un atome. Pour des images, la base de Fourier 2D est définie par produit tensoriel

$$\psi_m(x_1, x_2) = \psi_{m_1}(x_1)\psi_{m_2}(x_2)$$

et $m = (m_1, m_2)$ indexe une fréquence 2D. Les atomes de Fourier sont utiles pour calculer des convolutions qui sont diagonalisées dans la base de Fourier. Ils sont donc au cœur de nombreux traitements linéaires, par exemple pour effectuer un débruitage par lissage ou pour effectuer une régularisation de Tikhonov pour enlever le flou d'une image. Le point capital qui fait en pratique le succès de cette base est que l'ensemble des produits scalaires $\{\langle f, \psi_m \rangle\}_{m=0}^{N-1}$ se calcule en $O(N \log(N))$ opérations à l'aide de l'algorithme de transformée de Fourier rapide (FFT). La base de Fourier est également importante pour le traitement du son, par exemple en calculant une transformée de Fourier à fenêtre (spectrogramme). Ceci permet de déterminer une fréquence instantanée et d'effectuer différents traitements tels que la séparation de sources audio, ou le changement de la hauteur d'un son.

Bien que ces traitements en Fourier soient au centre de bien des méthodes en traitement du signal et des images, cette base n'est pas adaptée pour représenter efficacement des discontinuités. Une base de $L^2(\mathbb{R})$ d'ondelettes $\{\psi_{j,n}\}_{(j,n) \in \mathbb{Z}^2}$ est obtenue en translatant et dilatant une fonction d'ondelette mère ψ afin d'obtenir $\psi_{j,n}(x) = \psi(2^{-j}x - n)$. Une base d'ondelettes 2D est obtenue à l'aide de trois fonctions d'ondelettes mères $\{\psi^k\}_{k \in \{V,H,D\}}$ afin de définir des atomes dans les directions horizontales, verticales et diagonales. Ces atomes sont également des translations et dilatations $\psi_{j,n}^k(x) = \psi^k(2^{-j}x - n)$ où $2^j(n_1, n_2)$ est la position 2D des ondelettes. Un choix judicieux de la fonction ψ permet de définir une base orthogonale $\{\psi_m\}_{m=(j,n,k)}$ de $L^2(\mathbb{R}^2)$ en considérant $j \in \mathbb{Z}, n \in \mathbb{Z}^2, k \in \{V, H, D\}$. Cette base est définie à l'aide d'une cascade de filtrage discrets, voir [22]. Cette construction s'étend de façon naturelle à des signaux et images discrétisés en appliquant la cascade de filtres à un vecteur $f \in \mathbb{R}^N$. Ceci donne naissance à une base orthogonale discrète $\{\psi_m\}_{m=(j,n,k)}$ pour $0 \leq n_1, n_2 < 2^{-j}, 1/N \leq 2^j < 1, k \in \{V, H, D\}$. L'ensemble des produits scalaires $\{\langle f, \psi_m \rangle\}_{m=0}^{N-1}$ est calculé en $O(N)$ opérations à l'aide de la transformée en ondelettes rapide, voir la section 4.1.

Plusieurs TPs sont consacrés à l'implémentation de la transformée en ondelettes en 1D, 2D et 3D. Certains TPs s'intéressent à l'application de la transformée de Fourier à l'analyse de sons.

3.2 Approximation et compression

L'opération élémentaire la plus importante appliquée à un signal $f \in \mathbb{R}^N$ est le calcul de la meilleure approximation non-linéaire avec M coefficients dans une base $\mathcal{B} = \{\psi_m\}_m$. Cette opération peut être considérée comme un prototype pour des opérations plus complexes comme le débruitage, la compression ou l'inversion de problèmes inverses. Elle est définie par un seuillage des coefficients $\{\langle f, \psi_m \rangle\}_m$

$$f_M = H_T(f, \mathcal{B}) = \sum_{|\langle f, \psi_m \rangle| > T} \langle f, \psi_m \rangle \psi_m. \quad (1)$$

Le nombre de coefficients non nuls conservé est $M = |\{m \mid |\langle f, \psi_m \rangle| > T\}|$. Le signal f_M est reconstruit à partir des coefficients seuillés à l'aide d'un algorithme rapide de reconstruction qui a la même complexité que l'algorithme qui calcule les coefficients. La décroissance de l'erreur d'approximation $\|f - f_M\|^2$ reflète l'efficacité de la base pour des applications comme la compression, le débruitage et la régularisation de problèmes inverses. On peut montrer que l'erreur d'approximation dans la base de Fourier d'une image régulière de classe C^α satisfait $\|f - f_M\|^2 = O(M^{-\alpha})$, alors que cette décroissance n'est que de l'ordre de $\|f - f_M\|^2 = O(M^{-1/2})$ pour une image régulière par morceaux avec des discontinuités. La situation est meilleure pour l'approximation en ondelettes d'une image régulière par morceaux avec des contours de discontinuités de longueur finie. Pour ce type d'image, on peut montrer que l'erreur décroît comme $\|f - f_M\|^2 = O(M^{-1})$, ce qui est optimal pour la classe des fonctions à variations bornées, voir [22]. Il existe d'autres familles de bases telles que les curvelets [6] et les bandlets [23] qui améliorent cette vitesse de décroissance pour les images « cartoon », dont les contours sont des courbes régulières.

De nombreux algorithmes de compression utilisent la méthode dite de codage par transformée, qui est proche de l'approximation non linéaire (1). Le seuillage est remplacé par une quantification entière des coefficients

$$q_m = Q_T(\langle f, \psi_m \rangle) \in \mathbb{Z} \quad \text{où} \quad Q_T(x) = \text{sign}(x) \left\lfloor \frac{|x|}{T} \right\rfloor. \quad (2)$$

Les coefficients quantifiés $\{q_m\}_m$ sont ensuite transformés sous la forme d'une succession de 0 et de 1 à l'aide d'un codeur entropique. Pour un $T > 0$ donné, ce flot binaire est court (et mène à une bonne compression) si les coefficients du signal dans la base sont parcimonieux, c'est à dire que la plus part des q_m sont nuls. La décompression à partir du flot binaire s'obtient en inversant le codage entropique pour retrouver les $\{q_m\}_m$ (un codeur entropique est bijectif) puis en reconstruisant un signal approché à partir des coefficients entiers

$$f_R = \sum_m T \text{sign}(q_m) \left(|q_m| + \frac{1}{2} \right) \psi_m.$$

The processus de quantification et dé-quantification produit une erreur $\|f - f_R\|$ dont on peut montrer qu'elle est proche de l'erreur d'approximation non-linéaire $\|f - f_M\|$ obtenue pour le même seuil T . Les meilleurs compresseurs à l'heure actuelle, tel JPEG-2000, utilisent des codages entropiques avancés afin de prendre en compte les dépendances statistiques locales entre des coefficients d'ondelettes voisins, voir [22].

La figure 3 montre un exemple de compression à l'aide d'un codage entropique de coefficients d'ondelettes quantifiés.

Plusieurs TPs des *Numerical Tours* détaillent l'utilisation de bases orthogonales telles les base de Fourier, cosinus local et ondelettes pour effectuer de l'approximation de signaux et d'images. Certains TPs étudient le codage binaire entropique à l'aide d'arbres de Huffman et du codage arithmétique, et appliquent ces méthodes au codage de signaux, images et vidéos.

3.3 Débruitage

Les appareils d'acquisition détériorent les valeurs échantillonnées par de petites fluctuations, dues à la nature physique du processus de mesure. La modélisation mathématique de ce bruit

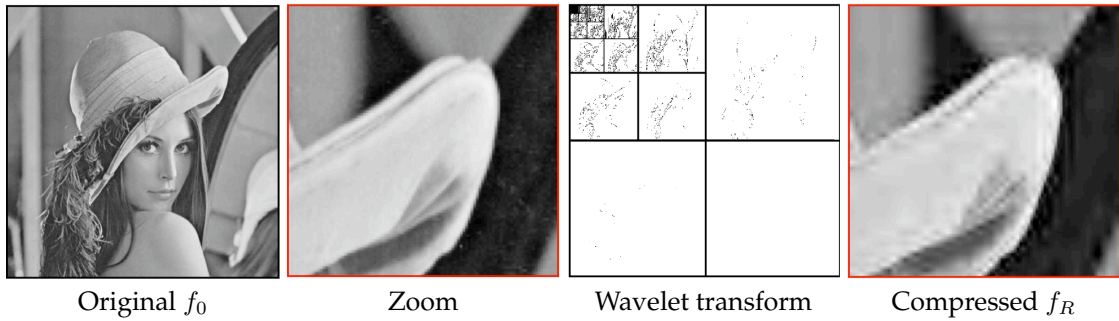


Figure 3: Compression d'image à l'aide du codage des coefficients d'ondelettes.

nécessite l'utilisation de distributions aléatoires, telles que le bruit additif Gaussien, le bruit impulsif de Poisson, ou des bruits multiplicatifs.

Le modèle le plus simple est celui du bruit blanc Gaussien, où les observations sont supposées satisfaire $y = f_0 + w \in \mathbb{R}^N$ où w est un bruit blanc Gaussien de variance σ^2 (ses entrées sont identiques indépendamment distribuées selon une variable Gaussienne). Un débruiteur calcule une estimée f^* du signal ou de l'image inconnu f_0 à partir uniquement des observations y . L'étude théorique de la performance d'un débruiteur borne le risque moyen $E_w(\|f_0 - f^*\|)$, où l'espérance est calculée par rapport à w . Ce risque ne peut être calculé de façon analytique, et les *Numerical Tours* proposent de calculer ce risque de façon numérique à partir de cas tests $f_0 + w$ où l'on suppose connaître f_0 .

Un débruiteur linéaire réalise un lissage $f^* = y * h$ à l'aide d'une convolution par un filtre passe bas h . Ce filtrage enlève le contenu haute fréquence, et donc du bruit. Malheureusement, un tel lissage détruit aussi une partie des contours des images, de telle sorte qu'il n'est pas très efficace sur des images naturelles. Des débruiteurs plus efficaces sont obtenus par seuillage dans une base d'ondelettes, comme proposés initialement par Donoho et Johnstone [15]. Ils calculent $f^* = H_T(y, \mathcal{B})$ où l'opérateur de seuillage est défini en (1). Un tel seuillage est capable de restaurer de façon précise les contours car les ondelettes sont plus efficaces que la base de Fourier pour capturer les singularités. La valeur théorique du seuil est $T = \sqrt{2 \log(N)}\sigma$, ce qui assure que le risque moyen $E_w(\|f_0 - f^*\|)$ décroît rapidement vers 0 lorsque le niveau du bruit σ décroît. Cette décroissance est d'autant plus rapide (et le débruiteur efficace) que la base \mathcal{B} est capable de bien approcher le signal d'origine f_0 , ce qui signifie que $\|f_0 - f_{0,M}\|$ décroît rapidement lorsque M augmente, où $f_{0,M}$ est l'approximation non-linéaire de f_0 avec M coefficients, voir [15, 22]. Empiriquement, on peut constater que la valeur $T = 3\sigma$ marche bien pour des images naturelles. La figure 4 montre un exemple de débruitage en ondelettes.

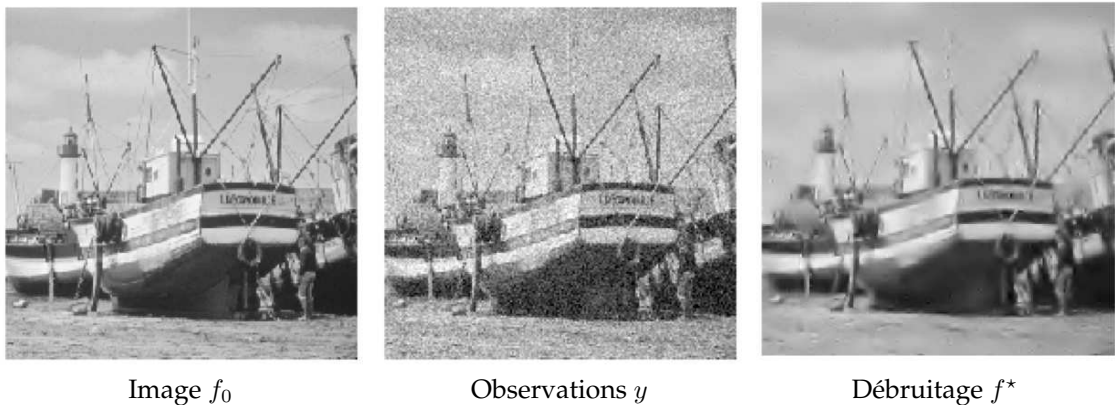


Figure 4: Exemple de débruitage à l'aide d'un seuillage en ondelettes.

Ce débruitage dans une base orthogonale d'ondelettes est nettement amélioré en ajoutant de l'invariance par translation. Ceci peut être obtenu en débruitant des versions translattées de

l'image ou du signal, puis en moyennant les résultats obtenus. De nombreuses autres améliorations sont également possibles, comme par exemple le seuillage de groupes de coefficients afin de prendre en compte les dépendances statistiques entre les coefficients, voir [5].

Une autre classe de débruiteurs calcule l'image débruitée par la minimisation d'un problème variationnel du type

$$f^* = \operatorname{argmin}_{f \in \mathbb{R}^N} \frac{1}{2} \|y - f\|^2 + \lambda J(f), \quad (3)$$

où $J(f)$ est une fonctionnelle convexe de régularisation. Accroître la valeur de $\lambda > 0$ amplifie la force du débruitage. Un a priori de Sobolev pour les images est défini par

$$J(f) = \sum_x \|\nabla f(x)\|^2, \quad (4)$$

où $\nabla f(x)$ est une approximation par différences finie du gradient de f au pixel x . La minimisation de J à travers (3) est équivalente au calcul d'un filtre passe base qui rend flou les contours. Un a priori populaire pour les images est la variation totale

$$J(f) = \sum_x \|\nabla f(x)\|, \quad (5)$$

qui permet de mieux respecter les contours [29].

Les TPs des *Numerical Tours* font un tour d'horizon des méthodes de débruitage standards, ce qui inclus les méthodes à base d'ondelettes et les méthodes variationnelles. Certains TPs s'intéressent à des variations autour de ces méthodes de base, comme le seuillage par blocs. D'autres TPs détaillent les algorithmes permettant de calculer la solution du problème d'optimisation (3). Enfin, d'autres TPs abordent des méthodes récentes parmi l'état de l'art, comme par exemple les filtrages non-locaux [4] et l'apprentissage de dictionnaires [18].

3.4 Problèmes inverses

Les systèmes d'acquisition introduisent, en plus du bruit, une perte d'information par rapport à un signal ou une image haute résolution idéale $f_0 \in \mathbb{R}^N$. Une acquisition est ainsi modélisée par $y = \Phi f_0 + w$ où $\Phi : \mathbb{R}^N \rightarrow \mathbb{R}^Q$ ne récupère que Q mesures basse résolution. Ceci permet de prendre en compte par exemple le flou de caméra qui enlève les hautes fréquences ou les pixels manquant à cause de capteurs endommagés.

Inverser directement l'opérateur Φ est impossible car $Q \leq N$ et Φ est mal posé. Les méthodes de régularisation ajoutent des contraintes sur la solution f^* cherchée en imposant qu'elle est la solution d'un problème d'optimisation

$$f^* \in \operatorname{argmin}_{f \in \mathbb{R}^N} \frac{1}{2} \|y - \Phi f\|^2 + \lambda J(f). \quad (6)$$

La régularisation avec l'a priori de Sobolev (4) évite l'explosion du bruit par l'inversion de l'opérateur mais lisse les discontinuités. L'a priori de variation totale (5) est efficace pour la déconvolution, lorsque Φ est un filtre passe bas, car il est capable de restaurer des contours nets. Des schémas numériques pour l'optimisation non-lisse permettent de minimiser (6) lorsque J est la variation totale, voir par exemple [9] et la section 4.2.

La régularisation parcimonieuse fait l'hypothèse que le signal ou l'image f_0 à retrouver est parcimonieux (ou creux) dans une base orthogonale $\mathcal{B} = \{\psi_m\}_m$. Ceci signifie que la base \mathcal{B} est efficace pour l'approximation non-linéaire de f . Cette contrainte de parcimonie est mesurée à l'aide de la norme ℓ^1 des coefficients, qui définit un a priori convexe

$$J(f) = \sum_m |\langle f, \psi_m \rangle|.$$

La minimisation (6) se calcule à l'aide d'algorithmes d'optimisation non-lisse comme par exemple l'algorithme de seuillage itératif [13], qui fait parti de la classe des algorithmes proximaux [12]. On

peut également considérer un dictionnaire redondant \mathcal{B} , et chercher directement les coefficients a_m du signal ou de l'image $f^* = \sum_m a_m^* \psi_m$ en résolvant

$$a^* \in \underset{a}{\operatorname{argmin}} \frac{1}{2} \|y - \Phi \sum_m a_m \psi_m\|^2 + \lambda \sum_m |a_m|. \quad (7)$$

Cette minimisation est équivalente à la méthode de basis pursuit [11] pour calculer une approximation parcimonieuse de y dans le dictionnaire $\{\Phi \psi_m\}_m$.

La figure 5 montre un exemple d'inpainting (aussi appelé « désocclusion » [24]), qui correspond à l'inversion d'un opérateur de masquage

$$(\Phi f)(x) = \begin{cases} f(x) & \text{si } x \notin \Omega, \\ 0 & \text{sinon,} \end{cases}$$

où le masque Ω couvre 70% des pixels. L'image est reconstruite en résolvant (7) dans un dictionnaire d'ondelettes invariantes par translation.

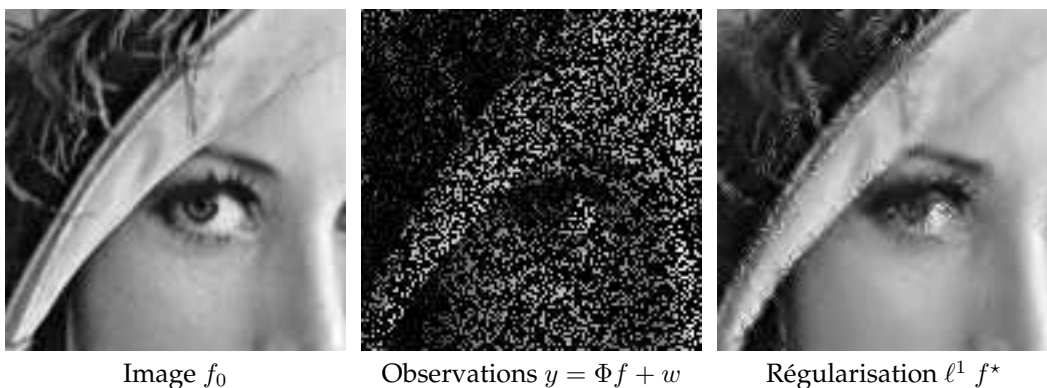


Figure 5: Exemple d'inpainting à l'aide de la régularisation ℓ^1 en ondelettes.

L'échantillonnage compressé est une nouvelle méthode d'échantillonnage proposée conjointement par [14, 7]. L'échantillonnage $y = \Phi f_0 + w$ d'un signal haute résolution f_0 est obtenu en projetant ce signal sur un petit nombre Q de vecteurs φ_i aléatoires, de telle sorte que $\Phi f_0 = \langle f_0, \varphi_i \rangle$ (les φ_i sont les lignes de la matrice Φ). Si le signal f_0 est suffisamment parcimonieux, la théorie de l'échantillonnage compressé montre que l'on peut retrouver un signal f^* en résolvant (6) qui est proche du signal haute résolution f_0 , voir [14, 7].

Les TP de *Numerical Tours* étudient de nombreux problèmes inverses, tels que la déconvolution [16], l'inpainting [24, 2, 19], la tomographie [22] et l'échantillonnage compressé [14, 7]. A chaque fois, plusieurs schémas de régularisation sont proposés, tels que Sobolev, TV, ℓ^1 dans une base orthogonale et dans un dictionnaire redondant. Ces différents scénarios permettent d'étudier différents algorithmes d'optimisation pour résoudre (6).

3.5 Traitement des surfaces

La plus part des méthodes de traitement des signaux et des images s'étendent aux surfaces. Le schéma de discrétisation le plus simple est celui d'une triangulation de la surface. Ceci correspond à un échantillonnage non uniforme $\mathcal{V} = \{x_n\}_{n=0}^{N-1} \subset \mathbb{R}^d$, où $d = 2$ pour les maillages plans et $d = 3$ pour les maillages surfaciques. La connectivité du maillage est définie par un ensemble de faces $\mathcal{F} \subset \{0, \dots, N-1\}^3$, de telle sorte que le maillage est entièrement décrit par une matrice de taille $3 \times N$ pour représenter \mathcal{V} et une matrice de taille $3 \times |\mathcal{F}|$ pour représenter \mathcal{F} . Un couple d'indices $(i, j) \in \mathcal{E}$ est une arête si il appartient à une face. Une triangulation est topologiquement valide si chaque arête est incident à soit deux faces soit une face (pour les arêtes du bord de la surface).

Un signal $f \in \mathbb{R}^N$ défini sur le maillage assigne à chaque sommet indexé par i une valeur $f_i \in \mathbb{R}$. On peut alors appliquer des traitements à ce vecteur pour le débruiter ou le compresser.

Dans certaines situations, on peut vouloir modifier la surface elle-même, dans quel cas les sommets $\{x_i = (x_i^1, \dots, x_i^d)\}_i$ définissent d signaux différents ($x^1 \in \mathbb{R}^N, \dots, x^d \in \mathbb{R}^N$), qui sont les coordonnées des points.

Les opérateurs de traitement les plus simples sont linéaires, et peuvent se représenter à l'aide d'une matrice $W \in \mathbb{R}^{N \times N}$. On utilise le plus souvent des matrices creuses, de telle sorte que $W_{i,j} \neq 0$ seulement si $(i, j) \in \mathcal{E}$. Si $W_{i,j} \geq 0$ et $\sum_j W_{i,j} = 1$, alors Wf est un filtrage passe bas qui enlève du bruit dans le signal. Itérer l'opérateur W afin de calculer $W^k f$ pour $k > 0$ effectue un débruitage progressif du signal ou de la surface, voir la figure 6. Un tel opérateur peut aussi être utilisé pour définir un a priori de Sobolev sur un maillage comme

$$J(f) = \sum_{i,j} W_{i,j} |f_i - f_j|^2.$$

Cet énergie peut être utilisé pour réaliser du débruitage en minimisant (6), ou pour interpoler une déformation définie seulement en certains sommets en résolvant un problème similaire à (3).

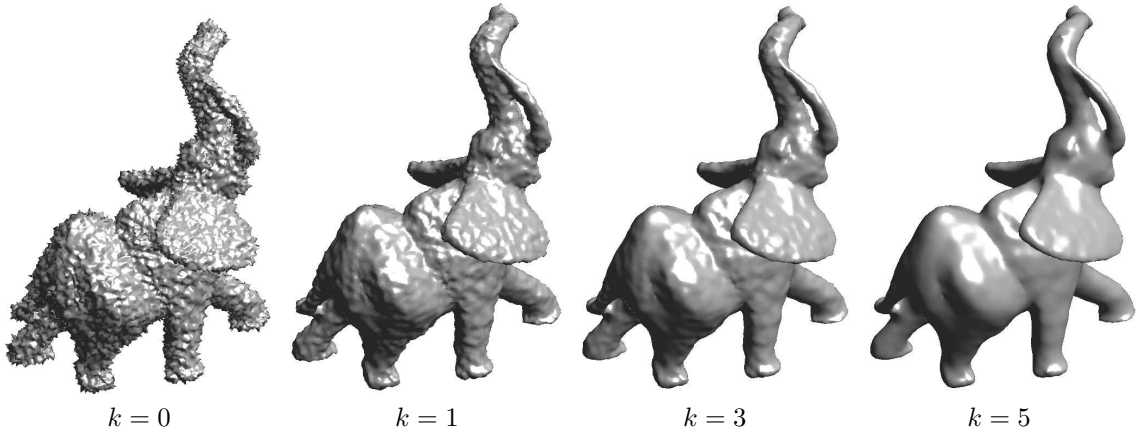


Figure 6: Exemple de débruitage des positions $(x_i^1, x_i^2, x_i^3) \in \mathbb{R}^3$ d'un maillage 3D par filtrage ($W^k x^1, W^k x^2, W^k x^3$) pour différentes valeurs de k .

Une paramétrisation 2D d'un maillage 3D assigne à chaque sommet x_i une position $(f_i^0, f_i^1) \in \mathbb{R}^2$. Les méthodes classiques calculent une paramétrisation lisse en minimisant $J(f^0) + J(f^1)$ tout en fixant les positions des sommets sur le bord de la surface le long d'une courbe convexe, voir [3].

La base de Fourier est étendue à un maillage en considérant les vecteurs singuliers d'un opérateur de filtrage W . Ces vecteurs singuliers sont ordonnés par leurs fréquences, et sont utilisés pour effectuer de la compression de maillages comme décrit en (2), voir [20]. Les bases d'ondelettes peuvent aussi être étendues aux maillages avec des applications à la compression et au débruitage [30].

Les *Numerical Tours* détaillent les extensions de plusieurs problèmes de traitement d'image à des maillages, comme le débruitage, l'interpolation et la compression. Ils étudient aussi la déformation de maillages, l'aplatissement et la paramétrisation.

3.6 Traitement des courbes

Un problème central en vision par ordinateur est la segmentation d'images, qui consiste à chercher une courbe $t \in [0, 1] \mapsto \gamma(t) \in [0, 1]^2$ suivant les bords des objets dans une image f . La segmentation est définie en minimisant une énergie non-convexe, comportant un terme de régularisation (par exemple la longueur euclidienne de la courbe) et un terme d'attache au données calculé à partir de f .

Une classe populaire d'attache au données, introduite initialement par [21], consiste à chercher

une courbe de longueur pondérée minimum, voir [8]

$$L(\gamma) = \int_0^1 P(\gamma(t)) \|\gamma'(t)\| dt \quad \text{où} \quad P(x) = \frac{1}{\varepsilon^2 + \|\nabla f(x)\|^2} \quad (8)$$

Le potentiel $P(x)$ est un détecteur de contour qui attire γ vers les zones de fort gradient de f . Minimiser $L(\gamma)$ correspond au calcul d'une courbe de longueur géodésique localement minimale [8]. Une autre classe de méthodes utilise une attache au données exploitant les valeurs de f à l'intérieur de la région entourée par γ (et non pas le long de γ comme (8)), ce qui correspond à chercher une approximation régulière par morceaux de la fonction f , voir [25] et [10].

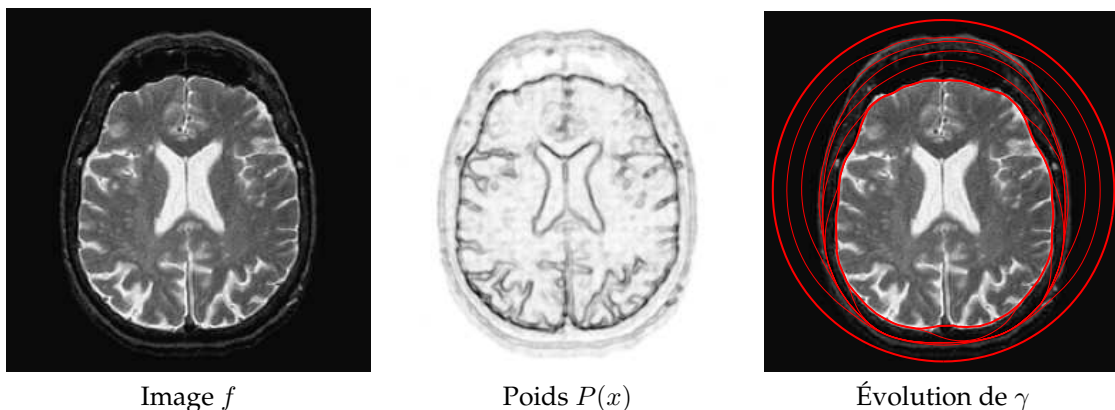


Figure 7: Exemple d'évolution d'un contour actif géodésique pour la segmentation d'une image médicale. La courbe initiale est un cercle entourant l'objet d'intérêt.

La segmentation est obtenue en minimisant (8), par exemple en imposant des conditions au bord de sorte que la courbe soit fermée, c'est-à-dire $\gamma(0) = \gamma(1)$. On peut calculer une descente de gradient de L par rapport à γ , ce qui correspond à discrétiser en temps une EDP d'évolution convergeant vers un minimum local de L . On peut réaliser cette descente de gradient en supposant que γ est donnée de façon explicite via une courbe paramétrée qui est discrétisée. On peut également supposer que cette courbe est donnée comme l'ensemble de niveau 0 d'une fonction φ , c'est-à-dire que

$$\{\gamma(t) \mid t \in [0, 1]\} = \{x \in \mathbb{R}^2 \mid \varphi(x) = 0\},$$

cette fonction φ étant discrétisée sur une grille carrée régulière, comme une image. Les méthodes d'ensembles de niveaux, bien que plus lourde numériquement, ont l'avantage de permettre de gérer des changements topologiques pendant l'évolution, comme par exemple la séparation en deux d'une courbe [27]. La figure 7 montre un exemple d'évolution d'une courbe fermée.

Afin de détecter des structures filamenteuses dans les images (comme par exemple des vaisseaux sanguins dans une image médicale), on peut utiliser une courbe ouverte et imposer des conditions aux bords $\gamma(0) = x_0, \gamma(1) = x_1$. Il est alors possible de calculer un minimum global de L , qui correspond à la courbe géodésique de longueur minimale joignant x_0 à x_1 . On calcule tout d'abord la distance géodésique entre x_0 et tous les points de l'image par l'algorithme de Fast Marching [31] dont la complexité est $O(N \log(N))$ pour une image de N pixels. La courbe minimale γ est ensuite extraite par une descente de gradient de cette carte de distance, initialisée à la position x_1 .

Les TP des *Numerical Tours* explorent le problème de la segmentation d'image à la fois à l'aide de représentations explicites et implicites des courbes, pour différentes énergies. Ils contiennent aussi une étude détaillée des calculs de courbes géodésiques sur des images 2D, 3D et des surfaces triangulées. Le calcul de distances géodésiques est aussi utilisé afin de résoudre le problème de l'échantillonnage de surfaces et pour la reconnaissance de formes [28].

4 Exemples de Numerical Tours

Cette section détaille deux TPs des *Numerical Tours*. Elle ne décrit pas les détails d'implémentation, mais donne un bref aperçu des idées sous-jacentes à deux méthodes importantes en traitement du signal et des images. Il est conseillé d'aller voir en ligne les TPs complets afin d'obtenir des explications détaillées.

4.1 Exemple #1 – Calcul d'une transformée en ondelettes

La transformée en ondelettes d'un signal 1D $f \in \mathbb{R}^N$ correspond au calcul des produits scalaires $d_j[n] = \langle f, \psi_{j,n} \rangle$ avec des atomes d'ondelettes discrets $\psi_{j,n} \in \mathbb{R}^N$ indexés par leur échelle $2^j \leq 1$ et leur position n . Le seul paramètre d'une transformée en ondelettes est un filtre passe bas h . Le filtre passe haut associé g est défini par $g[i] = (-1)^{i+1}h[1-i]$. Des listes de filtres h valides peuvent être trouvés par exemple dans [22].

La transformée en ondelettes rapide calcule tous les coefficients en ondelettes $\{d_j\}_{j=J+1}^0$ et un résidu basse fréquence $a_0 \in \mathbb{R}$. L'échelle la plus fine est $J = -\log_2(N)$. La transformée utilise des vecteurs basse fréquence intermédiaires $a_j \in \mathbb{R}^{2^{-j}}$. Le vecteur initial est $a_J = f \in \mathbb{R}^N$. Les coefficients en ondelettes d_{j+1} ainsi que le résidu a_{j+1} suivant sont calculés à l'aide de convolutions sous-échantillonnées, pour $j = J, \dots, -1$

$$a_{j+1} = (a_j \star h) \downarrow 2 \quad \text{et} \quad d_{j+1} = (d_j \star g) \downarrow 2 \quad (9)$$

où $\downarrow 2$ est l'opérateur de sous-échantillonnage défini par $(a \downarrow 2)[k] = a[2k]$.

La figure 8 montre le code Matlab/Scilab pour réaliser le calcul de ces convolutions sous-échantillonnées. Cette figure montre également les coefficients en ondelette d_j obtenus. On pourra noter que le code réalise les calculs « sur place », ce qui signifie qu'un seul et unique vecteur (noté fw) permet de stocker à la fois les coefficients déjà calculés ainsi que le vecteur a_j courant.

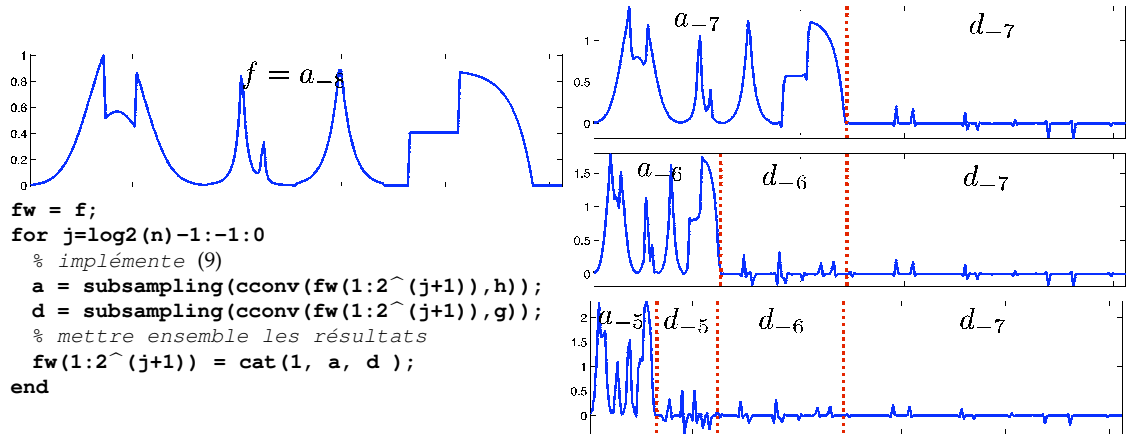


Figure 8: Gauche : algorithme de décomposition en ondelettes 1D. Droite : évolution de la variable fw pendant les itérations.

Afin de réaliser une approximation non-linéaire, les coefficients en ondelettes sont modifiés à l'aide d'un seuillage, comme indiqué en (1). Cette modification donne

$$\forall j, \forall n, \quad d_j[n] \leftarrow \begin{cases} d_j[n] & \text{if } |d_j[n]| > T, \\ 0 & \text{sinon.} \end{cases} \quad (10)$$

A partir de ces coefficients modifiés, on effectue une transformée en ondelettes inverse en inversant les étapes (11) pour $j = -1, \dots, J$

$$a_j = (a_{j+1} \star \tilde{h}) \uparrow 2 + (d_{j+1} \star \tilde{g}) \uparrow 2 \quad (11)$$

où $\tilde{h}[i] = h[-i]$ et $b = \uparrow 2a$ est l'opérateur de sur-échantillonnage défini par $b[2i] = a[i]$ et $b[2i + 1] = 0$. Le signal approché est au final obtenu comme $f_M = a_J \in \mathbb{R}^N$, où M est le nombre de coefficients non nuls. La figure 9 montre des exemples d'approximations non-linéaires en ondelettes pour différentes valeurs de M .

```
% implémente (10)
fw = (abs(fw)>T).*fw;
f1 = fw;
for j=0:log2(n)-1
    a = f1(1:2^j);
    d = f1(2^j+1:2^(j+1));
    % implémente (11)
    a = cconv(upsampling(a,1),h(end:-1:1),1);
    d = cconv(upsampling(d,1),g(end:-1:1),1);
    f1(1:2^(j+1)) = a + d;
end
```

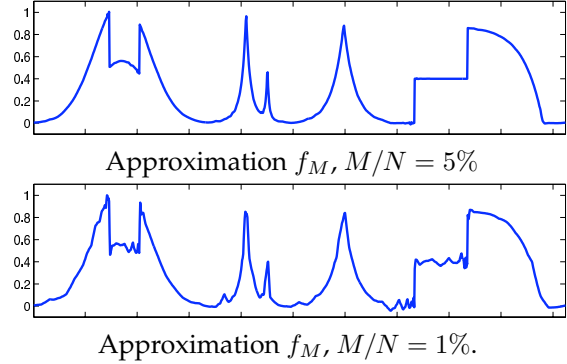


Figure 9: Gauche : algorithme d'approximation en ondelettes à l'aide d'un seuillage dur et d'une transformée en ondelettes inverse. Droite : approximations f_M calculées pour deux valeurs différentes du seuil T .

4.2 Exemple #2 – Déconvolution par minimisation de la variation totale

Le problème de déconvolution correspond à un problème inverse $y = \Phi f_0 + w$ où $\Phi f = f \star h$ est un opérateur de convolution par un filtre passe bas h supposé connu. Afin de restaurer avec précision les contours, il est possible de résoudre (6) en utilisant l'a priori de variation totale $J(f) = \sum_x \|\nabla f(x)\|$ déjà introduit en (5). Malheureusement, la fonctionnelle $f \mapsto J(f)$ n'est pas différentiable, ce qui rend l'optimisation difficile. Pour simplifier la procédure d'optimisation, il est possible de considérer une version lissée de la variation totale

$$J_\varepsilon(f) = \sum_x \sqrt{\varepsilon^2 + \|\nabla f(x)\|^2}.$$

Lorsque ε tend vers zero, $J_\varepsilon(f)$ approche J mais le problème d'optimisation devient plus difficile à résoudre.

Puisque $f \mapsto J_\varepsilon(f)$ est une fonctionnelle lisse, il est possible d'utiliser une descente de gradient pour minimiser (6). En démarrant par $f^{(0)} = y$, on itère

$$f^{(k+1)} = f^{(k)} - \tau \left(\Phi^*(\Phi f^{(k)} - y) + \lambda \text{Grad}(J_\varepsilon)(f^{(k)}) \right). \quad (12)$$

Le gradient de la fonctionnelle J_ε est

$$\text{Grad}(J_\varepsilon)(f) = -\text{div} \left(\frac{\nabla f}{\sqrt{\varepsilon^2 + \|\nabla f(x)\|^2}} \right), \quad (13)$$

où $\text{div} = -\nabla^*$ est une approximation par différences finies de la divergence de champs de vecteurs. Les itérés $f^{(k)}$ convergent vers un minimiseur f^* de (6) si le pas de descente est assez petit par rapport à ε ,

$$\tau < \frac{2}{\|\Phi^*\Phi\| + \lambda \|\text{div} \circ \nabla\|/\varepsilon}.$$

Ici $\|\Phi^*\Phi\| = \max_\omega |\hat{h}(\omega)|^2$ est le carré de la plus grande valeur singulière de Φ et $\|\text{div} \circ \nabla\| = 8$ pour une approximation standard par différence finies du gradient. La figure 10 montre les résultats obtenus pour deux valeurs différentes de λ .

```

% raccourci pour la convolution
Phi = @(x) real(iff22(ff22(x).*ff22(h)));
% initialization
f = y;
for i=1:niter
    % calcule du gradient (13)
    Gr = grad(f);
    d = sqrt(epsilon^2 + sum3(Gr.^2,3));
    G = -div(Gr./repmat(d,[1 1 2]));
    % étape de descente (12)
    f = f - tau*(Phi(Phi(f))-y) + lambda*G;
end

```

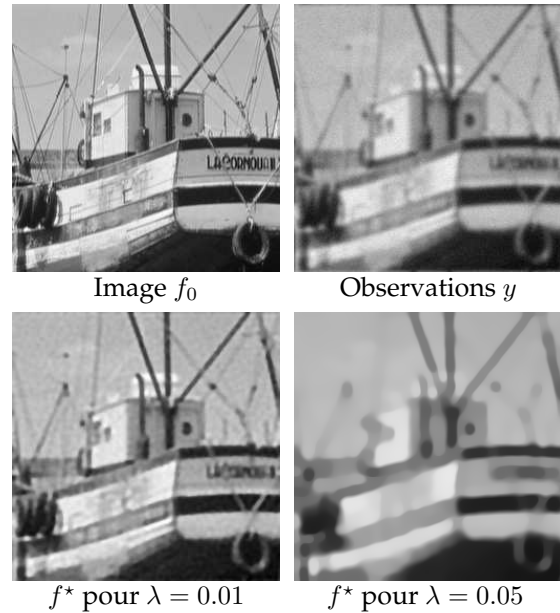


Figure 10: Déconvolution par minimisation de la variation totale.

Conclusion

Nous avons présenté les *Numerical Tour of Signal Processing*, une large collection de TPs en Matlab et Scilab qui guidera l'utilisateur dans la jungle des avancées récentes en traitement du signal, de l'image et des surfaces. Nous encourageons le lecteur à visiter la page web (www.numerical-tours.com) des *Numerical Tours* pour se faire une idée par lui-même de l'utilité de ces ressources.

References

- [1] M. Barni and F. Perez-Gonzalez. Pushing science into signal processing. *Computing in Science and Engineering*, 22(4):120–119, 2005.
- [2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. Siggraph 2000*, pages 417–424, 2000.
- [3] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. *Polygon Mesh Processing*. AK Peters, 2010.
- [4] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *SIAM Multiscale Modeling and Simulation*, 4(2):490–530, 2005.
- [5] T. Cai. Adaptive wavelet estimation: a block thresholding and oracle inequality approach. *Ann. Statist.*, 27(3):898–924, 1999.
- [6] E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Commun. on Pure and Appl. Math.*, 57(2):219–266, 2004.
- [7] E. Candès, J. Romberg, and T. Tao. Signal recovery from incomplete and inaccurate measurements. *Commun. on Pure and Appl. Math.*, 59(8):1207–1223, 2005.
- [8] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, February 1997.
- [9] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, 2004.

- [10] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277, February 2001.
- [11] S. S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.
- [12] P. L. Combettes and V. R. Wajs. Gsignal recovery by proximal forward-backward splitting. *SIAM Multiscale Modeling and Simulation*, 4(4), 2005.
- [13] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. on Pure and Appl. Math.*, 57:1413–1541, 2004.
- [14] D. Donoho. Compressed sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, 2006.
- [15] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, Dec 1994.
- [16] D. L. Donoho. Superresolution via sparsity constraints. *SIAM Journal on Mathematical Analysis*, 23(5):1309–1331, September 1992.
- [17] D. L. Donoho, A. Maleki, I. Ur Rahman, M. Shahram, and V. Stodden. Reproducible research in computational harmonic analysis. *Computing in Science and Engineering*, 11(1):8–18, 2009.
- [18] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Proc.*, 15(12):3736–3745, 2006.
- [19] M. Elad, J.-L. Starck, D. Donoho, and P. Querre. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Journal on Applied and Computational Harmonic Analysis*, 19:340–358, 2005.
- [20] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [21] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.
- [22] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.
- [23] S. Mallat and G. Peyré. Orthogonal bandlet bases for geometric images approximation. *Commun. on Pure and Appl. Math.*, 61(9):1173–1212, 2008.
- [24] S. Masnou. Disocclusion: a variational approach using level lines. *IEEE Trans. Image Processing*, 11(2):68–76, February 2002.
- [25] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and variational problems. *Communications on Pure and Applied Mathematics*, XLII(5):577–685, 1988.
- [26] G. C. Orsak and D. M. Etter. Collaborative SP education using the Internet and MATLAB. *IEEE Signal Processing Magazine*, 12(6):23–32, November 1995.
- [27] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [28] G. Peyré, M. Pechaud, R. Keriven, and L.D. Cohen. Geodesic methods in computer vision and graphics. *Foundations and Trends in Computer Graphics and Vision*, 5(3-4):197–397, 2010.
- [29] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.
- [30] P. Schröder and W. Sweldens. Spherical Wavelets: Efficiently Representing Functions on the Sphere. In *Proc. of SIGGRAPH 95*, pages 161–172, 1995.

- [31] J. Sethian. *Level Sets Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.
- [32] P. Vandewalle, J. Kovacevic, and M. Vetterli. Reproducible research in signal processing. *IEEE Signal Processing Magazine*, 26(3):37–47, May 2009.