Contributions in topology optimization: extension of the adjoint method and applications in image processing

Stanislas Larnier

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse III - Paul Sabatier
Spécialité: Mathématiques appliquées

Présentée et soutenue par **Stanislas Larnier**
le 2 décembre 2011

Contributions in topology optimization: extension of the adjoint method and applications in image processing

(Contributions en optimisation topologique : extension de la méthode adjointe et applications au traitement d'images)

JURY

Mohamed Masmoudi	Directeur de thèse	*Université Toulouse III*
Jérôme Fehrenbach	Codirecteur de thèse	*Université Toulouse III*
Yves Capdeboscq	Rapporteur	*Université d'Oxford*
François Jouve	Rapporteur	*Université Paris VII*
Samuel Amstutz	Examinateur	*Université d'Avignon*
Didier Auroux	Examinateur	*Université de Nice*
Marc Bonnet	Examinateur	*ENSTA ParisTech*

École doctorale	:	Mathématiques Informatique Télécommunications (MITT)
Unité de recherche	:	Institut de Mathématiques de Toulouse - UMR 5219
Directeurs de thèse	:	Mohamed Masmoudi, Jérôme Fehrenbach

“The most exciting phrase to hear in science, the one that heralds new discoveries, is not ’Eureka!’ but ’That’s funny ...’.”
- Isaac Asimov (1920-1992)

This work is dedicated to Usain and Pepito.

Contents

List of Figures

List of Tables

Acknowledgment

First of all, I would like to thank my supervisors Mohamed Masmoudi and Jérôme Fehrenbach. They gave me the opportunity to do this work. They allowed me to improve my knowledge of topological optimization and thanks to them, I have realized some biological and medical applications, areas of particular interest for me.

I am honoured that Yves Capdeboscq and François Jouve accepted to review my thesis, I thank them for the time they devoted and their suggestions on different points of my thesis. Samuel Amstutz, Didier Auroux and Marc Bonnet have accepted to be members of my thesis jury. I thank them for the interest they showed for my work.

I am greatly indebted to some people without whom none of this work would exist. I would like to thank:

- the people who shared my office room: Ritesh Kumar and Yogesh Parte for helping me improving my English and bringing a piece of India, Mathieu Fructus for his advices and help concerning administrative problems of our foreign coworkers, Clément Jourdana for the arrow-words at some lunch breaks, Caï Yongyong for his discretion, Luis Faria for his serenity, Alain Bonnafé for his seriousness and the concert places, Ngoc Ming Dao for his Vietnam and Vietnamese lessons, and last but not least Fabien Monfreda, for his theatrality and his sense of humor.

- Pierre Weiss for his mathematical knowledge, his advice and our running trainings, Jean-Denis Durou for his interest for my work and Roberto Mecca for our collaborative work.

- the librarians and the personnel of the institute: cleaners, computer scientists and secretaries and members in particular Mathieu Causse, Issam Fatnassi and Marcel Mongeau.

- the people with whom I worked as a teaching assistant: my tutrice Marjolaine Puel and my coworkers Étienne Fieux, Michel Fournié, Yann Grisel and Philippe Monnier.

- my students, they were oxygen during my research work.

- the many people on internet who share their codes and help diffusing the knowledge.

- my teachers and my professors since the primary school until the university which have developed my knowledge and grew my interest for mathematics.

- my friends from North and South of France who keep me going and support me, more specifically Marion Gabarrou and her large family, Anaïs Montoya, Marion Lebellego, Jérôme Neuville and Victor Ng.

- my family specially my parents Catherine and Patrick Larnier, thank you for giving me so much love.

Introduction

Nowadays, topology optimization has been extensively studied in structural optimization which is a major interest in the design of mechanical systems in industry and in inverse problems with the detection of defects or inclusions. In this thesis, I propose to use topological derivatives to new problems in signal and image processing, domains which share a common goal with topology optimization. There exist some applications of the topological gradient method to image processing, I go into them in more depth and propose a general framework suited to several image processing problems. Some drawbacks of topological derivatives exist: they are limited to simple problems, we do not know how to fill holes, ... In order to overcome these difficulties, I develop an extension of the adjoint method. It allows to obtain a new theorical and applied tool in order to obtain better results and explore new fields of applications for the topology optimization. This chapter presents briefly the topology optimization and the main contributions of this thesis.

Topology optimization formulates a design problem as an optimal material distribution problem. The search of an optimal domain is equivalent to finding its characteristic function, it is a 0-1 optimization problem. It can be seen as a generalization of shape optimization because it does not impose restriction on the topology of the optimal shape, for example its number of holes.

Different approaches allow these kind of problems to be tackled and make them differentiable:

- ➢ Relaxation, homogenization,
- ➢ Level set,
- ➢ Topological derivatives.

The homogenization technique [4, 3, 29, 28] is based on relaxed formulations leading to the introduction of some intermediate material or micro-structures. The drawback is precisely that the optimal solution is not a classical design: it is a distribution of composite materials. Then penalization methods must be applied in order to retrieve a feasible shape. This method is mainly restricted to linear elasticity and particular objective functions

The level set method [88, 89, 6, 111] can handle boundary propagation with topological changes. In practice, the level set method can easily remove holes but cannot create new ones. For this reason, in [5], the authors couple the level set method with the

topological gradient method, which is precisely designed for introducing new holes in the optimization process.

This work focuses on the topological derivatives [49, 105, 107, 75, 49, 56, 57, 13, 14, 77]. They are used for problems where explicit expressions have been derived to describe the variation of a cost function with respect to a modification of the topology of a domain.

To present the basic idea, let us consider the case of a circular hole. Let Ω be a variable domain of $\mathbb{R}^2$ and a cost function $j(\Omega) = J(u_\Omega)$ to be minimized, where u_Ω is solution to a given partial differential equation defined over Ω. For a small parameter $\rho > 0$, let $\Omega \backslash B(x_0, \rho)$ be the perturbed domain obtained by making a circular hole of radius ρ around the point x_0. The topological sensitivity analysis provides an asymptotic expansion of $j(\Omega \backslash B(x_0; \rho))$ when ρ tends to zero in the form:

$$j(\Omega \backslash B(x_0; \rho)) = j(\Omega) + f(\rho)g(x_0) + o(f(\rho)).$$

In this expansion, $f(\rho)$ denotes an explicit positive function vanishing to zero, $g(x_0)$ is called the topological gradient. It is usually simple to compute and is obtained using the solution of direct and adjoint problems defined on the initial domain. To minimize the criterion j, one has to create holes at some points x where $g(x)$ is negative.

The topological derivatives have been applied to different kinds of problems:

- ➢ the elasticity case [49],
- ➢ the Poisson equation [56],
- ➢ the Navier-Stokes equation [57],
- ➢ the Helmholtz equation [14].

Most of the time, the topological asymptotic expansion provides only the topological gradient. Currently, some authors like M. Bonnet are interested in asymptotic topological expansions of higher-order [31, 32].

In this work, the topological gradient method is applied to image processing problems. There is a link between topology optimization and image processing, they have a common goal: the partition of a given domain. In topology optimization, from an initial domain, the goal is to look for the optimal design and its complementary. In image processing, a very common problem is to split the image in an edge set and its complementary. For this reason, topology optimization and image processing problems share common mathematical methods coming from the PDE community like variational methods, level set approaches and topological gradient.

In topology optimization, there are some drawbacks of topological gradient approaches:

- ➢ The asymptotic topological expansion is not easy to obtain for complex problems.

- ➢ It needs to be adapted for many particular cases such as the creation of a hole on the boundary of an existing one or on the original boundary of the domain.
- ➢ It is difficult to determine the variation of a cost function when a hole is to be filled.
- ➢ In real applications of topology optimization, a finite perturbation is performed and not an infinitesimal one.

Certain issues arise here, as for example the question of how large the topological change should be. In the present work, more precisely in Chapter 2, I propose an extension of the adjoint method to overcome these problems.

Practical applications like image processing and inverse problem in electroencephalography have been studied. My objective is to propose simple and efficient solutions to different kind of problems by exploring new approaches and tools in topology optimization.

In the image processing part, several methods already exist. The topological gradient method is barely used in this domain and the present work intends to complete the knowledge in this recent research field. In topology optimization, the adjoint method is extended, creating a universal approach able to deal with finite singular perturbations. In electroencephalography, a new approach to perform dipolar source localizations is proposed. This approach attempts to find the dipolar sources without a priori knowledge on them.

Due to its simple definition, the extension of the adjoint method can be easily adapted to any kind of problem. Simplicity is linked to the speed of the codes, all the proposed algorithms are fast or could easily be used with parallel computing. The obtention of acceptable results very quickly is of importance for example in medical imaging, the computing time needs to be negligible to avoid delays in the diagnosis.

During this work, collaborations with biologists and medical doctors allowed me to discover new applications and have greatly motivated theoretical and applied developments. Some projects of this document originate from a meeting or a workshop with biologists and medical doctors. In this work, even if some applications are not performed on real data, they are performed on data created by simulations in close relation with the people whom I have interacted with.

This document is composed of three independent chapters and three appendices. They can be read independently, thought I advice reading Chapter 1 on image processing before the appendices.

Chapter 1 presents some applications of the topological gradient in image processing. A unified theoretical framework is proposed allowing different image reconstruction problems to be considered including restoration, inpainting, demosaicing, segmentation and super-resolution. Furthermore an improvement of existing algorithms is proposed and

new ones are introduced. The performance of our approach is compared with conventional image reconstruction processes. The last section is a study of fractional derivatives applied to image processing with topological gradient information. Because of their nonlocal property, fractional derivatives provide better reconstructions. The work reported in this last section has been carried out in collaboration with Roberto Mecca, a PhD student from Dipartimento di Matematica "G. Castelnuovo" Sapienza, Universitá di Roma in Italia. Numerical results are presented and discussed.

Chapter 2 presents an extension of the adjoint method. Searching for the optimal partitioning of a domain leads to the use of the adjoint method in topological asymptotic expansions to evaluate the influence of a domain perturbation on a cost function. This approach works by restricting to local subproblems containing the perturbation and outperforms the adjoint method by providing approximations of higher order. It is a universal tool, easily adapted to different kinds of real problems, and does not need the fundamental solution of the problem; furthermore our approach allows finite perturbations to be considered and not infinitesimal ones. This chapter presents some applications with topological perturbations, continuous perturbations, mesh perturbations. This proposed approach can also be used to update the solution of singularly perturbed problems.

Chapter 3 is a continuation of my master thesis. The subject originates from a collaboration with the Institut National de la Santé et de la Recherche Médicale (INSERM). It presents a new approach to solve the inverse problem of dipolar source localizations in electroencephalography. The extension of the adjoint approach is used to find the development of different cost functions. A study of the numerical results in two dimensions allow to concentrate on the cost function related to the Kohn-Vogelius criterion. Results in three dimensions are presented and discussed.

The conclusion summarizes the work done during this thesis and presents research perspectives.

In Appendix A three diffusion schemes are studied in order to improve the anisotropic diffusion in image processing. Numerical results are presented and discussed.

Appendix B introduces an extension to color images of Bai and Feng's restoration algorithm [24]. They used fractional derivatives in image processing to restore gray-level images.

Appendix C presents a project in collaboration with physicians from the Centre Hospitalier Universitaire (CHU) Purpan in order to perform a segmentation of a human vertebrae in three dimensions.

Appendix D presents a project in collaboration with physicists and biologists from the Université Paul Sabatier (UPS). The aim is to detect abrupt changes in a one-dimensional signal and segment the signal accordingly. A theoretical study is provided and numerical tests are performed.

1. Application of topological gradient and anisotropic diffusion in image processing

1.1. Introduction

This chapter addresses linear inverse problems defined as follows. Let u be an original image, v the observed image, L a linear observation operator and n an additive noise. These quantities are related by

$$v = L\,u + n.$$

The objective is to reconstruct u from v. A typical example is the super-resolution image reconstruction, the objective being to recover a high-definition image from one or a number of noisy filtered and sub-sampled images [90, 52]. In this case, the observation operator L is the composition of a filter and a sub-sampling operator.

The first contribution of the present work is to present a general framework with an arbitrary linear observation operator L. This framework is then applied to different image processing problems: image restoration, inpainting, demosaicing, segmentation and super-resolution. In these applications, edge detection is crucial, as edges convey essential information in a picture.

An image can be viewed as a piecewise smooth function and edges are considered as its set of singularities. Topological gradient, which is a tool able to detect cracks inside a plane domain [13], was adapted to the field of image processing in [27] to detect edges. The superiority of topological gradient on classical gradient is illustrated in Appendix D with a one dimensional case. A major advantage of topological gradient is the reduced computing time. Only $\mathcal{O}(N.\log(N))$ operations are needed to detect edges, where N is the number of pixels, see [20] for details.

The topological gradient has been used in restoration, classification, inpainting and segmentation [27, 20, 19, 21]. This technique can be applied to gray-level and color images, but also to three-dimensional images, or movies. Appendix C presents a segmentation of a human vertebrae in three dimensions

In the majority of previous works using topological asymptotic analysis in image processing, all the information derived from the asymptotic analysis is not used. The edges

are detected using topological gradient and in a second step an isotropic diffusion is applied with two coefficients: a small coefficient on the edges, and a large coefficient on smooth parts. The second contribution of the present work is to propose a reconstruction algorithm (Algorithm 2) that takes into account the complete information provided by the topological asymptotic analysis: the edges are detected, and their orientation is also used. This additional information is used to define an anisotropic diffusion tensor on the edges. Outside the edges the diffusion tensor is isotropic, and the diffusion coefficient depends on the value of the topological gradient.

The general framework and the reconstruction algorithm mentioned above are illustrated by a variety of examples: restoration, segmentation, image inpainting, demosaicing and super-resolution. These examples are compared to the existing isotropic reconstruction algorithm, and to conventional methods.

With Roberto Mecca, an Italian PhD student, we studied the use of fractional derivatives for the regularization term instead of integer derivatives. Our bibliographical research lead us to select an implementation of fractional derivatives from Bai and Feng [24]. We combined this approach with topological gradient information and proposed a reconstruction algorithm that solves the problems included in the general framework. Our method is compared to Bai and Feng denoising algorithm and to conventional methods. Because of their nonlocal property, fractional derivatives provide better reconstructions.

Section 1.2 is dedicated to the presentation and theoretical study of the general problem. The theoretical part contains also a comparative study with the Mumford-Shah functional [82], widely studied in the literature. This approach is illustrated by an example of edge detection and an application to compression. In Section 1.3 the image reconstruction algorithm (Algorithm 2) and the numerical implementation are presented. Section 1.4 considers the case $L = Id$, and the restoration algorithm from [27, 20] is improved by taking into account anisotropic diffusion. Texture removal is also presented in this section. Demosaicing and inpainting problems when L is only a sub-sampling operator are treated in Section 1.5. Section 1.6 is dedicated to super-resolution image reconstruction. Finally, Section 1.7 presents a study using fractional derivatives for the regularization term instead of integer derivatives.

1.2. General framework

This work addresses the general problem of reconstructing an image $u \in L^2(\Omega)$ given some noisy observations $Lu + n$ in a Hilbert space E, where Ω is the (rectangular) domain where the image u is defined. The observation space E depends on the specific application as well as the linear observation operator $L : L^2(\Omega) \longrightarrow E$. The norm on the space E will be written $\|.\|_E$. Table 1.1 summarizes the different applications presented in this work. These will be detailed in Sections 1.4 to 1.6.

Specific application	space E	Lu
Restoration	$L^2(\Omega)$	u
Inpainting	$L^2(\Omega\backslash\omega)$	$u_{\|\Omega\backslash\omega}$
Super-resolution with one image	$\mathbb{R}^{m\times n}$	$((\varphi * u)(x_i))_{1\leq i\leq mn}$
Super-resolution with k images	$(\mathbb{R}^{m\times n})^k$	$((\varphi * u)(x_i^j))_{1\leq i\leq mn, 1\leq j\leq k}$

Table 1.1.: Applications of the general problem (2.1). The subset $\omega \subset \Omega$ is the region where the image has to be recovered, φ is a filter (e.g. a Gaussian filter) and $x_i \in \Omega$, $i = 1, \ldots, mn$ or $x_i^j \in \Omega, i = 1, \ldots, mn, j = 1..k$ are the sampling points.

The general image reconstruction problem that we address can be summed up by:

$$\begin{cases} \text{given } v = Lu + n, \\ \text{reconstruct } u. \end{cases} \tag{1.1}$$

1.2.1. The reconstruction problem

The first idea to recover the original image is to minimize the following functional for $u \in H^1(\Omega)$:

$$\|c^{1/2}\nabla u\|^2_{L^2(\Omega)} + \|Lu - v\|^2_E, \tag{1.2}$$

where c is a positive constant or a positive definite tensor. The first term in Equation (1.2) ensures that the recovered image u is regular, and the second term measures the discrepancy with the data. The reconstruction is improved by considering a diffusion coefficient $c(x)$ that depends on the space variable. This idea is the starting point of nonlinear diffusion methods, see [16] for a review and references. The present chapter proposes an alternative construction of the space-dependent coefficient $c(x)$ based on topological asymptotic analysis. This construction is detailed in Section 1.3.1. We will assume in all this work that the following hypothesis is satisfied:

Hypothesis 1. *1. L is a linear bounded operator on E.*

2. If $u = k$ in Ω with $k \in \mathbb{R}$ then $Lu = k$.

In other words constant images in Ω are mapped to constants in E, with the same value. The adjoint operator L^* of L is also bounded.

The minimization of Equation (1.2) is equivalent to find $u \in H^1(\Omega)$ such that

$$\begin{cases} -\nabla \cdot (c\nabla u) + L^*Lu = L^*v & \text{in } \Omega, \\ \partial_n u = 0 & \text{on } \partial\Omega, \end{cases} \tag{1.3}$$

where ∂_n denotes the normal derivative to $\partial\Omega$.

The corresponding variational formulation can be written as:

$$\begin{cases} \text{find } u \in H^1(\Omega) \text{ such that} \\ a_0(u,w) = \ell_0(w) \ \ \forall w \in H^1(\Omega), \end{cases} \tag{1.4}$$

where a_0 is the bilinear form, defined on $H^1(\Omega) \times H^1(\Omega)$ by

$$a_0(u,w) = \int_\Omega c\nabla u \nabla w \, dx + \int_\Omega Lu \; Lw \, dx,$$

and ℓ_0 is the linear form defined on $L^2(\Omega)$ by

$$\ell_0(w) = \int_\Omega L^* v \; w \, dx.$$

Lemma 1.2.1. *Assume c is a positive constant or a positive definite tensor. For $u \in H^1(\Omega)$, define $\|u\|_a = \sqrt{a_0(u,u)}$. Under Hypothesis 1, $\|.\|_a$ is a norm equivalent to the norm $\|.\|_{H^1(\Omega)}$*

Lemma 1.2.1 follows directly from the general Poincaré inequality and Hypothesis 1.

Theorem 1.2.2. *Under Hypothesis 1, the variational problem* (1.4) *has a unique solution.*

Proof. To simplify the proof, the case $c = 1$ is considered.

The first assertion of Hypothesis 1 implies that a_0 is a continuous bilinear form. Lemma 1.2.1 proves that a_0 is coercive. The following inequality

$$\forall w \in L^2(\Omega) \quad |\ell_0(w)| \leq \|L^*\| \, \|v\|_E \, \|w\|_{L^2(\Omega)}$$

proves that ℓ_0 is a continuous linear form. The result follows from the Lax-Milgram theorem. □

1.2.2. Perturbation of the domain and edge detection

The solution u of the problem (1.2), or equivalently (1.3) is an element of $H^1(\Omega)$. This is a limitation of the method since it does not allow discontinuities in the recovered image. To overcome this limitation, discontinuities are included in the domain in the form of insulating cracks. This generalizes the concept introduced for the restoration equation in [27].

At a given point $x_0 \in \Omega$, we insert a small insulating crack $\sigma_\rho = x_0 + \rho\sigma(\mathbf{n})$ where $\sigma(\mathbf{n})$ is a unit line segment, $\mathbf{n}$ is a unit vector normal to the crack and $\rho > 0$ is the length of

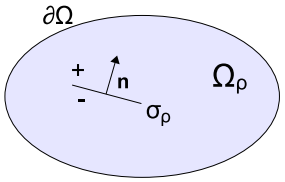


Figure 1.1.: Cracked domain.

the crack. Let $\Omega_\rho = \Omega \backslash \sigma_\rho$ be the perturbed domain created by inserting this crack. The perturbed solution $u_\rho \in H^1(\Omega_\rho)$ satisfies

$$\begin{cases} -\nabla \cdot (c \nabla u_\rho) + L^* L u_\rho = L^* v & \text{in } \Omega_\rho, \\ \partial_n u_\rho = 0 & \text{on } \partial\Omega, \\ \partial_n u_\rho = 0 & \text{on } \sigma_\rho. \end{cases} \tag{1.5}$$

The corresponding variational formulation is given by

$$\begin{cases} \text{find } u_\rho \in H^1(\Omega_\rho) \text{ such that} \\ a_\rho(u_\rho, w) = \ell_\rho(w) \ \ \forall w \in H^1(\Omega_\rho), \end{cases} \tag{1.6}$$

where a_ρ is the bilinear form, defined on $H^1(\Omega_\rho) \times H^1(\Omega_\rho)$ by

$$a_\rho(u, w) = \int_{\Omega_\rho} c \nabla u \nabla w \, dx + \int_{\Omega_\rho} L^* L u \, w \, dx,$$

and ℓ_ρ is the linear form, defined on $L^2(\Omega_\rho)$ by

$$\ell_\rho(w) = \int_{\Omega_\rho} L^* v \, w \, dx.$$

When $\rho = 0$ (i.e. the crack is reduced to a point) the solution of the variational formulation (1.6) coincides with the solution u of the unperturbed problem (1.4).

The edge detection method consists in looking for a crack σ such that the energy $j(\rho) = J_\rho(u_\rho) = \frac{1}{2} \int_{\Omega_\rho} |\nabla u_\rho|^2$ is as small as possible [27]. This amounts to state that the energy outside the edges is as small as possible.

The efficiency of topology optimization comes from the fact that the asymptotic variation of $j(\rho)$ as $\rho \to 0$ can be computed rapidly for any location x_0 and orientation $\mathbf{n}$ of the crack. It requires only to solve the direct problem (1.4) and the following adjoint problem

$$\begin{cases} \text{find } p_0 \in H^1(\Omega) \text{ such that} \\ a_0(w, p_0) = -\partial_u J(u_0)(w) \ \ \forall w \in H^1(\Omega). \end{cases} \tag{1.7}$$

The precise statement is detailed in the following

Theorem 1.2.3. *When $\rho \to 0$ the cost function j has the following asymptotic expansion*

$$j(\rho) - j(0) = \rho^2 g(x_0, \mathbf{n}) + o(\rho^2), \tag{1.8}$$

where the topological gradient g is given by

$$g(x_0, \mathbf{n}) = -\pi c(\nabla u_0(x_0).\mathbf{n})(\nabla p_0(x_0).\mathbf{n}) - \pi|\nabla u_0(x_0).\mathbf{n}|^2. \tag{1.9}$$

The function u_0 and p_0 are supposed locally regular around x_0.

Proof. The following asymptotic expansions are proved in [13, 27, 19, 14]:

i)
$$J_\rho(u_\rho) - J_\rho(u_0) = \partial_u J(u_\rho)(u_\rho - u_0) + \rho^2 \delta J_1 + o(\rho^2), \tag{1.10}$$
where
$$\delta J_1 = -\pi|\nabla u_0(x_0).\mathbf{n}|^2; \tag{1.11}$$

ii)
$$J_\rho(u_0) - J_0(u_0) = o(\rho^2); \tag{1.12}$$

iii)
$$(a_\rho - a_0)(u_0, p_\rho) = \rho^2 \delta a + o(\rho^2), \tag{1.13}$$
where the adjoint state p_ρ is defined by
$$\begin{cases} \text{find } p_\rho \in H^1(\Omega_\rho) \text{ such that} \\ a_\rho(w, p_\rho) = -\partial_u J(u_\rho)(w) \ \ \forall w \in H^1(\Omega_\rho), \end{cases} \tag{1.14}$$
and
$$\delta a = -\pi c(\nabla u_0(x_0).\mathbf{n})(\nabla p_0(x_0).\mathbf{n}); \tag{1.15}$$

iv)
$$(\ell_\rho - \ell_0)(p_\rho) = o(\rho^2). \tag{1.16}$$

Using Equations (1.10), (1.12), (1.14) and (1.6) we have

$$\begin{aligned} j(\rho) - j(0) &= J_\rho(u_\rho) - J_0(u_0) \\ &= (J_\rho(u_\rho) - J_\rho(u_0)) + (J_\rho(u_0) - J_0(u_0)) \\ &= \partial_u J(u_\rho)(u_\rho - u_0) + \rho^2 \delta J_1 + o(\rho^2) \\ &= (a_\rho - a_0)(u_0, p_\rho) + a_0(u_0, p_\rho) - a_\rho(u_\rho, p_\rho) + \rho^2 \delta J_1 + o(\rho^2) \\ &= (a_\rho - a_0)(u_0, p_\rho) - (\ell_\rho - \ell_0)(p_\rho) + \rho^2 \delta J_1 + o(\rho^2). \end{aligned}$$

Applying Equations (1.13) and (1.16) end the proof. □

The direct state u_0 and the adjoint state p_0 are computed in the initial domain without cracks. The topological gradient can be written $g(x, \mathbf{n}) = \mathbf{n}^T M(x)\mathbf{n}$, where $M(x)$ is the 2×2 symmetric matrix defined by

$$M(x) = -\pi c \frac{\nabla u_0(x)\nabla p_0(x)^T + \nabla p_0(x)\nabla u_0(x)^T}{2} - \pi \nabla u_0(x)\nabla u_0(x)^T. \qquad (1.17)$$

For a given x, $g(x, \mathbf{n})$ takes its minimal value when $\mathbf{n}$ is the eigenvector associated to the lowest eigenvalue $\lambda_{min}(x)$ of $M(x)$. This value is by definition the topological gradient associated to the optimal orientation of the crack σ_ρ at the location x. $M(x)$ is composed of two terms, see Equation (1.17). The second term takes into account the structure tensor, like other methods in anisotropic diffusion [114] and corrects it with information given by the adjoint state included in the first term. The solution of the adjoint state contains information of higher order and is less sensitive to noise (see Figure 1.6 for an example).

The edges are located at points x where $\lambda_{min}(x)$ is the most negative and their orientation is given by the corresponding eigenvector.

1.2.3. Example of edge detection: the case of a noisy image

The potential of topological gradient to detect edges and their orientation is illustrated here using an ellipse representation of the tensor $M(x)$ (1.17). In this first example the observation space is $E = L^2(\Omega)$ and the observation operator is $L = Id$.

Figure 1.2 presents an image (Barbara) perturbed by an additive Gaussian noise, and the topological gradient value $\lambda_{min}(x)$ defined as the smallest eigenvalue of the matrix $M(x)$ defined by Equation (1.17).

Figure 1.2.: From left to right, noisy Barbara image and topological gradient.

Figure 1.3 shows details of the image together with ellipses that represent, at each pixel of the image, the topological gradient tensor $M(x)$ (1.17). The orientation of the axes of the ellipse is given by the eigenvectors of $M(x)$, the ratio of the semi-axes corresponds to the ratio of the eigenvalues, and the color of the ellipse depends on the value of $\lambda_{min}(x)$: the more negative is λ_{min}, the redder is the ellipse. The following observations

confirm the efficiency of topological asymptotic analysis: the ellipses are oriented along the texture edges (the orientation is correct), and the homogeneous regions are in dark blue (the topological gradient is not very negative in homogeneous regions).

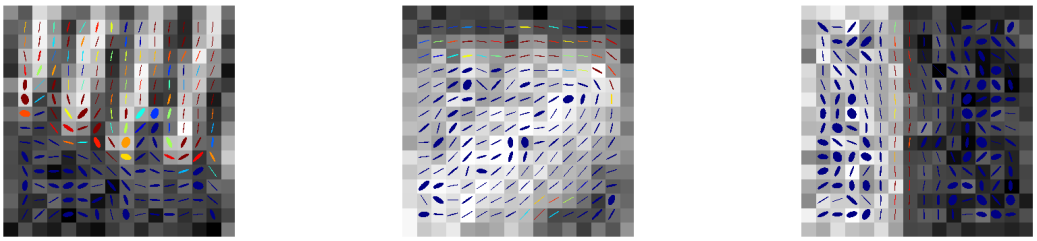


Figure 1.3.: Ellipse representation of $M(x)$: parts of noisy Barbara image 1.2. From left to right: scarf, books and table. The orientation of the axes of the ellipse is given by the eigenvectors of $M(x)$, the ratio of the semi-axes is the ratio of the eigenvalues, and the color of the ellipse depends on the value of $\lambda_{min}(x)$: the more negative is λ_{min}, the redder is the ellipse.

Figure 1.4 shows the edges obtained by the topological gradient (left) and the edges detected with the Canny detector [35] (right). We have used the Canny detector implemented by MATLAB. It consists in detecting the local maxima of the norm of the image gradient, followed by an hysteresis method. In order to allow a fair comparison, the same hysteresis method is applied to the topological gradient.

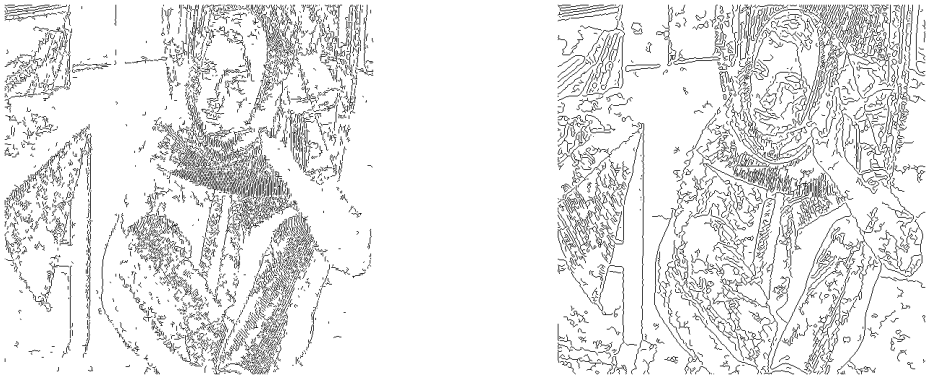


Figure 1.4.: From left to right, edges detected with respectively topological gradient and Canny detector.

1.2.4. Mumford-Shah cost function

The Mumford-Shah functional [82] is one of the most widely studied models in image processing. It solves simultaneously the problem of denoising and edge detection. The solution u is obtained by minimizing the following functional:

$$E(u,K) = \frac{\alpha}{2}\int_{\Omega\setminus K} |\nabla u|\,dx + \frac{\beta}{2}\int_{\Omega}(u-v)^2\,dx + \gamma|K|, \tag{1.18}$$

where K is the edge set. The first term controls the variation of u outside the edges. The second term measures the discrepancy to the data. The third term, the Hausdorff measure, determines the length of the edge set.

The purpose of this section is to compare the two approaches. There exists some links in literature between Mumford-Shah functional and Perona-Malik model [78, 67]. In [18], some comparison with the Ambrosio-Tortorelli approximation of the Mumford-Shah functional [7] were performed. We propose here a topological asymptotic analysis of a Mumford-Shah cost function. To simplify the reasoning, the study is done in the isotropic case, the parameters α, β and γ are supposed to be strictly positive.

The Mumford-Shah functional is decomposed in three parts for the topological asymptotic expansion in presence of a crack of length ρ. The cost functions are the following

$$J_\rho^{MS}(u_\rho) = \alpha J_{1,\rho}(u_\rho) + \beta J_{2,\rho}(u_\rho) + \gamma\rho, \tag{1.19}$$

and

$$J_0^{MS}(u_0) = \alpha J_{1,0}(u_0) + \beta J_{2,0}(u_0). \tag{1.20}$$

where $J_{1,\rho}$, $J_{1,0}$, $J_{2,\rho}$ and $J_{2,0}$ are defined by the following equations

$$J_{1,\rho}(u_\rho) = \frac{1}{2}\int_{\Omega_\rho} |\nabla u_\rho|^2 \, dx, \tag{1.21}$$

$$J_{1,0}(u0) = \frac{1}{2}\int_{\Omega} |\nabla u_0|^2 \, dx, \tag{1.22}$$

$$J_{2,\rho}(u_\rho) = \frac{1}{2}\int_{\Omega} |Lu_\rho - f|^2 \, dx, \tag{1.23}$$

$$J_{2,0}(u_0) = \frac{1}{2}\int_{\Omega} |Lu_0 - f|^2 \, dx. \tag{1.24}$$

They have the following asymptotic expansions

$$J_{1,\rho}(u_\rho) - J_{1,0}(u0) = \partial_u J_1(u_\rho)(u_\rho - u_0) - \rho^2\pi \, |\nabla u_0(x_0).n|^2 + o(\rho^2), \tag{1.25}$$

and

$$J_{2,\rho}(u_\rho) - J_{2,0}(u0) = \partial_u J_2(u_\rho)(u_\rho - u_0) + o(\rho^2). \tag{1.26}$$

Remark 1. If $J_{2,\rho}(u_\rho) = \frac{1}{2}\int_{\Omega_\rho} |Lu_\rho - f|^2 \, dx$, the asymptotic expansion is the same.

Let $c = \frac{\alpha}{\beta}$. The minimization of u_0 in (1.20) is equivalent to find $u_0 \in H^1(\Omega)$ such that

$$\begin{cases} -\nabla \cdot (c\nabla u_0) + L^*Lu_0 = L^*v & \text{in } \Omega, \\ \partial_n u_0 = 0 & \text{on } \partial\Omega, \end{cases} \tag{1.27}$$

and the minimization of u_ρ in (1.19) is equivalent to find $u_\rho \in H^1(\Omega_\rho)$ such that

$$\begin{cases} -\nabla \cdot (c\nabla u_\rho) + L^*Lu_\rho = L^*v & \text{in } \Omega_\rho, \\ \partial_n u_\rho = 0 & \text{on } \partial\Omega, \\ \partial_n u_\rho = 0 & \text{on } \sigma_\rho. \end{cases} \tag{1.28}$$

Let $p_{1,\rho}$ be the solution of the following equation

$$\begin{cases} \text{find } p_{1,\rho} \in H^1(\Omega_\rho) \text{ such that} \\ a_\rho(w, p_{1,\rho}) = -\partial_u J_{1,\rho}(u_\rho)(w) \ \ \forall w \in H^1(\Omega_\rho). \end{cases} \tag{1.29}$$

Let $p_{2,\rho}$ be the solution of the following equation

$$\begin{cases} \text{find } p_{2,\rho} \in H^1(\Omega_\rho) \text{ such that} \\ a_\rho(w, p_{2,\rho}) = -\partial_u J_{2,\rho}(u_\rho)(w) \ \ \forall w \in H^1(\Omega_\rho). \end{cases} \tag{1.30}$$

The direct equation (1.28) gives us the following equality

$$\begin{aligned} &-\nabla \cdot (c\nabla u_\rho) + L^*Lu_\rho = L^*v, \\ \Leftrightarrow &-\nabla \cdot (c\nabla u_\rho) = L^*(v - Lu_\rho), \\ \Leftrightarrow &-c\partial_u J_{1,\rho}(u_\rho) = \partial_u J_{2,\rho}(u_\rho) \end{aligned} \tag{1.31}$$

so that the solutions of Equations (1.29) and (1.30) satisfy:

$$p_{2,\rho} = -c\, p_{1,\rho}. \tag{1.32}$$

The cost function (1.19) has the following asymptotic expansion

$$\begin{aligned} &J_\rho^{MS}(u_\rho) - J_0^{MS}(u_0) \\ =&\alpha J_{1,\rho}(u_\rho) + \beta J_{2,\rho}(u_\rho) + \gamma\rho - \alpha J_{1,0}(u_0) + \beta J_{2,0}(u_0) \\ =&\alpha\partial_u J_1(u_\rho)(u_\rho - u_0) + \beta\partial_u J_2(u_\rho)(u_\rho - u_0) - \alpha\rho^2\pi \left|\nabla u_0(x_0).n\right|^2 + \gamma\rho + o(\rho^2) \\ =&a_\rho(u_0 - u_\rho, \alpha p_{1,\rho} + \beta p_{2,\rho}) - \alpha\rho^2\pi \left|\nabla u_0(x_0).n\right|^2 + \gamma\rho + o(\rho^2) \\ =&(\alpha - c\beta)a_\rho(u_0 - u_\rho, p_{1,\rho}) - \alpha\rho^2\pi \left|\nabla u_0(x_0).n\right|^2 + \gamma\rho + o(\rho^2). \end{aligned} \tag{1.33}$$

The coefficient c in Equations (1.27) and (1.28) is equal to $\frac{\alpha}{\beta}$, so the first term of the asymptotic equation is equal to zero. The topological gradient analysis for the Mumford-Shah cost function gives the following result

Theorem 1.2.4. *When $\rho \to 0$, the cost function J^{MS} has the following asymptotic expansion*

$$J_\rho^{MS}(u_\rho) - J_0^{MS}(u_0) = \rho\gamma + \rho^2 g(x_0, \mathbf{n}) + o(\rho^2), \tag{1.34}$$

where the topological gradient g_{MS} is given by

$$g_{MS}(x_0, \mathbf{n}) = -\alpha\pi |\nabla u_0(x_0).\mathbf{n}|^2. \tag{1.35}$$

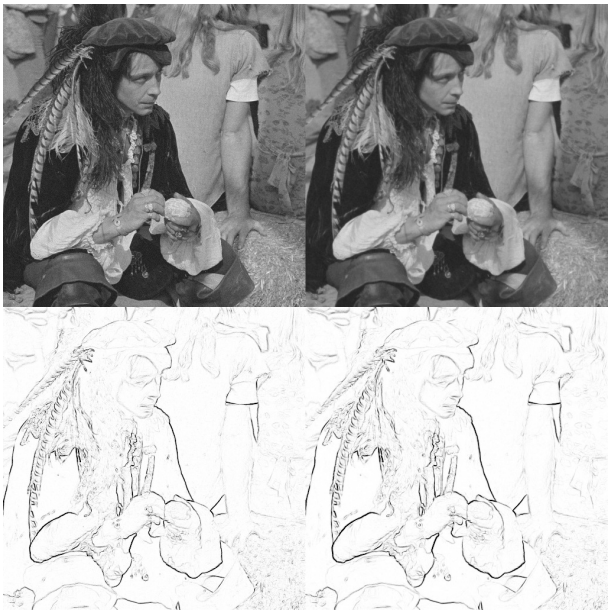


Figure 1.5.: From left to right, first line noisy Man image ($\sigma = 0.02$), solution of Equation (1.3), second line, lowest eigenvalues of $M_{MS}(x)$ (1.36), lowest eigenvalues of $M(x)$ (1.17).

Considering Equation (1.34), there seems to be no interest to create an infinitesimal crack when $\rho \to 0$. On second thought hoowever, if the purpose is to insert a crack, the length penalty which is positive is the same for any crack of the same size, only the Mumford-Shah topological gradient $g_{MS}(x_0, \mathbf{n})$, which is negative, is of interest.

Theorem 1.2.4 suggests a minimization procedure for the Mumford-Shah functional. The Mumford-Shah topological gradient $g_{MS}(x_0, \mathbf{n})$ is negative, while the length penalty γ is positive. Thus, a greedy strategy to minimize the Mumford-Shah asymptotic expansion (1.34) would consist in adding cracks where $g_{MS}(x_0, \mathbf{n})$ is minimal.

Now let us compare this result with the one obtained in Theorem 1.2.3 for the topological gradient. The Mumford-Shah topological gradient (1.35) can be rewritten $g_{MS}(x, \mathbf{n}) = \mathbf{n}^T M_{MS}(x)\mathbf{n}$, where M_{MS} is the 2×2 symmetric matrix defined by

$$M_{MS}(x) = -\alpha\pi \nabla u_0(x) \nabla u_0(x)^T. \tag{1.36}$$

One can note that there is only the structure tensor in Equation (1.36). This should be compared to the matrix M obtained in (2.37) where one can note the presence of the adjoint state. This term allows to use information of higher order and lowers the sensitivity to noise, see Figure 1.6.

In Figure 1.5, a 512×512 pixels image was perturbed by an additive Gaussian noise of standard deviation $\sigma = 0.02$. The direct solution of Equation (1.3) with a diffusion coefficient c equal to 0.5 is given. The lowest eigenvalues of M_{MS} (1.36) and the lowest eigenvalues of M (1.17) have been displayed for comparison. It can be seen than the results are similar.

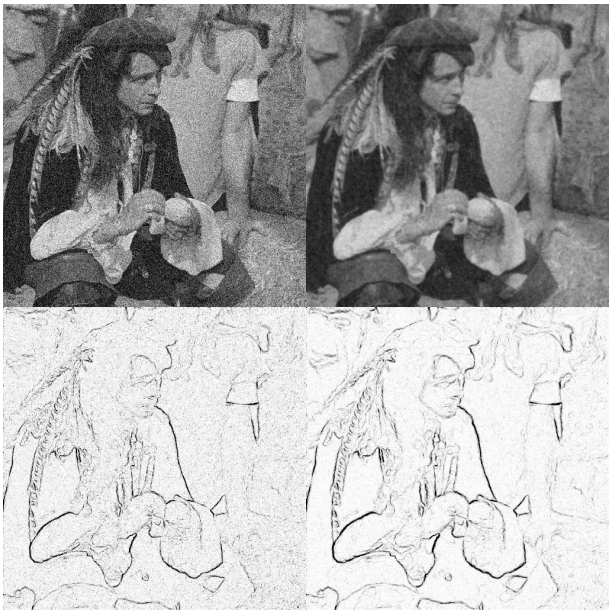


Figure 1.6.: From left to right, first line noisy Man image ($\sigma = 0.12$), solution of Equation (1.3), second line, lowest eigenvalues of $M_{MS}(x)$ (1.36), lowest eigenvalues of $M(x)$ (1.17).

In Figure 1.6, the same experiment has been performed with an additive Gaussian noise of standard deviation $\sigma = 0.12$ and $c = 1.5$. The topological gradient, less sensitive to noise, gives better results than the structure tensor of the direct solution.

1.2.5. Application of edge detection to image compression

Compression is an important field of digital image processing. The capabilities of partial differential equations only begin to be explored in this context. In [73, 104], the authors present a method that in some experiments, performs similarly or with better results compared to the widely-used JPEG standard and even JPEG 2000. To sum it up, this method uses edge detectors and keeps only the pixel information on and around the edges. An inpainting process involving homogeneous diffusion is then applied to restore the unknown information. Figure 1.7 presents a toy example.

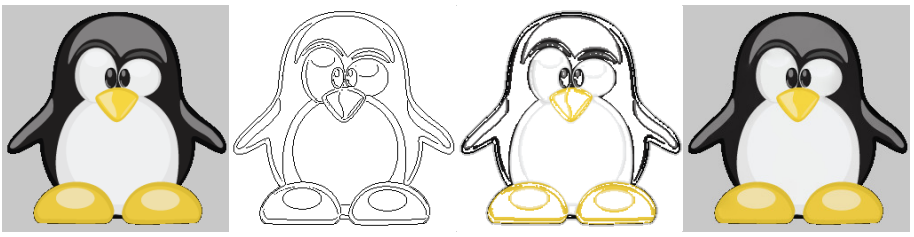


Figure 1.7.: From left to right, original image, detected edges, kept pixels (20%), reconstruction.

The compression gain is mainly provided by a good implementation of the information retained. This method is very efficient in cartoon-likes images (cliparts, for example).

1.3. Algorithms and numerical implementation

1.3.1. Algorithms

In previous works on image processing using topological gradient [27, 20, 19, 21] the tensor c is an isotropic tensor taking only two values: c_0 in the smooth part of the image and a small value ϵ on the edges. The edges are detected with a threshold δ on the topological gradient values. These ideas are summarized in Algorithm 1, which is the isotropic diffusion (ID) reconstruction algorithm. It is a generalization with an arbitrary observation operator L of the algorithm presented in [27].

Algorithm 1 (ID) Generalization of the algorithm for isotropic diffusion [27]

Input: perturbed image v, diffusion coefficient c_0, $\epsilon > 0$, threshold $\delta < 0$.
Output: restored image u

1: Initialization: $c = c_0$.
2: Computation of u_0 and p_0, solutions of direct and adjoint problems, see Equations (1.4) and (1.7).
3: Computation of the tensor $M(x)$ using equation (1.17).
4: Computation of the smallest eigenvalue $\lambda_{min}(x)$ of $M(x)$ at each point of the domain.
5: Set $c(x) = \begin{cases} \epsilon & \text{if } \lambda_{min}(x) < \delta \\ c_0 & \text{otherwise} \end{cases}$
6: Computation of u, solution to problem (1.4).

The present work proposes to take into account the whole information provided by topological asymptotic analysis, i. e. edge orientation and jump amplitude. These quantities allow to define a diffusion tensor $c(x)$ that is adapted to the image content.

More precisely the tensor $M(x)$ is computed using Equation (1.17). Let the smallest eigenvalue of $M(x)$ be denoted $\lambda_{min}(x)$. The valley bottoms are local minima of λ_{min} along the horizontal direction, the vertical direction or one of the diagonals. The edges are valley bottoms where λ_{min} is below a threshold δ. The diffusion tensor $c(x)$ is anisotropic along the edges with the principal direction given by $M(x)$, and $c(x)$ is isotropic outside the edges. This is detailed in Algorithm 2.

1.3.2. Numerical implementation

The algorithms were coded in MATLAB and equations (1.1) and (1.7) were solved with a finite difference method.

Our experiments used 8-bits images. The intensity of these images was divided by 255, so we considered normalized images with an intensity in the interval $[0, 1]$. When an

Algorithm 2 (AD) Algorithm to solve the image reconstruction problem (1.1)

Input: perturbed image v, diffusion coefficient c_0, $\epsilon > 0$, threshold $\delta < 0$.
Output: restored image u

1: Initialization: $c = c_0$.
2: Computation of the solutions u_0 and p_0 of the direct and adjoint problems, see Equations (1.4) and (1.7).
3: Computation of the tensor $M(x)$ using Equation (1.17).
4: Computation of the eigenvalue decomposition: $M(x) = (PDP^{-1})(x)$.
5: Extraction of the valley bottoms of λ_{min}.
6: Set

$$c(x) = \begin{cases} P(x) \begin{pmatrix} \epsilon & 0 \\ 0 & c_0 \end{pmatrix} P^{-1}(x) & \text{if } x \text{ is in a valley and } \lambda_{min}(x) < \delta, \\ c_0 \exp((\lambda_{min}(x) - \delta)/|\delta|) I_d & \text{otherwise.} \end{cases}$$

7: Computation of u, solution of (1.4) with a diffusion tensor $c(x)$.

additive Gaussian noise is addded, the standard deviation of the noise is denoted σ.

The different reconstructions are compared quantitatively using Peak Signal to Noise Ratio (PSNR) expressed in dB and the Structural SIMilarity (SSIM) [112]. Let I_2 be a degraded image of an original image I_1, m and n be the image dimensions and ch the number of channels. The PSNR of I_2 is given by the following formula:

$$PSNR(I_2) = 10 \log_{10} \left(\frac{m\, n\, ch}{\sum_{m,n} (I_1(i,j) - I_2(i,j))^2} \right).$$

The SSIM of I_2 is given by the following formula:

$$SSIM = \frac{(2\mu_1\mu_2 + c_1)(2\sigma_{1,2} + c_2)}{(\mu_1^2 + \mu_2^2 + c_1)(\sigma_1^2 + \sigma_2^2 + c_2)},$$

where μ_1 is the average of I_1, μ_2 is the average of K, σ_1^2 is the variance of I_1, σ_2^2 is the variance of I_2, $\sigma_{1,2}^2$ is the covariance of I_1 and I_2, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ two variables to stabilize the division by a weak denominator (where L is the dynamic range, usually $2^{\#\text{bits per pixel}} - 1$, and $k_1 = 0.01$, $k_2 = 0.03$ by default)

In order to perform a comparative evaluation, benchmark methods had to be chosen among many reconstruction methods for each application considered (restoration, demosaicing, super-resolution). Our choice was driven by two criteria :

- ➢ the use of well-established methods,
- ➢ the availability of efficient codes.

In these methods, some parameters often have to be adjusted. These parameters were chosen to optimize the results.

The color images were treated with a vectorial minimization problem, involving the resolution of vectorial problems. The topological asymptotic expansion is still given by Equation (1.8), (1.9) and (1.17), where all functions are vectorial, i.e. the topological gradient is the sum of the corresponding expressions for each channel [18].

1.4. The observation operator is $L = Id$

A classical way to restore an image u from its noisy version $v \in L^2(\Omega)$ is to find $u \in H^1(\Omega)$ which minimizes the following functional:

$$\|c^{1/2}\nabla u\|^2_{L^2(\Omega)} + \|u - v\|^2_{L^2(\Omega)}. \tag{1.37}$$

It is equivalent to solve the following PDE:

$$\begin{cases} -\nabla \cdot (c\nabla u) + u = v \text{ in } \Omega, \\ \partial_n u = 0 \text{ on } \partial\Omega, \end{cases} \tag{1.38}$$

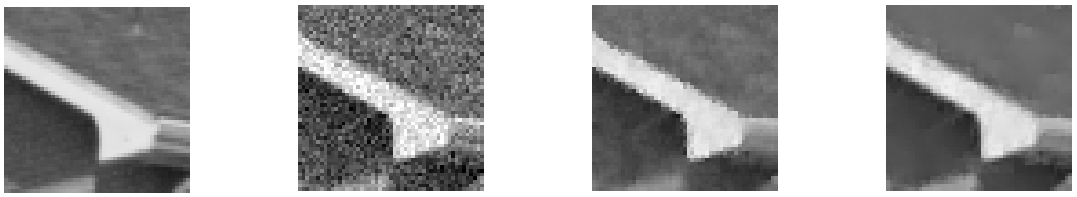
where c is a positive constant or a tensor and ∂_n denotes the normal derivative to $\partial\Omega$. It is a particular case of Equation (1.3) with the observation operator $L = Id$. This equation and improvements using anisotropic diffusion have been the subject of numerous works [93, 94, 114, 16].

In Section 1.4.1, our Algorithm 2 is compared to the existing Algorithm 1 and to the Non-Local Means method [34] on one example. Section 1.4.2 is dedicated to results obtained when the same equation is used to remove texture. The main interest is the possibility to obtain a cartoon image without texture.

1.4.1. Image restoration

Figure 1.8.: Noisy and AD restoration of House and Barbara images

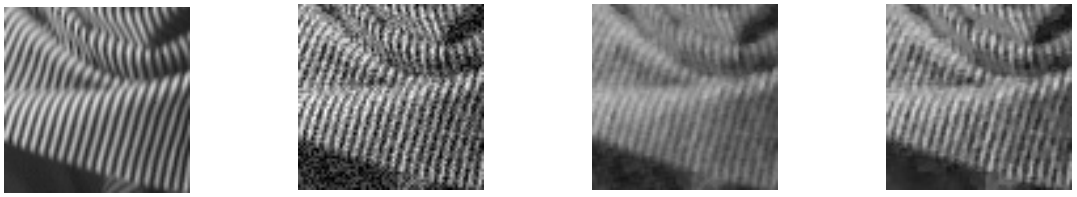
A 512×512 pixels image was perturbed by an additive Gaussian noise of standard deviation $\sigma = 0.1$. The resulting image is the House image, see Figure 1.8 (left). The



(a) Original image. (b) Noisy image. (PSNR 20.94 dB) (c) ID restoration. (PSNR 29.47 dB) (d) AD restoration. (PSNR 31.46 dB)

Figure 1.9.: Parts of House image (from left to right): original image, noisy image, ID and AD restorations.

results of the anisotropic restoration Algorithm 2 is presented in Figure 1.8 (center left). In order to illustrate the difference between Algorithm 1 (ID) and Algorithm 2 (AD), details are shown in Figure 1.9 (c) and (d). It can be seen that the restoration of edges is more accurate using the AD restoration.



(a) Original image. (b) Noisy image. (PSNR 19.86 dB) (c) ID restoration. (PSNR 25.09 dB) (d) AD restoration (PSNR 25.93 dB)

Figure 1.10.: Parts of original Barbara image, noisy one, ID and AD restorations.

A 512×512 pixels image was perturbed by an additive noise following a uniform distribution between $[-0.176, 0.176]$. The resulting image is the Barbara image, see Figure 1.8 (center right). The results of the anisotropic restoration Algorithm 2 is presented in Figure 1.8 (right). The details presented in Figure 1.10 (c) and (d) show the interest of anisotropic diffusion along the edges.

Figure 1.11.: From right to left: noisy, NLM and AD restorations of Mandrill image.

In presence of an additive Gaussian noise, a comparative study with the Non-Local Means (NLM) [34] was performed to restore an image that is not composed of self-similar texture (Mandrill, Figure 1.11). We use the toolbox provided by [95].

It must be noted that on images that contain self-similar texture (e.g. Barbara), the results provided by the NLM outperform our diffusion method. This is sensible since in the NLM algorithm, the denoised value at a pixel x is a mean of the values of all pixels whose Gaussian neighborhood looks like the neighborhood of x. The efficiency of this algorithm on a self similar image is therefore foreseeable.

	Mandrill Figure 1.11		
	Noisy	NLM	AD
PSNR	20.16	22.63	23.00
SSIM	0.552	0.708	0.702

Table 1.2.: Comparative results of restoration using Non-Local Means (NLM) and Algorithm 2 (AD).

(a) Original image. (b) Noisy image. (PSNR 20.16 dB) (c) Non-Local Means. (PSNR 22.63 dB) (d) AD restoration. (PSNR 23.00 dB)

Figure 1.12.: Parts of original Mandrill image, noisy one, NLM and AD restorations.

Figure 1.11 shows the Mandrill ($512 \times 512 \times 3$) color image with an additive Gaussian noise of standard deviation $\sigma = 0.09$ and the restorations with NLM and AD. The three channels (RGB) were treated separately. The Non-Local Means and AD restorations provide similar results in terms of quality (see Table 1.2 and Figure 1.12 (c) and (d)).

This restoration algorithm is not only useful to suppress noise, it can also help removing JPEG artifacts. A $512 \times 512 \times 3$ pixels color image is perturbed by a JPEG compression using the GNU Image Manipulation Program [108]. The resulting image is the Peppers image, see Figure 1.13 (left). The results of the anisotropic restoration Algorithm 2 is presented in Figure 1.13 (right).

1.4.2. Image segmentation

The objective of cartoon plus texture decomposition is to remove the texture of an image while preserving its main features. This can be achieved by using a large diffusion coefficient in Equation (1.38). In order to prevent edge degradation, the following isotropic diffusion coefficient was used:

$$c(x) = \begin{cases} \epsilon & \text{in } \omega \\ \frac{1}{\epsilon} & \text{outside } \omega \end{cases}$$

Figure 1.13.: Suppression of JPEG artefacts.

where $\omega \subset \Omega$ represent the edge set. The following problem adapted from (1.38) was solved:

$$\begin{cases} -\nabla \cdot (\epsilon \nabla u_\epsilon) + u_\epsilon = v \text{ in } \omega, \\ -\nabla \cdot \left(\frac{1}{\epsilon} \nabla u_\epsilon\right) + u_\epsilon = v \text{ in } \Omega \backslash \omega, \\ \partial_n u = 0 \text{ on } \partial\Omega, \end{cases}$$

The segmentation provides an image composed of homogeneous regions. The suppression of texture or the separation of an image between cartoon part (smooth regions) and texture part is a developing task [17, 115]. Figure 1.14 presents a texture removal example with an image of a cross stitch.

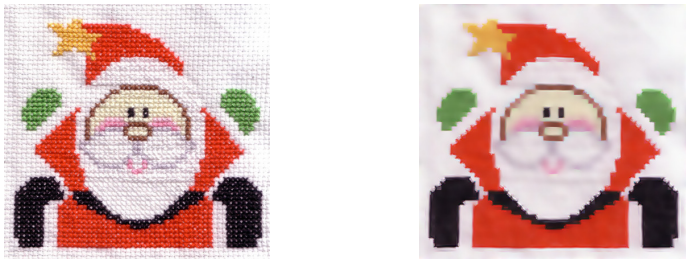


Figure 1.14.: Texture removal.

1.5. The observation operator L is a mask operator

Image inpainting has become a generic term to refer to the process of restoring missing and/or damaged areas in digital images. The problem can be stated as follows: given a region ω to be restored, use the valid surrounding information for synthesizing the most plausible data in ω. Different approaches address the inpainting problem. The first category of approaches focuses on recovering the geometry [109, 33]. The second category is dedicated to texture synthesis [43, 113]. Our approach enters the first one.

Let ω be the missing part of the image. The observation operator is $L : L^2(\Omega) \mapsto L^2(\Omega \backslash \omega)$ defined by $Lu = u|_{\Omega \backslash \omega}$. To solve inpainting problems, the following functional is minimized with $u \in H^1(\Omega)$:

$$\|c^{1/2} \nabla u\|^2_{L^2(\Omega)} + \|Lu - v\|^2_{L^2(\Omega \backslash \omega)}, \tag{1.39}$$

where the anisotropic tensor $c(x)$ is computed using Algorithm 2.

Section 1.5.1 addresses the problem of denoising a salt and pepper image, by considering that ω is the union of the pixels where the gray-level is saturated in black or white (gray-level 0 or 255 for 8-bit images). Section 1.5.2 proposes a unified approach to perform demosaicing and image denoising simultaneously.

1.5.1. Salt and Pepper denoising

Figure 1.15.: From left to right, first line, original Lena image and noisy image (PSNR 6.30 dB), second line, isotropic restoration (PSNR 32.78 dB) and anisotropic restoration (PSNR 35.66 dB).

In Figure 1.15, the gray-level Lena image (1024×1024) suffers from a Salt and Pepper noise, and 80% of the pixels are affected. The pixels in black and white are identified,

their union is the unknown region ω. The results of Algorithm 1 and Algorithm 2 on this problem are presented in Figure 1.15. The anisotropic diffusion provides better results in term of visual quality and PSNR assessment.

1.5.2. Demosaicing and denoising

Demosaicing, also called color filter array (CFA) interpolation, refers to the problem of reconstructing a color image from the charge-coupled device (CCD) samples. A color filtering array is a mosaic of color filters in front of the image sensor. While there are many kinds of color filters in use in digital cameras, the most commonly used is the Bayer filter. It is a grid composed of squares representing the three primary colors, see Figure 1.16. To be noted that there are twice as many green photo sensors as red or blue, as the human eye is more sensitive to green light.

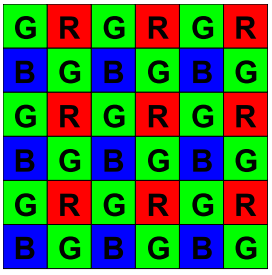


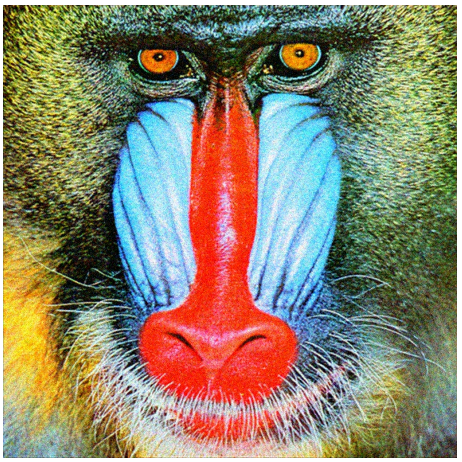
Figure 1.16.: The Bayer filter grid.

To reconstruct a color image from the uncompleted data, interpolation is needed to fill in the blanks. This specific inpainting application is called demosaicing. Demosaicing and image denoising are treated simultaneously with our method and the results are presented in Figure 1.17.

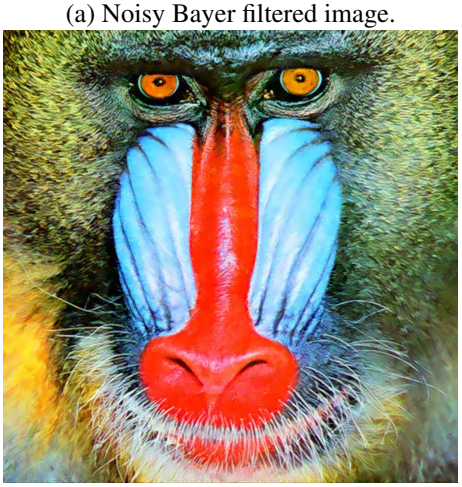
For this example, the Mandrill original image ($1024 \times 1024 \times 3$) is used. Figure 1.17 (a) is a Bayer filtered image perturbed by an additive Gaussian noise of standard deviation $\sigma = 0.1$. Algorithm 2 was compared with two existing methods. Figure 1.17 (b) presents the reconstruction with the Alternating Projection technique (AP) [58]. The AP is a benchmark method in demosaicing applications, and is not designed for denoising. The result is given to get an idea of the noise amount.

The coupling demosaicing-denoising has been treated in [63] using a Total Least Square method (TLS). The result of the TLS image denoising method from [63] is presented in Figure 1.17 (c). The reconstruction obtained with Algorithm 2 is Figure 1.17 (d). The quality of the results and computational aspects are summarized in Table 1.3. Is is clear that our Algorithm 2 provides similar results in terms of quality as Total Least Square and this with a lower computational load.

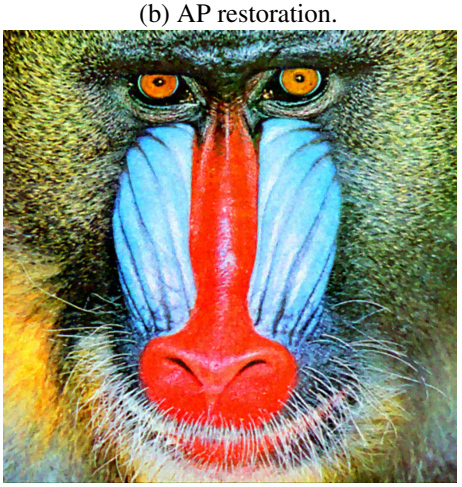
(a) Noisy Bayer filtered image.



(b) AP restoration.



(c) TLS restoration.



(d) AD restoration.

Figure 1.17.: Demosaicing and denoising of a color image.

	Mandrill Figure 1.17		
	AP	TLS	AD
PSNR	19.85	23.89	24.62
SSIM	0.5119	0.7438	0.7600
CPU (s)	14	3420	170

Table 1.3.: Comparative results for demosaicing and denoising using Alternative Projections (AP), Total Least Square (TLS) and Algorithm 2 (AD).

1.6. Case of super-resolution

The purpose of super-resolution techniques is to obtain a high resolution image from one or several low resolution images [90, 52]. Let k be the number of low resolution images. The observation operator is $L : L^2(\Omega) \to E = (\mathbb{R}^{m\times n})^k$ and it is the composition of a convolution by a filter followed by k subsampling operators. In the examples below a Gaussian noise is added to the observations.

To solve super-resolution problems, the following functional is minimized with $u \in H^1(\Omega)$:

$$\|c^{1/2}\nabla u\|^2_{L^2(\Omega)} + \|Lu - v\|^2_E, \tag{1.40}$$

where the anisotropic tensor $c(x)$ is computed using Algorithm 2.

Section 1.6.1 shows the capability of Algorithm 2 to treat one or many noisy filtered and sub-sampled images. In Section 1.6.2, a comparison with a Total Variation algorithm is proposed.

1.6.1. Super-resolution with one or many filtered and sub-sampled images

For this example, the original Lena color image ($512 \times 512 \times 3$) is used. The image is convolved with a Gaussian 5×5 filter with standard deviation $\tau = 5$. The sub-sampling operation keeps one pixel out of nine and three low-resolution images are sub-sampled differently. An additive Gaussian noise of standard deviation $\sigma = 0.075$ is added to each low-resolution image. In Figure 1.18 (center), only one filtered and sub-sampled image is used for the reconstruction. In Figure 1.18 (right), the observations are $k = 3$ filtered and sub-sampled images.

Figure 1.18.: Images from left to right: original, noisy filtered and sub-sampled, restored (PSNR 19.28 dB), three noisy filtered and sub-sampled, restored (PSNR 20.24 dB).

1.6.2. Comparison with Total Variation algorithm

The results of Algorithm 2 were compared with a TV-L^2 algorithm [74, 84]. A grey-level image is sub-sampled by a factor 2 in each direction with a mean 2×2 filter. A Gaussian noise with standard deviation $\sigma = 0.05$ is added to the sub-sampled image. Figure 1.19 presents the original Man image (512×512), the observed image, the image restored with a TV-L^2 algorithm and the image recovered using Algorithm 2. Table 1.4 presents quantitative estimators of the quality of reconstructions, the tests are performed on Man, Barbara and Boat images. The quality of the reconstruction with our Algorithm 2 outperforms the TV-L^2 method.

(a) Original image.

(b) Noisy low-resolution image.

(c) TV restoration.

(d) AD restoration.

Figure 1.19.: Super resolution of Man image.

	Man		Barbara		Boat	
	TV	AD	TV	AD	TV	AD
PSNR	24.81	28.07	22.63	24.32	24.03	27.41
SSIM	0.652	0.753	0.598	0.623	0.626	0.728

Table 1.4.: Comparative results of super-resolution using TV-L^2 optimization (TV) and Algorithm 2 (AD).

1.7. Fractional order diffusion

The main idea of this section is to use fractional derivatives for the regularization term instead of integer derivatives. In the last 30 years, fractional calculus began to shift from pure mathematics formulations to applications in various fields. Some of the areas where fractional calculus has been applied include biology, physics, mechanics and many other fields such as image processing. Some examples of recent works [2, 85, 76] illustrate the evolution of the fractional calculus in the last thirty years.

In particular in the image processing field, the nonlocal properties of fractional differential-based approaches appear to give better results than traditional integral-based algorithms. A variety of problems in image processing have already been addressed with fractional derivatives: image restoration [40, 42, 24], texture detection and enhancement [97, 98] and super-resolution [116].

Our purpose is to minimize the following functional:

$$\|c^{\frac{1}{2}}\nabla^{\alpha}u\|^2_{L^2(\Omega)} + \|Lu - v\|^2_E \tag{1.41}$$

where α represents the order of the derivative with a finite L^2 norm (i.e minimization in the $H^{2\alpha} \equiv W^{2\alpha,2}$ space functions with $\alpha > 0$). The Sobolev space can be defined as follows:

Definition 1.7.1. For any $s \in \mathbb{R}$, the space $H^s(\Omega)$ consists of tempered distributions g such that

$$(1 + |w_1|^2 + |w_2|^2)^{\frac{s}{2}}\hat{g} \in L^2(\Omega), \tag{1.42}$$

where $\hat{g}(w_1, w_2) = \int_\Omega g(x, y)\exp(-i(w_1x + w_2y))dxdy$.

It is also a Hilbert space equipped with the canonical inner product.

The minimization of the functional (1.41) is equivalent to consider the associated Euler-Lagrange equation:

$$\begin{cases} (\nabla^{\alpha})^*(c\nabla^{\alpha}u) + L^*Lu = L^*v & \text{in } \Omega, \\ \nabla^{\alpha}u \cdot n = 0 & \text{on } \partial\Omega. \end{cases} \tag{1.43}$$

where n is the external normal to the boundary $\partial\Omega$.

The first contribution of this section consists in using fractional partial differential equations to solve different kinds of image reconstruction problems. In [24], Bai and Feng use fractional derivatives for image denoising with an iterative process. However, the computing time remains a major drawback of their method and the second contribution of this section proposes an efficient algorithm able to solve this issue. The diffusion coefficient $c(x)$ depends on the space variable. Whereas in [24], this coefficient $c(x)$ evolves during the iterative process, we propose to fix it and reconstruct the image in one iteration using the topological gradient information.

Section 1.7.1 recalls a way to calculate fractional derivative using Fourier transform and presents our image reconstruction algorithm. Section 1.7.2 compares the numerical results in image denoising with Bai and Feng's algorithm, and some denoising and inpainting applications are performed and compared with other established methods involving partial differential equations.

1.7.1. Fractional implementation and algorithm

This section gives a definition of the fractional derivative in the frequency domain, then recalls the implementation of the fractional order gradient from Bai and Feng [24] and presents our algorithm.

For any function $u(x, y) \in L^2(\mathbb{R})$, the 2-D Fourier transform of u is defined as

$$\hat{u}(w_1, w_2) = \int_{\mathbb{R}^2} u(x, y) \exp(-i(w_1 x + w_2 y))dxdy. \tag{1.44}$$

The first order derivatives in the frequency domain are

$$\begin{aligned} \widehat{\partial_x u(x, y)} &= iw_1 \hat{u}(w_1, w_2), \\ \widehat{\partial_y u(x, y)} &= iw_2 \hat{u}(w_1, w_2). \end{aligned} \tag{1.45}$$

With an order $\alpha \in \mathbb{R}$, the fractional derivatives in the frequency domain are

$$\begin{aligned} \widehat{\partial_x^\alpha u(x, y)} &= (iw_1)^\alpha \hat{u}(w_1, w_2), \\ \widehat{\partial_y^\alpha u(x, y)} &= (iw_2)^\alpha \hat{u}(w_1, w_2). \end{aligned} \tag{1.46}$$

The fractional order derivative can be seen as a generalization of the integer order derivative. The frequency domain is used because of the easy implementation of the fractional derivatives.

For the next, let $D^\alpha \equiv \nabla^\alpha$ be the fractional operator having the same structure as the gradient operator, that is $D^\alpha u = \nabla^\alpha u = (D_x^\alpha u, D_y^\alpha u)$. The computation of fractional derivative is given for the discrete image domain where it is assumed that u has $m \times m$ pixels. This domain consists of a uniform grid starting at $(0, 0)$, with $u(x, y) = u(x\Delta x, y\Delta y)$ for x and y in $\{0, \ldots, m-1\}$, where the grid size is chosen so that $\Delta x = \Delta y = 1$.

In the discrete image domain, and then for the effective calculation of the fractional derivative, the following definition of two-dimensional Discrete Fourier Transform (2D-DFT) is used

$$F(u)(w_1, w_2) = \frac{1}{m^2} \sum_{x,y=0}^{m-1} u(x, y) \exp\left(-i2\pi \frac{w_1 x + w_2 y}{m}\right). \tag{1.47}$$

where w_1 and w_2 belong to $\{0, \ldots, m-1\}$. The DFT views both the time domain and the frequency domain as periodic.

Using the gradient approximation with the finite difference, it is possible to write the following relation

$$F(u - T_x u) = K_x^1 F(u), \tag{1.48}$$

where K_x^1 is a diagonal operator defined by $K_x^1 = \text{diag}\left(1 - \exp\left(-i2\pi\frac{w_1}{m}\right)\right)$ and T_x a translation operator with periodic boundary conditions, $T_x u(x,y) = u(x-1,y)$.

So we define the fractional derivative as

$$D_x^\alpha u = F^{-1}\left(K_x^\alpha F(u)\right), \tag{1.49}$$

where $K_x^\alpha = \text{diag}\left(\left(1 - \exp\left(-i2\pi\frac{w_1}{m}\right)\right)^\alpha\right)$.

In order to use a centred difference scheme to compute the fractional derivative, a translation of D_x^α is made by $\frac{\alpha}{2}$. The last formulation of fractional derivative takes the following form

$$\tilde{D}_x^\alpha u = D_x^\alpha\left(\mathfrak{u}\left(x + \frac{\alpha}{2}, y\right)\right), \tag{1.50}$$

where $\mathfrak{u}$ is the interpolation of u outside the discrete set of points of the image. As a correspondence of this equivalence (1.50) it is possible to write the following relation:

$$\tilde{D}_x^\alpha u = F^{-1}\left(\tilde{K}_x^\alpha F(u)\right), \tag{1.51}$$

where $\tilde{K}_x^\alpha = \text{diag}\left(\left(1 - \exp\left(-i2\pi\frac{w_1}{m}\right)\right)^\alpha \exp\left(i\pi\alpha\frac{w_1}{m}\right)\right)$.

The adjoint of F is F^{-1}, thanks to the following relation

$$\begin{aligned}\left\langle \tilde{D}_x^\alpha u, v\right\rangle_{L^2(\Omega)} &= \left\langle F^{-1}\left(\tilde{K}_x^\alpha F(u)\right), v\right\rangle_{L^2(\Omega)} \\ &= \left\langle u, F^{-1}\left(\tilde{K}_x^{\alpha*} F(v)\right)\right\rangle_{L^2(\Omega)} = \left\langle u, \tilde{D}_x^{\alpha*} v\right\rangle_{L^2(\Omega)},\end{aligned}$$

the adjoint operator $\tilde{D}_x^{\alpha*}$ is defined as follows:

$$\tilde{D}_x^{\alpha*} u = F^{-1}\left(\tilde{K}_x^{\alpha*} F(u)\right). \tag{1.52}$$

Since $\tilde{K}_x^\alpha$ is a purely diagonal operator, $\tilde{K}_x^{\alpha*}$ is the complex conjugate of $\tilde{K}_x^\alpha$.

Bai and Feng reflect the image symmetrically across the border in order to reduce discontinuities across the image border due to the periodization. The same procedure is used in this section.

The aim is to solve the following equation:

$$D_x^{\alpha*} c_x D_x^\alpha u + D_y^{\alpha*} c_y D_y^\alpha u + u = v \tag{1.53}$$

In order to obtain a diffusion function in the vertical and horizontal directions, the definition of the topological gradient g (1.9) can be simplified as:

$$\begin{aligned} g_x(\mathbf{x}) &= -\pi c \partial_1 u_0(\mathbf{x}) \partial_1 p_0(\mathbf{x}) - \pi(\partial_1 u_0(\mathbf{x}))^2, \\ g_y(\mathbf{x}) &= -\pi c \partial_2 u_0(\mathbf{x}) \partial_2 p_0(\mathbf{x}) - \pi(\partial_2 u_0(\mathbf{x}))^2. \end{aligned} \tag{1.54}$$

Algorithm 3 Algorithm to solve the image reconstruction problem (1.41)

Input: v, c_0, ϵ and δ

Output: u

1: Initialization: $c = c_0$.
2: Compute u_0 and p_0, solutions of the direct (1.4) and adjoint (1.7) problems.
3: Compute g_x and g_y given by Equations (1.54).
4: Set $c_i(\mathbf{x}) = \begin{cases} \epsilon & \text{if } g_i(\mathbf{x}) < \delta, \\ c_0 & \text{otherwise.} \end{cases}$, i is x or y
5: Using the Fourier transform and the centred scheme, solve the equation $D_x^{\alpha *} c_x D_x^{\alpha} u + D_y^{\alpha *} c_y D_y^{\alpha} u + u = v$.

The color images are addressed as a vectorial minimization problem, involving the resolution of vectorial problems. The topological asymptotic expansion is still given by Equations (1.8), (1.9) and (1.17), where all functions are vectorial, i.e. the topological gradient is given by the sum on all channels of the corresponding expressions for each channel [18].

1.7.2. Numerical results

Images	Method	1	1.25	1.5	1.75	2
Lena	BF [24]	31.58/843	32.57/865	32.77/870	**32.81**/**870**	32.26/855
$\sigma = 0.06$	Algo 3	32.38/845	32.58/853	32.55/858	32.29/856	32.18/854
Lena	BF [24]	29.19/791	30.26/822	30.49/**830**	**30.50**/**830**	29.74/797
$\sigma = 0.1$	Algo 3	30.19/801	30.29/807	30.26/809	30.11/807	30.02/803
Boat	BF [24]	29.82/792	30.53/811	30.63/814	30.61/813	30.07/795
$\sigma = 0.06$	Algo 3	30.43/803	30.73/815	**30.76**/**820**	30.32/812	30.06/806
Boat	BF [24]	27.35/712	28.11/738	28.22/743	28.12/740	27.54/705
$\sigma = 0.1$	Algo 3	28.17/732	28.38/744	**28.44**/**749**	28.18/744	28.01/739
Peppers	BF [24]	30.81/873	31.95/901	32.11/906	32.27/**909**	31.52/892
$\sigma = 0.06$	Algo 3	31.97/885	32.33/897	**32.38**/903	32.07/905	31.92/903
Peppers	BF [24]	27.94/812	29.12/851	29.35/861	29.46/**864**	28.59/828
$\sigma = 0.1$	Algo 3	28.78/793	29.28/818	**29.56**/841	29.32/845	29.17/842

Table 1.5.: PSNR and SSIM results for image denoising with Bai and Feng algorithm [24] and with Algorithm 3.

Table 1.5 provides a comparison between Bai and Feng algorithm [24] and Algorithm 3, the images are corrupted by an additive Gaussian noise of standard deviation $\sigma = 0.06$ or $\sigma = 0.1$.

Figure 1.20.: From left to right, the noisy image ($\sigma = 15$), the reconstructions with respectively Bai and Feng's algorithm [24] and Algorithm 3, the fractional order α is equal to 1.5. From top to bottom, the Lena image, the Boat image and the Peppers image.

When $\sigma = 0.06$, the diffusion coefficient c_0 and the threshold δ are $c_0 = 1.3$, $\delta = -300$ for the Lena and Peppers images and $c_0 = 1$, $\delta = -300$ for the Boat image. When $\sigma = 0.1$, the diffusion coefficient c_0 and the threshold δ are $c_0 = 2$, $\delta = -300$ for the Boat and Peppers images and $c_0 = 2.5$, $\delta = -400$ for the Lena image. The same diffusion coefficient c_0 and the threshold δ are applied to all values of α to emphasize the fractional order influence. One can remark that for each image, the PSNR and SSIM values have one peak located most generally near 1.5 and 1.75. The results are most significally improved compared with the ones determined with an integer order. Figure 1.20 shows the reconstructions obtained with $\alpha = 1.5$.

In [24] Bai and Feng compare their algorithm with the common Perona-Malik algorithm [94]. The main difference between the two is the utilization of a centred scheme for Bai and Feng. This particularity allows them to perform a better reconstruction with $\alpha = 1$.

Boat ($\sigma = 0.1$)	$BF_{\alpha=1}$	$Algo_{\alpha=1}$	$BF_{\alpha=1.5}$	$Algo_{\alpha=1.5}$	$BF_{\alpha=2}$	$Algo_{\alpha=2}$
PSNR	27.35	28.17	28.22	28.44	27.54	28.01
SSIM	0.712	0.732	0.743	0.794	0.705	0.739
CPU (s)	940	45	1820	70	3800	70

Table 1.6.: CPU for Bai and Feng's algorithm [24] and for our algorithm with $\alpha = 1, 1.5$ and 2.

Table 1.6 compares the CPU relative to restoration processes for $\alpha = 1$, 1.5 and 2, carried out for Bai and Feng's algorithm [24] and for Algorithm 3. All the tests have been conducted on the same computer with MATLAB. The results have a similar quality, but our algorithm needs a shorter computing time.

		Lena		Boat		Peppers	
Method	σ	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$Algo_{\alpha=1.5}$	0.06	32.55	0.858	**30.76**	**0.820**	**32.38**	0.903
	0.1	30.26	0.809	**28.44**	**0.794**	**29.53**	0.839
$BF_{\alpha=1.5}$ [24]	0.06	**32.77**	**0.870**	30.63	0.814	32.11	**0.906**
	0.1	**30.49**	**0.830**	28.22	0.743	29.35	**0.861**
ROF [101]	0.06	31.61	0.841	30.42	0.810	31.48	0.881
	0.1	28.99	0.754	27.91	0.725	28.59	0.800
Weickert [114]	0.06	29.82	0.714	29.22	0.735	29.84	0.759
	0.1	26.00	0.535	25.69	0.573	25.91	0.592

Table 1.7.: Comparison of different denoising algorithms.

A comparison is presented in Table 1.7 with the results of Algorithm 3 with $\alpha = 1.5$ and two denoising algorithms. The first one, the ROF algorithm, originally proposed by Rudin, Osher and Fatemi [101], solves the denoising problem using a method based on total variation. The second one uses an algorithm proposed by Weickert [114]. One can note that the best quality in PSNR and SSIM are obtained with the fractional derivative denoising algorithm.

Figure 1.21 presents the results of the Boat image from Table 1.7. The first line of Figure 1.21 shows the noisy image with $\sigma = 0.06$ and the reconstruction with Algorithm 3. On the second line is shown the reconstruction with Rudin-Osher-Fatemi's algorithm and Weickert's algorithm.

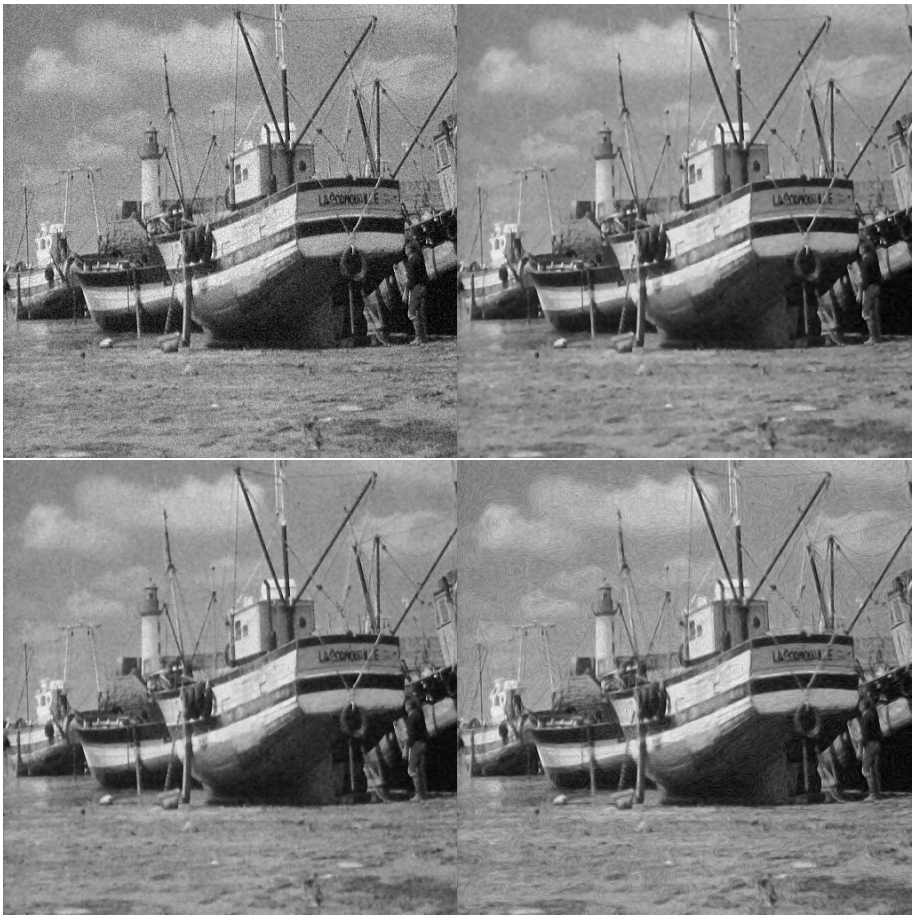


Figure 1.21.: From left to right and up to down, the noisy Boat image, the reconstruction with respectively Algorithm 3, Rudin-Osher-Fatemi's algorithm [101] and Weickert's algorithm [114].

	Lena		Boat		Peppers	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$Algo_{\alpha=1.5}$	**30.12**	**0.863**	**26.84**	0.767	**27.84**	**0.888**
Laplace	28.31	0.854	26.46	**0.771**	26.48	0.868
Total variation	27.48	0.830	25.35	0.735	25.81	0.851

Table 1.8.: Comparison of different inpainting algorithms with 80% unknown pixels.

A comparative study of inpainting reconstruction methods has been performed. In this experiment, the diffusion coefficient c_0 and the threshold δ are respectively equal to 0.1 and -500. Table 1.8 presents the results of different numerical schemes. The first one uses the approximation of a Laplace equation with homogeneous boundary conditions. This kind of method is considered the standard way to solve the inpainting problem [38]. The second one is based on total variation minimization; it uses the split Bregman method improved by Goldstein and Osher [53, 106]. Figure 1.22 presents the results on the Lena image from Table 1.8.

Figure 1.22.: From left to right and up to down, the reconstruction with respectively our algorithm, Laplace equation and total variation regularization [53].

	Lena		Boat		Peppers	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$Algo_{\alpha=1.5}$	32.95	0.906	29.09	0.835	31.76	0.942
Bicubic	29.1	0.879	26.90	0.788	28.10	0.915

Table 1.9.: Comparison with a regular grid of hidden pixels.

In Table 1.9, a comparison with the Bicubic method is performed on a regular grid where one pixel out of four is kept. No noise is added. The fractional order used is $\alpha = 1.5$. Once again, the diffusion coefficient c_0 and the threshold δ are the same for all images. The diffusion coefficient c_0 and the threshold δ are respectively equal to 0.05 and -500. One can ascertain than the fractional diffusion surpasses the classic Bicubic method. Figure 1.23 presents the result for the Peppers image from Table 1.9.

Figure 1.23.: From left to right, original, reduced and restored peppers images.

An inpainting application on color image is presented in Figure 1.24 with a Flowers image (548×548). The data is chosen randomly and is composed of 20% of the pixels from the original image. The fractional order used is $\alpha = 1.5$. The diffusion coefficient c_0 and the threshold δ are respectively equal to 0.3 and -2000. The result obtained is of good quality: 23.62 for the PSNR and 805 for the SSIM. In comparison to the same inpainting problem solved with the Laplacian: 22.95 for the PSNR and 754 for the SSIM.

Demosaicing and image denoising are treated simultaneously with our algorithm in Figure 1.25 with a Parrot image (286×286). The noise is an additive Gaussian noise of standard deviation $\sigma = 0.02$. The fractional order used to perform this reconstruction is $\alpha = 2.5$. The diffusion coefficient c_0 and the threshold δ are respectively equal to 0.5 and -2000. The result obtained is of good quality: 29.52 for the PSNR and 883 for the SSIM.

Figure 1.24.: First line: the original image and the data. Second line: the reconstructions with our algorithm $\alpha = 2.5$ and with the Laplace equation.

Figure 1.25.: From left to right, original Parrot image, noisy data, reconstructed image.

1.8. Conclusion

This work presents a general approach to image processing using topological asymptotic analysis. This approach leads to the proposal of a general reconstruction algorithm (Algorithm 2) that incorporates two novelties: the observation operator L is a general linear operator satisfying mild hypotheses (Hypothesis 1), and an anisotropic diffusion is performed to reconstruct the image. The fact that the observation operator is general allows an extension of the scope of topological asymptotic methods in image processing to new problems: image demosaicing and super-resolution image reconstruction. The introduction of anisotropic diffusion improve the results obtained with existing algorithms using topological gradient in image restoration and inpainting (Algorithm 1). The results obtained were also compared to reference methods, and show better or similar quality in terms of PSNR and SSIM.

A study with Roberto Mecca, an Italian PhD student, leads to the proposal of a general reconstruction algorithm that incorporates two previous works: the fractional derivative implementation from [24] and the edge detection by topological gradient from [27]. Because of their nonlocal property, fractional derivatives provide better reconstructions. Concerning denoising, better results are obtained with an order α which is fractional rather than an integer. For the PSNR and SSIM, the interesting values for the fractional order α seem to be around 1.5 and 1.75. It corroborates the results of Bai and Feng [24]. In Appendix B, it is shown that a fractional order α around 2.5 may also be interesting. Contrary to existing iterative processes with a fractional order, the algorithm presented here is non-iterative. It gives similar results for a shorter computer time and can be used to solve the classes of problems presented in this chapter. The comparison with state-of-the-art methods involving partial differential equations showed better results in terms of quality.

When $\alpha = 1$, the results presented in Table 1.5 are significantly better with our algorithm. However, the superiority is less important when the order is fractional. A future topic in fractional order diffusion would be a topological fractional order gradient. Other higher order differential operators such as bilaplacian could also be a perspective. Another direction could be the combination with patch methods.

2. Extension of the adjoint method

2.1. Introduction

The optimal partitioning of a domain Ω has important real-life applications like shape optimal design, detection of inclusions, image classification and segmentation. Topological derivative methods have been used to solve this kind of problems [49, 105, 107, 75, 56, 57, 13, 14, 77]. These approaches have some drawbacks:

- ➢ The asymptotic topological expansion is not easy to obtain for complex problems.
- ➢ It needs to be adapted for many particular cases like the creation of a hole on the boundary of an existing one or on the original boundary of the domain.
- ➢ We don't know how to calculate the variation of a cost function when a hole is to be filled.
- ➢ In real applications of topology optimization, a finite perturbation is performed and not an infinitesimal one such as an element deletion in a mesh.

Certain issues arise here, as for example the question of how large the topological change should be. In the present chapter, I propose an extension of the adjoint method to overcome these problems.

The problem of optimal partitioning of a domain Ω is equivalent to the problem of looking for an optimal function c which takes a finite number of values $0, 1, \ldots, m-1$. If $m = 2$, c is the characteristic function of an unknown optimal subdomain of Ω. For the sake of simplicity we will limit the study to the case where $m = 2$.

Even if we are dealing with a $0-1$ optimization problem, it is possible to use variational methods. For a cost function

$$\begin{array}{rccl} j: & L^p(\Omega) & \to & \mathbb{R},\ 1 \leq p < \infty \\ & c & \mapsto & j(c), \end{array}$$

the topological asymptotic expansion consists in calculating the variation of j when c switches from 1 to 0 or from 0 to 1 in a small area. Generally, j depends on c via the resolution of a set of partial differential equations. Explicit regularization terms related to the volume, or the measure of the boundary, as well as the mean curvature, could be added to j and a more regular space than L^p could be considered.

It is possible to use differential calculus tools [62] for estimating the variation of j when c switches from 1 to 0 or from 0 to 1 in a small region $\omega_\varepsilon = x_0 + \varepsilon\omega$, where x_0 is a point of Ω, ε is a small scalar, and ω is a given domain. The perturbation $\delta c = \pm\chi_{\omega_\varepsilon}$ obtained when c switches from 1 to 0 or from 0 to 1 in a small region ω_ε, is small in $L^p(\Omega)$ even if not small in magnitude. We will refer to δc as a singular perturbation. If j is differentiable, we have

$$\left| j(c+\delta c) - j(c) - \int_\Omega g_1 \delta c \right| = \mathcal{O}(||\delta c||^2_{L^p}),$$

where $g_1 \in L^q(\Omega), \frac{1}{p} + \frac{1}{q} = 1$ is the gradient of j. Moreover, if g_1 is regular, we obtain the following expansion:

$$|j(c+\delta c) - j(c) - \ |\omega_\varepsilon|\ g_1(x_0)| = \mathcal{O}(|\omega_\varepsilon|^{\frac{2}{p}}) + \circ(|\omega_\varepsilon|). \tag{2.1}$$

There are two cases:

- when $p < 2$, the rest in Equation (2.1) is of higher order and the topological gradient is equal to g_1, the gradient of j. We refer to [62] for more details.
- when $p \geq 2$, the quantity g_1 is the first term of the topological gradient, the "rest" $\mathcal{O}(|\omega_\varepsilon|^{\frac{2}{p}})$ could contribute to the topological gradient by a second term g_2.

The goal of this chapter is to provide a simple, efficient and general way to calculate the second term g_2. In the case $p \geq 2$, the asymptotic expansion (2.1) is technically correct, but it may lead to wrong conclusions.

The adjoint method is the classical way to calculate g_1, the gradient of j, at each point of the domain Ω and at a lower computing cost. In fact, at least for the case $p < 2$, there is no need to calculate the variation of the state with respect to c, because its contribution is of higher order. And when $p \geq 2$, the variation of the solution gives the second term g_2 of the topological gradient. In simple cases, this second term g_2 is quite easy to estimate using the explicit knowledge of the fundamental solution of the problem and there is no need to calculate the unknown variation of the state with respect to the considered perturbation [105, 107, 75, 49, 56, 57, 13, 14, 5, 77]. In this case, the formal application of the chain rule gives only the first term. In some cases, such as the insertion of infinitesimal cracks [13], since $|\omega_\varepsilon| = 0$, the first term is equal to zero and the variation of the cost function is equal to the hidden contribution. Currently, some authors such as Bonnet are interested by asymptotic topological expansions of higher-order [31, 32].

Like we said before, these methods have many limitations. The fundamental solution is not known explicitly for more realistic models. The asymptotic expansion needs to be adapted for too many particular cases like the creation of a hole on the boundary of an existing one or on the original boundary of the domain. The number of particular cases increases if the type of boundary conditions on the hole is considered. In [103], the case

of the Laplace equation with the creation of a hole on a polygonal boundary has been studied. In topology optimization, we need to know how to fill an existing small hole, a problem studied by Guillaume et al. [55].

The state-of-the-art is such that it is possible to write a new paper for each particular case. With this chapter, our goal is to go one step in the direction of a universal tool.

In most contributions, the elementary solution is used to calculate the local variation of the state around the singular perturbation. The key idea of this chapter is to take into account this variation by solving a local perturbed problem defined in a small fixed domain D containing the singular perturbation. The state of the initial uniform problem is imposed as a Dirichlet condition on the boundary of D. The size of the domain D varies with the studied problem.

From the application point of view, this method is non-invasive and can be easily adapted to any solver. It could be used with parallel computing algorithms when multiple evaluations are needed.

From the theoretical point of view, this method could be seen as an improvement of the domain truncation method, where the variation of the Dirichlet to Neumann operator is needed [75, 49, 14]. Here we just need to study the variation of a local solution. Thus, our method could be a tool in theoretical investigations.

This extension of the adjoint method is called the numerical vault for two reasons. The imposed Dirichlet condition on the boundary of D makes our solution very stable and recalls a vault in architecture which is supported by its surrounding walls. Moreover, similarly to the vault, our method is very simple and has great potential for application.

This chapter is quite representative of a general trend in applied mathematics, where asymptotic analysis could help to introduce new numerical methods for solving complex problems with the actual ε which is finite in real-life applications. We can cite asymptotic preserving schemes [39] or numerical homogenization technique using multiscale finite elements [64].

Section 2.2 recalls the adjoint technique and presents the basic concept of the vault method. In Section 2.3, we consider the linear case and present a theoretical justification for the numerical vault as an extension to the adjoint method. Some simple examples are also considered to illustrate the theoretical study. In Section 2.4, we show that the numerical vault is not limited to the estimation of the variation of a cost function, but can be applied to update the solution of a singularly perturbed problem. In Section 2.5, we present some applications of the numerical vault to topological perturbations and we show that the hidden term g_2 is not small in comparison to g_1. Moreover, in the case of a continuous perturbation, the numerical vault allows us to obtain a higher order behavior. We applied our method to continuous material properties and mesh perturbations. An application to elastography is presented in Section 2.6 and in Section 2.7, the numerical vault is applied to update the solution in the case of an image restoration problem.

2.2. The adjoint method

2.2.1. The adjoint technique

We first recall the adjoint method in a formal way. Consider the following steady state equation

$$F(c, u) = 0 \text{ in } \Omega, \tag{2.2}$$

where c is a distributed parameter in a domain Ω. The aim is to minimize a cost function $j(c) := J(u_c)$ where u_c is the solution of Equation (2.2) for a given c.

Let us suppose that every term is differentiable. We are considering a perturbation δc of the parameter c.

Since c is a distributed parameter, its discretization leads to a huge vector. A fast gradient computation is therefore of high importance.

Equation (2.2) can be seen as a constraint, and as a consequence, the Lagrangian is considered:

$$\mathcal{L}(c, u, p) = J(u) + (F(c, u), p),$$

where p is a Lagrange multiplier and $(\cdot, \cdot)$ denotes the scalar product in a well-chosen Hilbert space.

To compute the derivative of j, one can remark that $j(c) = \mathcal{L}(c, u_c, p)$ for all c, if u_c is the solution of Equation (2.2). The derivative of j is then equal to the derivative of $\mathcal{L}$ with respect to c:

$$d_c j(c)\delta c = \partial_c \mathcal{L}(c, u_c, p)\delta c + \partial_u \mathcal{L}(c, u_c, p)\partial_c u\, \delta c.$$

All these terms can be calculated easily, except $\partial_c u\, \delta c$, the solution of the linearized problem:

$$\partial_u F(c, u_c)(\partial_c u\, \delta c) = -\partial_c F(c, u_c)\, \delta c.$$

To avoid the resolution of this equation for each δc, the term $\partial_u \mathcal{L}(c, u_c, p)$ is cancelled by solving the following adjoint equation in p. Let p_c be the solution of the adjoint equation:

$$\partial_u F(c, u_c)^T p_c = -\partial_u J^T.$$

So the derivative of j is explicitly given by

$$d_c j(c)\delta c = \partial_c \mathcal{L}(c, u_c, p_c).$$

This method works only when the problem is differentiable and δc is an admissible perturbation. Several counter-examples, where one of these conditions is not satisfied, are given in [62].

2.2.2. Generalized adjoint technique

Note that if the Lagrangians $\mathcal{L}(c+\delta c,\dots,\dots)$ and $\mathcal{L}(c,\dots,\dots)$ are defined on the same space, we have

$$j(c+\delta c)-j(c)=\mathcal{L}(c+\delta c,u_{c+\delta c},p_c)-\mathcal{L}(c,u_c,p_c),$$

it can be split in two terms:

$$j(c+\delta c)-j(c)=(\mathcal{L}(c+\delta c,u_{c+\delta c},p_c)-\mathcal{L}(c+\delta c,u_c,p_c))+(\mathcal{L}(c+\delta c,u_c,p_c)-\mathcal{L}(c,u_c,p_c)).$$

In the case of a regular perturbation δc, the second term gives the main variation and the first term is of higher order. In the case of a singular perturbation, the first term is of the same order as the second one and cannot be ignored. Then the variation of u_c has to be estimated.

In the topology optimization papers [13, 14, 19, 49, 56, 57, 75], the local variation of u_c can be expressed thanks to the elementary solution of the problem to solve. This solution is singular at x_0 and gives the local variation of u_c around x_0. In general, it is not always simple to get an explicit expression of the elementary solution. This is particularly true for problems with discontinuous coefficients, if the discontinuity is located at x_0.

Due to this difficulty, a numerical estimation of this variation is presented. The basic idea of the numerical vault is to update the solution u_c by solving a local problem defined in a small domain around x_0.

If the Lagrangians $\mathcal{L}(c+\delta c,\dots,\dots)$ and $\mathcal{L}(c,\dots,\dots)$ are not defined on the same space, we have

$$j(c+\delta c)-j(c)=\mathcal{L}(c+\delta c,u_{c+\delta c},p_{c+\delta c})-\mathcal{L}(c,u_c,p_c).$$

In this case, the direct solution u_c and also the adjoint p_c are updated with the numerical vault.

2.3. Estimation of the variation of a cost function with the numerical vault

For the sake of simplicity, only the linear case is studied. Consider the variational problem depending on a parameter ε

$$a^\varepsilon(u,v)=\ell^\varepsilon(v)\quad\forall v\in\mathcal{V}^\varepsilon, \tag{2.3}$$

where $\mathcal{V}^\varepsilon$ is a Hilbert space, a^ε is a bilinear, continuous and coercive form and ℓ^ε is a linear and continuous form. Typically, $\mathcal{V}^\varepsilon$ is such that $\mathrm{H}^1_0\subset\mathcal{V}^\varepsilon\subset\mathrm{H}^1$.

In this study, a singular perturbation means a finite variation located in an infinitesimal area: the size of the perturbation goes to zero when ε goes to zero. Depending on the problem, ε can be the thickness of a coating layer around a variety of dimension $\leq n-1$ in the space of dimension n. In the case of a hole, ε can be the diameter of a tube around a curve in the space of dimension 3. And we do not forget the classical case, where ε is the diameter of a hole created around a point x_0.

Let u^ε be the solution of the problem (2.3) and u^0 be the solution of the initial problem without any perturbation at $\varepsilon = 0$. The objective is to calculate the variation, with respect to ε, of

$$j(\varepsilon) := J^\varepsilon(u^\varepsilon).$$

The cost function J^ε is of class C^1, the adjoint problem associated to the problem (2.3) and the cost function J^ε is defined as follows

$$a^\varepsilon(w, p^\varepsilon) = -\partial_u J^\varepsilon(u^\varepsilon)w \quad \forall w \in \mathcal{V}^\varepsilon, \tag{2.4}$$

where p^ε is the solution of this problem.

Suppose that a^ε, ℓ^ε and J^ε are integrals over a domain Ω.

The domain Ω is split into two parts, a part D containing the perturbation, and its complementary $\Omega_0 = \Omega \backslash \overline{D}$. The perturbation can be located on the boundary of Ω. For this reason, the domain $\overline{D}$ is not necessarily included in the domain Ω.

The forms a^ε, ℓ^ε, and the cost function J^ε are decomposed in the following way:

- $a^\varepsilon = a_{\Omega_0} + a^\varepsilon_D$,
- $\ell^\varepsilon = \ell_{\Omega_0} + \ell^\varepsilon_D$,
- $J^\varepsilon = J_{\Omega_0} + J^\varepsilon_D$,

where a_{Ω_0}, l_{Ω_0} et J_{Ω_0} are independent of ε.

The constants α and M are positive constants used in respectively the minoration and the majoration of the bilinear forms.

Similarly, we consider

- $\mathcal{V}_{\Omega_0}$, the space consisting of functions of $\mathcal{V}^\varepsilon$ and $\mathcal{V}^0$ restricted to Ω_0,
- $\mathcal{V}^\varepsilon_D$, the space consisting of functions of $\mathcal{V}^\varepsilon$ restricted to D,
- $\mathcal{V}^0_D$, the space consisting of functions of $\mathcal{V}^0$ restricted to D,
- $\mathcal{V}^\varepsilon_{D,0}$, the subspace of $\mathcal{V}^\varepsilon_D$, with null trace on ∂D,
- $\mathcal{V}_{\partial D}$, the space of traces on ∂D of $\mathcal{V}_{\Omega_0}$.

2.3.1. Updating the direct solution

In this section, we assume that $\mathcal{V}^0 \subset \mathcal{V}^\varepsilon$. When the perturbation consists in inserting a hole in the domain with a Dirichlet condition on the boundary of the hole, this condition does not hold. This case will be considered in Section 2.3.2.

Let us consider u_D^ε, the local update of u^0:

$$\left\{\begin{array}{rcl} \text{Find } u_D^\varepsilon \in & \mathcal{V}_D^\varepsilon & \text{solution of} \\ a_D^\varepsilon(u_D^\varepsilon, v) & = & \ell_D^\varepsilon(v) \quad \forall v \in \mathcal{V}_{D,0}^\varepsilon, \\ u_D^\varepsilon & = & u^0 \text{ on } \partial D. \end{array}\right. \tag{2.5}$$

and $\tilde{u}^\varepsilon$ the update of u^0 is given by:

$$\tilde{u}^\varepsilon = \left\{\begin{array}{ll} u_D^\varepsilon & \text{in } D, \\ u^0 & \text{in } \Omega_0. \end{array}\right. \tag{2.6}$$

We assume the following hypothesis.

Hypothesis 2. *There exist three positive constants η, C and C_u independent of ε and a positive real valued function f defined on $\mathbb{R}_+$ such that*

$$\lim_{\varepsilon \to 0} f(\varepsilon) = 0,$$

$$\|J^\varepsilon(v) - J^\varepsilon(u) - \partial_u J^\varepsilon(u)(v-u)\|_{\mathcal{V}^\varepsilon} \leq C\|v-u\|_{\mathcal{V}^\varepsilon}^2, \ \forall v, u \in B(u^0, \eta),$$

$$\|u^\varepsilon - u^0\|_{\mathcal{V}_{\Omega_0}} \leq C_u f(\varepsilon),$$

$$\lim_{\varepsilon \to 0} \|p^\varepsilon - p^0\|_{\mathcal{V}^\varepsilon} = 0.$$

Proposition 1. *Under Hypothesis 2, we have*

$$\|u^\varepsilon - \tilde{u}^\varepsilon\|_{\mathcal{V}^\varepsilon} = \mathcal{O}(f(\varepsilon)).$$

Proof. Hypothesis 2 tells us that

$$\|u^\varepsilon - u^0\|_{\mathcal{V}_{\Omega_0}} = \mathcal{O}(f(\varepsilon)).$$

The trace properties allows us to write:

$$\|u^\varepsilon - u^0\|_{\mathcal{V}_{\partial D}} = \mathcal{O}(f(\varepsilon)).$$

We can write it under the following form

$$\|u^\varepsilon - u^0\|_{\mathcal{V}_{\partial D}} = \min_{\phi = u^\varepsilon - u^0 \text{ on } \partial D} \|\phi\|_{\mathcal{V}_D^\varepsilon}.$$

As a consequence, we define $\varphi \in \mathcal{V}_D^\varepsilon$ as the minimum of $\|\varphi\|_{\mathcal{V}_D^\varepsilon}$ under the constraint $\varphi = u^\varepsilon - u^0$ on ∂D, then

$$\|\varphi\|_{\mathcal{V}_D^\varepsilon} = \|u^\varepsilon - u^0\|_{\mathcal{V}_{\partial D}} = \mathcal{O}(f(\varepsilon)).$$

We have $\varphi - (u^\varepsilon - u_D^\varepsilon) = 0$ on ∂D, we can use the coercivity of a_D^ε:

$$\begin{aligned}
&\alpha\|\varphi - (u^\varepsilon - u_D^\varepsilon)\|^2_{\mathcal{V}_D^\varepsilon} \\
\leq& a_D^\varepsilon(\varphi - (u^\varepsilon - u_D^\varepsilon), \varphi - (u^\varepsilon - u_D^\varepsilon)) \\
\leq& a_D^\varepsilon(\varphi, \varphi - (u^\varepsilon - u_D^\varepsilon)) - a_D^\varepsilon(u^\varepsilon, \varphi - (u^\varepsilon - u_D^\varepsilon)) + a_D^\varepsilon(u_D^\varepsilon, \varphi - (u^\varepsilon - u_D^\varepsilon)).
\end{aligned}$$

and the continuity of the bilinear form a_D^ε:

$$\begin{aligned}
&\alpha\|\varphi - (u^\varepsilon - u_D^\varepsilon)\|^2_{\mathcal{V}_D^\varepsilon} \\
\leq& M\|\varphi\|_{\mathcal{V}_D}\|\varphi - (u^\varepsilon - u_D^\varepsilon)\|_{\mathcal{V}_D} - \ell_D^\varepsilon(\varphi - (u^\varepsilon - u_D^\varepsilon)) + \ell_D^\varepsilon(\varphi - (u^\varepsilon - u_D^\varepsilon)) \\
\leq& M\|u^\varepsilon - u^0\|_{\mathcal{V}_{\partial D}}\|\varphi - (u^\varepsilon - u_D^\varepsilon)\|_{\mathcal{V}_D^\varepsilon}.
\end{aligned}$$

Using $\|\varphi\|_{\mathcal{V}_D^\varepsilon} = \mathcal{O}(f(\varepsilon))$, $\|\varphi - (u^\varepsilon - u_D^\varepsilon)\|_{\mathcal{V}_D^\varepsilon} = \mathcal{O}(f(\varepsilon))$ and the triangular inequality, we obtain:

$$\|u^\varepsilon - u_D^\varepsilon\|_{\mathcal{V}_D^\varepsilon} = \mathcal{O}(f(\varepsilon)).$$

With the equality

$$\|u^\varepsilon - \tilde{u}^\varepsilon\|^2_{\mathcal{V}^\varepsilon} = \|u^\varepsilon - u^0\|^2_{\mathcal{V}_{\Omega_0}} + \|u^\varepsilon - u_D^\varepsilon\|^2_{\mathcal{V}_D^\varepsilon},$$

we obtain the final result. □

Theorem 2.3.1. *Under Hypothesis 2, we have*

$$j(\varepsilon) - j(0) = \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, p^0) - \mathcal{L}^0(u^0, p^0) + o\left(f(\varepsilon)\right).$$

According to litterature[49, 75, 56, 57, 13, 14], the difference $\mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, p^0) - \mathcal{L}^0(u^0, p^0)$ is of order $f(\varepsilon)$.

Proof: We just split this variation into two terms

$$j(\varepsilon) - j(0) = \mathcal{L}^\varepsilon(u^\varepsilon, p^0) - \mathcal{L}^0(u^0, p^0) = \left[\mathcal{L}^\varepsilon(u^\varepsilon, p^0) - \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, p^0)\right] + \left[\mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, p^0) - \mathcal{L}^0(u^0, p^0)\right].$$

and prove that

$$\mathcal{L}^\varepsilon(u^\varepsilon, p^0) - \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, p^0) = o\left(f(\varepsilon)\right).$$

Using Hypothesis 2, we have

$$\begin{aligned}
\mathcal{L}^\varepsilon(u^\varepsilon, p^0) - \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, p^0) &= J^\varepsilon(u^\varepsilon) - J^\varepsilon(\tilde{u}^\varepsilon) + a^\varepsilon(u^\varepsilon, p^0) - a^\varepsilon(\tilde{u}^\varepsilon, p^0) \\
&= \partial_u J^\varepsilon(u^\varepsilon)(u^\varepsilon - \tilde{u}^\varepsilon) + o\left(f(\varepsilon)\right) + a^\varepsilon(u^\varepsilon - \tilde{u}^\varepsilon, p^0),
\end{aligned}$$

and with Equation (2.4), we obtain

$$\begin{aligned}\mathcal{L}^\varepsilon(u^\varepsilon,p^0)-\mathcal{L}^\varepsilon(\tilde{u}^\varepsilon,p^0) &= -a^\varepsilon(u^\varepsilon-\tilde{u}^\varepsilon,p^\varepsilon)+a^\varepsilon(u^\varepsilon-\tilde{u}^\varepsilon,p^0)+o\left(f(\varepsilon)\right)\\ &= a^\varepsilon(u^\varepsilon-\tilde{u}^\varepsilon,p^0-p^\varepsilon)+o\left(f(\varepsilon)\right).\end{aligned}$$

Using the continuity of a^ε, Hypothesis 2 and Proposition 1, we obtain the final result.

□

Remark 2. Generally, for topological perturbations $\|p^\varepsilon-p^0\|_{\mathcal{V}^\varepsilon}=\mathcal{O}\left((f(\varepsilon))^{\frac{1}{2}}\right)$ and $j(\varepsilon)-j(0)=\mathcal{L}^\varepsilon(\tilde{u}^\varepsilon,p^0)-\mathcal{L}^0(u^0,p^0)+\mathcal{O}\left((f(\varepsilon))^{\frac{3}{2}}\right)$ [75]. We will see in the following section that if we update locally u and p, the rest could be of order $\mathcal{O}\left(f(\varepsilon)^2\right)$.

2.3.2. Updating the direct and adjoint solutions

In this section $\mathcal{V}^0$ is not necessary a sub-space of $\mathcal{V}^\varepsilon$. This is the case when the perturbation is a hole of radius ε with a Dirichlet boundary condition on its boundary. The definition of $\tilde{u}^\varepsilon$ remains unchanged and we update p^0 in the same way.

Let us consider p_D^ε the local update of p^0

$$\left\{\begin{array}{lll}\text{Find } p_D^\varepsilon\in\mathcal{V}_D^\varepsilon \text{ solution of} & & \\ a_D^\varepsilon(w,p_D^\varepsilon) = -\partial_u J_D^\varepsilon(u_D^\varepsilon)w, & \forall w\in\mathcal{V}_{D,0}^\varepsilon, & \\ p_D^\varepsilon = p^0 & \text{on } \partial D. & \end{array}\right. \tag{2.7}$$

and $\tilde{p}^\varepsilon$ the update of p^0 is given by:

$$\tilde{p}^\varepsilon=\left\{\begin{array}{ll} p_D^\varepsilon & \text{in } D,\\ p^0 & \text{in } \Omega_0.\end{array}\right. \tag{2.8}$$

We assume the following hypothesis.

Hypothesis 3. *There exist four positive constants η, C, C_u and C_p independent of ε and a positive real valued function f defined on $\mathbb{R}_+$ such that*

$$\lim_{\varepsilon\to 0} f(\varepsilon)=0,$$

$$\begin{gathered}\|J^\varepsilon(v)-J^\varepsilon(u)-\partial_u J^\varepsilon(u)(v-u)\|_{\mathcal{V}^\varepsilon}\leq C\|v-u\|_{\mathcal{V}^\varepsilon}^2,\ \forall v,u\in B(u^0,\eta),\\ \|u^\varepsilon-u^0\|_{\mathcal{V}_{\Omega_0}}\leq C_u f(\varepsilon),\\ \|p^\varepsilon-p^0\|_{\mathcal{V}_{\Omega_0}}\leq C_p f(\varepsilon).\end{gathered}$$

Proposition 2. *Under Hypothesis 3, we have*

$$\|p^\varepsilon - \tilde{p}^\varepsilon\|_{\mathcal{V}^\varepsilon} = \mathcal{O}(f(\varepsilon)).$$

Proof. Hypothesis 3 tells us that

$$\|p^\varepsilon - p^0\|_{\mathcal{V}_{\Omega_0}} = \mathcal{O}(f(\varepsilon)).$$

The trace properties allows us to write:

$$\|p^\varepsilon - p^0\|_{\mathcal{V}_{\partial D}} = \mathcal{O}(f(\varepsilon)).$$

We can write it under the following form

$$\|p^\varepsilon - p^0\|_{\mathcal{V}_{\partial D}} = \min_{\phi = p^\varepsilon - p^0 \text{ on } \partial D} \|\phi\|_{\mathcal{V}_D^\varepsilon}.$$

As a consequence, we define $\varphi \in \mathcal{V}_D^\varepsilon$ as the minimum of $\|\varphi\|_{\mathcal{V}_D^\varepsilon}$ under the constraint $\varphi = p^\varepsilon - p^0$ on ∂D, then

$$\|\varphi\|_{\mathcal{V}_D^\varepsilon} = \|p^\varepsilon - p^0\|_{\mathcal{V}_{\partial D}} = \mathcal{O}(f(\varepsilon)).$$

We have $\varphi - (p^\varepsilon - p_D^\varepsilon) = 0$ on ∂D, we can use the coercivity of a_D^ε:

$$\begin{aligned}
&\alpha\|\varphi - (p^\varepsilon - p_D^\varepsilon)\|^2_{\mathcal{V}_D^\varepsilon} \\
\leq& a_D^\varepsilon(\varphi - (p^\varepsilon - p_D^\varepsilon), \varphi - (p^\varepsilon - p_D^\varepsilon)) \\
\leq& a_D^\varepsilon(\varphi - (p^\varepsilon - p_D^\varepsilon), \varphi) - a_D^\varepsilon(\varphi - (p^\varepsilon - p_D^\varepsilon), p^\varepsilon) + a_D^\varepsilon(\varphi - (p^\varepsilon - p_D^\varepsilon), p_D^\varepsilon).
\end{aligned}$$

The Equations 2.4 and 2.7 give us:

$$\begin{aligned}
&\alpha\|\varphi - (p^\varepsilon - p_D^\varepsilon)\|^2_{\mathcal{V}_D^\varepsilon} \\
\leq& a_D^\varepsilon(\varphi - (p^\varepsilon - p_D^\varepsilon), \varphi) + \partial_u J_D^\varepsilon(u^\varepsilon)\,(\varphi - (p^\varepsilon - p_D^\varepsilon)) - \partial_u J_D^\varepsilon(u_D^\varepsilon)\,(\varphi - (p^\varepsilon - p_D^\varepsilon)) \\
\leq& a_D^\varepsilon(\varphi - (p^\varepsilon - p_D^\varepsilon), \varphi) + \partial_{uu} J_D^\varepsilon(u^\varepsilon)\,(\varphi - (p^\varepsilon - p_D^\varepsilon), u^\varepsilon - u_D^\varepsilon) \\
&\qquad + \|\varphi - (p^\varepsilon - p_D^\varepsilon)\|_{\mathcal{V}_D^\varepsilon} o(f(\varepsilon)).
\end{aligned}$$

The continuity of the bilinear form a_D^ε and the continuity of the linear form $\partial_u J_D^\varepsilon$ lead to the following result:

$$\begin{aligned}
\alpha\|\varphi - (p^\varepsilon - p_D^\varepsilon)\|_{\mathcal{V}_D^\varepsilon} \leq& M\left(\|\varphi\|_{\mathcal{V}_D^\varepsilon} + \|u^\varepsilon - u_D^\varepsilon\|_{\mathcal{V}_D^\varepsilon} + o(f(\varepsilon))\right). \\
\leq& M\left(\|p^\varepsilon - p^0\|_{\mathcal{V}_{\partial D}} + \|u^\varepsilon - u_D^\varepsilon\|_{\mathcal{V}_D^\varepsilon} + o(f(\varepsilon))\right).
\end{aligned}$$

Thanks to $\|p^\varepsilon - p^0\|_{\mathcal{V}_{\partial D}} = \mathcal{O}(f(\varepsilon))$ and Proposition 1 $\|u^\varepsilon - u_D^\varepsilon\|_{\mathcal{V}_D^\varepsilon} = \mathcal{O}(f(\varepsilon))$, we have $\|\varphi - (p^\varepsilon - p_D^\varepsilon)\|_{\mathcal{V}_D^\varepsilon} = \mathcal{O}(f(\varepsilon))$.

Using $\|\varphi\|_{\mathcal{V}_D^\varepsilon} = \mathcal{O}(f(\varepsilon))$, $\|\varphi - (p^\varepsilon - p_D^\varepsilon)\|_{\mathcal{V}_D^\varepsilon} = \mathcal{O}(f(\varepsilon))$ and the triangular inequality, we obtain:

$$\|p^\varepsilon - p_D^\varepsilon\|_{\mathcal{V}_D^\varepsilon} = \mathcal{O}(f(\varepsilon)).$$

With the equality

$$\|p^\varepsilon - \tilde{p}^\varepsilon\|^2_{\mathcal{V}^\varepsilon} = \|p^\varepsilon - p^0\|^2_{\mathcal{V}_{\Omega_0}} + \|p^\varepsilon - p_D^\varepsilon\|^2_{\mathcal{V}_D^\varepsilon},$$

we obtain the final result. □

Theorem 2.3.2. *Under Hypothesis 3, we have*

$$j(\varepsilon) - j(0) = \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, \tilde{p}^\varepsilon) - \mathcal{L}^0(u^0, p^0) + \mathcal{O}(f(\varepsilon)^2).$$

According to litterature[49, 75, 56, 57, 13, 14], the difference $\mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, \tilde{p}^\varepsilon) - \mathcal{L}^0(u^0, p^0)$ is of order $f(\varepsilon)$.

Proof: We just split this variation into two terms

$$\begin{aligned} j(\varepsilon) - j(0) &= \mathcal{L}^\varepsilon(u^\varepsilon, p^\varepsilon) - \mathcal{L}^0(u^0, p^0) \\ &= [\mathcal{L}^\varepsilon(u^\varepsilon, p^\varepsilon) - \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, p^\varepsilon)] + \left[\mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, p^\varepsilon) - \mathcal{L}^0(u^0, p^0)\right]. \end{aligned}$$

and prove that

$$\mathcal{L}^\varepsilon(u^\varepsilon, p^\varepsilon) - \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, \tilde{p}^\varepsilon) = \mathcal{O}(f(\varepsilon)^2).$$

Using Hypothesis 3, we have

$$\begin{aligned} &\mathcal{L}^\varepsilon(u^\varepsilon, p^\varepsilon) - \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, \tilde{p}^\varepsilon) \\ =&J^\varepsilon(u^\varepsilon) - J^\varepsilon(\tilde{u}^\varepsilon) + a^\varepsilon(u^\varepsilon, p^\varepsilon) - a^\varepsilon(\tilde{u}^\varepsilon, \tilde{p}^\varepsilon) - \ell^\varepsilon(p^\varepsilon) + \ell^\varepsilon(\tilde{p}^\varepsilon) \\ =&\partial_u J^\varepsilon(u^\varepsilon)(u^\varepsilon - \tilde{u}^\varepsilon) + \mathcal{O}(f(\varepsilon)^2) + a^\varepsilon(u^\varepsilon - \tilde{u}^\varepsilon, p^\varepsilon) + a^\varepsilon(\tilde{u}^\varepsilon, p^\varepsilon - \tilde{p}^\varepsilon) - \ell^\varepsilon(p^\varepsilon - \tilde{p}^\varepsilon) \end{aligned}$$

and with Equations (2.3) and (2.4), we obtain

$$\begin{aligned} \mathcal{L}^\varepsilon(u^\varepsilon, p^\varepsilon) - \mathcal{L}^\varepsilon(\tilde{u}^\varepsilon, \tilde{p}^\varepsilon) &= a^\varepsilon(\tilde{u}^\varepsilon, p^\varepsilon - \tilde{p}^\varepsilon) - \ell^\varepsilon(p^\varepsilon - \tilde{p}^\varepsilon) + \mathcal{O}(f(\varepsilon)^2) \\ &= a^\varepsilon(\tilde{u}^\varepsilon - u^\varepsilon, p^\varepsilon - \tilde{p}^\varepsilon) + \mathcal{O}(f(\varepsilon)^2). \end{aligned}$$

Using the continuity of a^ε, Propositions 1 and 2, we obtain the final result. □

Remark 3. Updating p is not necessary when $\mathcal{V}^0 \subset \mathcal{V}^\varepsilon$, but according to Theorem 2.3.2, we obtain a better estimation of the variation of j when p is updated.

In the case of a Dirichlet condition on the boundary of the hole, $\mathcal{V}^0$ is not a subspace of $\mathcal{V}^\varepsilon$ and we need to update the adjoint.

2.3.3. Examples

Examples in literature

The most important hypothesis is

$$\|u^\varepsilon - u^0\|_{\mathcal{V}_{\Omega_0}} = \mathcal{O}(f(\varepsilon)). \tag{2.9}$$

This assumption (2.9) is generally satisfied. See

- ➢ the elasticity case [49],
- ➢ the Poisson equation [56],
- ➢ the Navier-Stokes equation [57],
- ➢ the Helmholtz equation [14].

These results have been obtained in the frame of the domain truncation method.

The capacity case

This section presents an application of the previous results to the capacity problem. Let $\Omega \in \mathbb{R}^2$ be a domain.

$$\begin{cases} \Delta u^0 = 0 & \text{in } \Omega, \\ u^0 = 1 & \text{on } \partial\Omega. \end{cases} \tag{2.10}$$

This equation implies that $u^0 = 1$ in Ω is solution.

The cost function is the following

$$J^\varepsilon(v) = \frac{1}{2}\int_{\Omega_\varepsilon} |\nabla v|^2 \, dx, \quad \forall v \in V. \tag{2.11}$$

Let $B(x_0, \varepsilon)$ be a hole in the domain Ω, the perturbed domain is Ω_ε.

$$\begin{cases} \Delta u^\varepsilon = 0 & \text{in } \Omega_\varepsilon, \\ u^\varepsilon = 0 & \text{on } \partial B(x_0, \varepsilon), \\ u^\varepsilon = 1 & \text{on } \partial\Omega. \end{cases} \tag{2.12}$$

Let D be equal to $B(x_0, R)\backslash B(x_0, \varepsilon)$ with $R > \varepsilon$ and $B(x, R) \subset \Omega$.

The update of u^0 is defined by

$$\tilde{u}^\varepsilon = \begin{cases} u_D^\varepsilon & \text{in } D, \\ u^0 & \text{in } \Omega\backslash D. \end{cases} \tag{2.13}$$

where u_D^ε is the solution of the following equation

$$\begin{cases} \Delta u_D^\varepsilon = 0 & \text{in } D, \\ u_D^\varepsilon = 0 & \text{on } \partial B(x_0, \varepsilon), \\ u_D^\varepsilon = 1 & \text{on } \partial B(x_0, R). \end{cases} \tag{2.14}$$

The hypothesis $\|u^\varepsilon - u^0\|_{\mathcal{V}_{\Omega_0}} = \mathcal{O}(f(\varepsilon))$ is satisfied [75].

The elementary solution of the Laplace equation in two dimensions is $-\frac{\ln(r)}{2\pi}$ where $r = |x - x_0|$.

The update is

$$\tilde{u}^\varepsilon = \begin{cases} \frac{\ln(|x-x_0|)-\ln(\varepsilon)}{\ln(R)-\ln(\varepsilon)} & \text{in } D, \\ 1 & \text{in } \Omega\backslash D. \end{cases} \tag{2.15}$$

Let Γ be the outer boundary of the domain D and n the normal oriented towards the exterior on the boundary. With the approximation $\tilde{u}^\varepsilon$, the cost function (2.11) can be written as follows

$$J^\varepsilon(\tilde{u}^\varepsilon) = \pi \int_\Gamma \frac{\partial \tilde{u}^\varepsilon}{\partial n} \tilde{u}^\varepsilon r \, dr. \tag{2.16}$$

Taking into account Theorem 2.3.1 we obtain

$$j(\varepsilon) - j(0) = J^\varepsilon(u^\varepsilon) - J^0(u^0) = J^\varepsilon(\tilde{u}^\varepsilon) - J^0(u^0) + o\left(f(\varepsilon)\right),$$

and given the update definition (2.15), we have

$$\begin{aligned} & J^\varepsilon(\tilde{u}^\varepsilon) - J^0(u^0) = \pi \int_\varepsilon^R |\nabla u^\varepsilon|^2 r \, dr \\ \Leftrightarrow\ & J^\varepsilon(\tilde{u}^\varepsilon) - J^0(u^0) = \frac{\pi}{\left(\ln(R) - \ln(\varepsilon)\right)^2} \int_{r=R} \frac{1}{r} \left(\ln(r) - \ln(\varepsilon)\right) r \, dr \\ \Leftrightarrow\ & J^\varepsilon(\tilde{u}^\varepsilon) - J^0(u^0) = \frac{\pi}{\left(\ln(R) - \ln(\varepsilon)\right)}. \end{aligned}$$

The final result is

$$j(\varepsilon) - j(0) = J^\varepsilon(u^\varepsilon) - J^0(u^0) = \frac{-\pi}{\ln(\varepsilon)} + o\left(\frac{1}{\ln(\varepsilon)}\right).$$

This result is exactly the same as the one obtained in [75].

2.4. Approximation of a perturbed solution

In this section, we assume that $\mathcal{V}^0 \subset \mathcal{V}^\varepsilon$. The aim of the previous section was to estimate the variation of a cost function with respect to a local perturbation. In this part, the vault method is used to construct an approximation of the perturbed solution u.

We assume the following hypothesis.

Hypothesis 4. *There exists a constant C_u independent of ε and a positive real valued function f defined on $\mathbb{R}_+$ such that*

$$\lim_{\varepsilon \to 0} f(\varepsilon) = 0,$$
$$\|u^\varepsilon - u^0\|_{\mathcal{V}_{\Omega_0}} \leq C_u f(\varepsilon),$$
$$\|a_D^\varepsilon - a_D^0\|_{\mathcal{L}^2(\mathcal{V}_D^\varepsilon)} = \mathcal{O}(f(\varepsilon)).$$

The norm $\|.\|_{\mathcal{L}^2(\mathcal{V}_D^\varepsilon)}$ means the norm of the bilinear form.

Proposition 3. *Under Hypothesis 4, we have*

$$\|u^\varepsilon - \tilde{u}^\varepsilon\|_{\mathcal{V}^\varepsilon} = \mathcal{O}(f(\varepsilon)),$$
$$\|a^\varepsilon - a^0\|_{\mathcal{L}^2(\mathcal{V}^\varepsilon)} = \mathcal{O}(f(\varepsilon)).$$

Proof. Thanks to Proposition 1, we have

$$\|u^\varepsilon - \tilde{u}^\varepsilon\|_{\mathcal{V}^\varepsilon} = \mathcal{O}(f(\varepsilon)).$$

As $a^0 = a_{\Omega_0} + a_D^0$ and $a^\varepsilon = a_{\Omega_0} + a_D^\varepsilon$, we have

$$\|a^\varepsilon - a^0\|_{\mathcal{L}^2(\mathcal{V}^\varepsilon)} = \mathcal{O}(f(\varepsilon)).$$

□

A singular local perturbation has two effects: a local high frequency effect and a global low frequency effect. The local effect is caught by $\tilde{u}^\varepsilon$. In the previous section, the global effect is taken into account by the adjoint. In this section, we are considering the variation of the solution, the adjoint cannot be used, and we need to solve an auxiliary problem in order to calculate the global low frequency variation.

Let us consider δu^ε the solution of

$$\begin{cases} \text{Find } \delta u^\varepsilon \in \mathcal{V}^0 \text{ solution of} \\ a^0(\delta u^\varepsilon, v) = a^\varepsilon(u^\varepsilon - \tilde{u}^\varepsilon, v), \quad \forall v \in \mathcal{V}^0. \end{cases} \tag{2.17}$$

The approximation of u^ε with the numerical vault is noted $\tilde{\tilde{u}}^\varepsilon$ and defined as follows

$$\tilde{\tilde{u}}^\varepsilon = \tilde{u}^\varepsilon + \delta u^\varepsilon. \tag{2.18}$$

Proposition 4. *Under Hypothesis 4, we have*

$$\|\delta u^\varepsilon\|_{\mathcal{V}^0} = \mathcal{O}(f(\varepsilon)).$$

Proof. Thanks to the coercivity of a^0, we have

$$\alpha\|\delta u^\varepsilon\|^2_{\mathcal{V}^0} \leq a^0\left(\delta u^\varepsilon, \delta u^\varepsilon\right) = a^\varepsilon\left(u^\varepsilon - \tilde{u}^\varepsilon, \delta u^\varepsilon\right).$$

Thanks to the continuity of a^ε, we have

$$a^\varepsilon\left(u^\varepsilon - \tilde{u}^\varepsilon, \delta u^\varepsilon\right) \leq M\|u^\varepsilon - \tilde{u}^\varepsilon\|_{\mathcal{V}^\varepsilon}\|\delta u^\varepsilon\|_{\mathcal{V}^0}.$$

Hypothesis 4 and Proposition 3 allow us to conclude. □

Theorem 2.4.1. *Under Hypothesis 4 and with the definition of $\tilde{\tilde{u}}^\varepsilon$, we have*

$$\left\|u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right\|_{\mathcal{V}^\varepsilon} = \mathcal{O}((f(\varepsilon))^2).$$

Proof. Thanks to the coercivity of a^ε, we have

$$\alpha\left\|u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right\|^2_{\mathcal{V}^\varepsilon} \leq a^\varepsilon\left(u^\varepsilon - \tilde{\tilde{u}}^\varepsilon, u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right) = a^\varepsilon\left(u^\varepsilon - \tilde{u}^\varepsilon, u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right) - a^\varepsilon\left(\delta u^\varepsilon, u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right).$$

Thanks to the definition of $\tilde{\tilde{u}}^\varepsilon$, we have

$$\alpha\left\|u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right\|^2_{\mathcal{V}^\varepsilon} \leq a^0\left(\delta u^\varepsilon, u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right) - a^\varepsilon\left(\delta u^\varepsilon, u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right).$$

Thanks to Proposition 3, we have

$$\|a^\varepsilon - a^0\|_{\mathcal{L}^2(\mathcal{V}^\varepsilon)} = \mathcal{O}(f(\varepsilon)),$$

so

$$|a^0\left(\delta u^\varepsilon, u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right) - a^\varepsilon\left(\delta u^\varepsilon, u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right)| \leq Cf(\varepsilon)\|\delta u^\varepsilon\|_{\mathcal{V}^0}\left\|u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right\|_{\mathcal{V}^\varepsilon}.$$

Thanks to Proposition 4, we have

$$\|\delta u^\varepsilon\|_{\mathcal{V}^0} = \mathcal{O}(f(\varepsilon)).$$

This implies that

$$\left\|u^\varepsilon - \tilde{\tilde{u}}^\varepsilon\right\|_{\mathcal{V}^\varepsilon} = \mathcal{O}((f(\varepsilon))^2).$$

□

Algorithm 4 Algorithm to approximate the perturbed solution

1: Solve the global problem using a^0 to obtain u^0 (2.3).
2: Solve the local problem using a_D^ε to obtain u_D^ε (2.5).
3: Solve the global problem using a^0 to obtain δu^ε (2.17).
4: The solution is $\tilde{\tilde{u}}^\varepsilon = \tilde{u}^\varepsilon + \delta u^\varepsilon$.

We solved a local problem and two global problems at $\varepsilon = 0$. The proposed method is interesting when the problem is easy to solve at $\varepsilon = 0$. This the case when the initial problem can be solved using spectral methods with constant coefficient.

It is also interesting when the matrix arising from a^0 is factorized. In the case of iterative method, when the same matrix is used to solve many problems, it is possible to build efficient preconditioners.

This method can be considered by researchers who do not use the adjoint method [10, 36, 37, 11, 9, 8]. The low frequency effect can be taken into account by solving several time the same initial systems.

2.5. Application to topological and continuous variations

2.5.1. Problem statement

The aim of this section is to approximate the variation of a cost function in the following cases: topological pertubation, mesh perturbation, continuous variation of stiffness perturbation.

We model an experiment as follows: the domain Ω is a rectangle filled with an elastic material, the left vertical side is fixed and a vertical force of intensity μ in the downward direction is applied on the middle of the right vertical side, see Figure 2.1. The other sides are free.

Let Ω be a rectangular bounded domain of $\mathbb{R}^2$ and Γ be its boundary, composed of two parts Γ_1 and Γ_2. The considered problem has Dirichlet boundary conditions on Γ_1 and Neumann boundary conditions on Γ_2. The rectangle is subject to a displacement u solution of the following equation:

$$\begin{cases} -\nabla.\sigma(u) = 0 & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_1, \\ \sigma(u)n = g & \text{on } \Gamma_2. \end{cases} \tag{2.19}$$

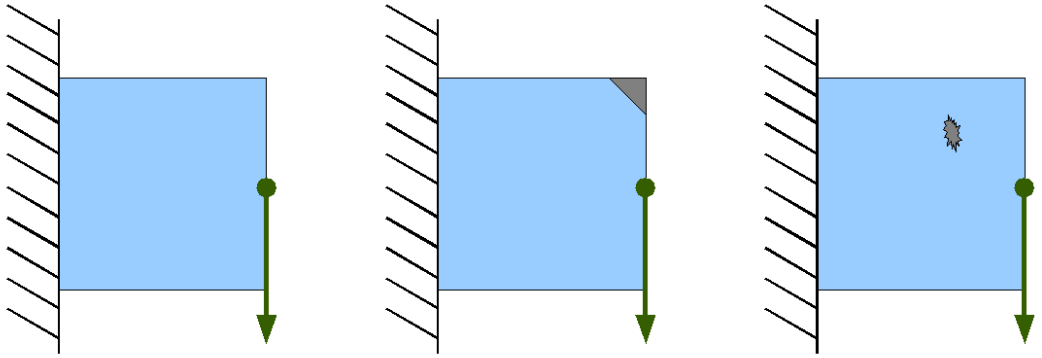


Figure 2.1.: From left to right, a model without perturbation, two models with perturbations.

with

$$\phi(u) = \frac{1}{2}(Du + Du^T),$$
$$\sigma(u) = hH_0\phi(u),$$

where H_0 is the Hooke tensor and $h(x) = 1$ and ρ represents the material presence or absence.

The optimization problem consists in minimizing the following cost function

$$j(h) = \int_{\Gamma_2} g.u\, dx, \tag{2.20}$$

with the following constraint

$$\int_\Omega h(x)dx = C,$$

where C is the material quantity to retain.

For each element, the function h is modified, then the exact variation is computed using the cost function (2.20) with u^0 the solution of Equation (2.19) and u^ε the solution of

$$\begin{cases} -\nabla.\sigma^\varepsilon(u^\varepsilon) = 0 & \text{in } \Omega^\varepsilon, \\ u^\varepsilon = 0 & \text{on } \Gamma_1, \\ \sigma^\varepsilon(u^\varepsilon)n = g & \text{on } \Gamma_2. \end{cases} \tag{2.21}$$

For each element, the numerical vault variation is computed using Theorem 2.3.2 which is equivalent to

$$j(h^\varepsilon) - j(h) = \int_{\Gamma_2} (g.\tilde{u}^\varepsilon - g.u)\ dx - \int_{\Omega^\varepsilon} \sigma(\tilde{u}^\varepsilon).\nabla(\tilde{u}^\varepsilon)\, dx + \int_\Omega \sigma(u).\nabla(u)\, dx, \tag{2.22}$$

with

$$\tilde{u}^\varepsilon = \begin{cases} u_D^\varepsilon & \text{in } D, \\ u^0 & \text{in } \Omega_0, \end{cases} \tag{2.23}$$

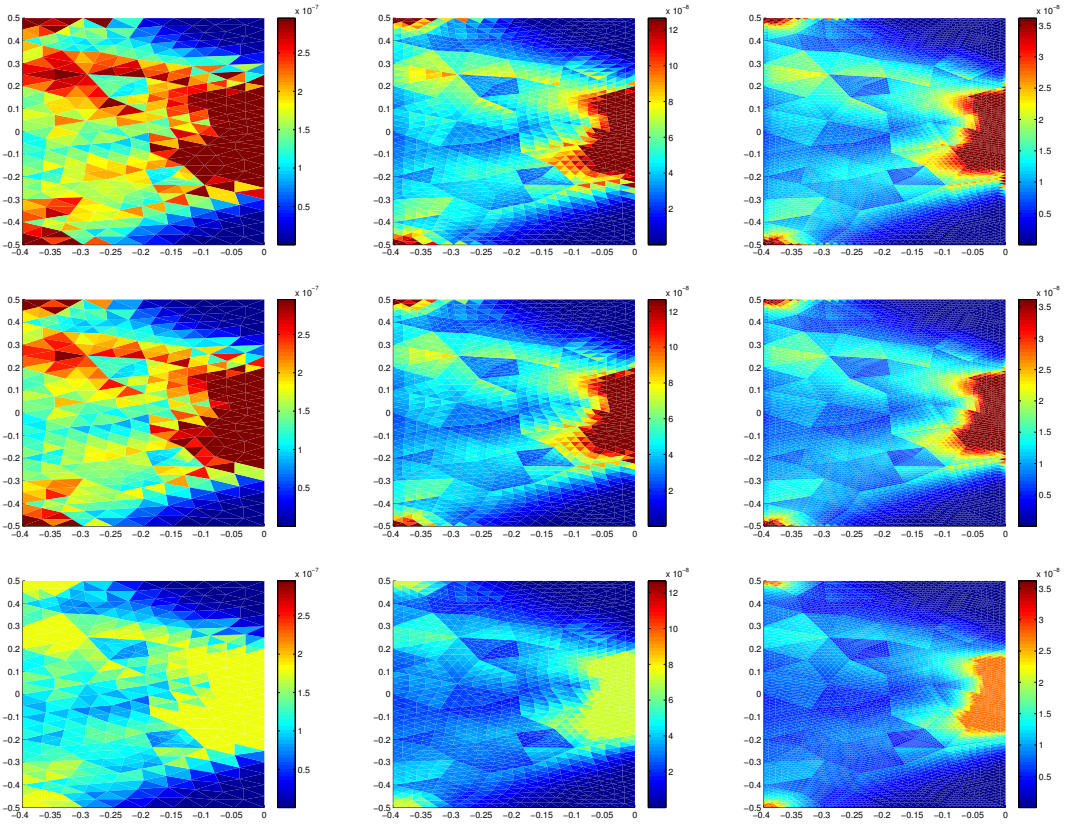


Figure 2.2.: From left to right, mesh with 516, 2064 and 8256 elements. On the top row, actual variation, on the middle row, the estimation obtained with the numerical vault and on the bottom row, the estimation obtained with the adjoint method.

where u_D^ε is the solution of the following equation

$$\begin{cases} -\nabla.\sigma^\varepsilon(u_D^\varepsilon) = 0 & \text{in } D, \\ u_D^\varepsilon = 0 & \text{on } \Gamma_1, \\ \sigma^\varepsilon(u_D^\varepsilon)n = g & \text{on } \Gamma_2. \end{cases} \tag{2.24}$$

Remark 4. In this particular problem, the adjoint solution p is equals to $-u$, so u and p have opposite updates.

2.5.2. Numerical results

In this section, the domain D is made of four layers of elements around the perturbed element. The first experiment is relative to topological perturbation, the aim being to approximate the variation of the cost function when an element is removed. The actual variation of the cost function, the estimations obtained with the adjoint method and with the numerical vault are computed on three different meshes made of 516, 2064 and 8256 elements. The deletion of each mesh element is considered. Figure 2.2 shows the

		L^1 norm		L^2 norm		L^∞ norm	
	method	difference	relative err.	difference	relative err.	difference	relative err.
mesh	adjoint	5.64×10^{-5}	0.4191	9.53×10^{-6}	0.6729	7.10×10^{-6}	0.8012
516	vault	5.64×10^{-6}	0.0419	1.02×10^{-6}	0.0718	8.41×10^{-7}	0.0948
mesh	adjoint	4.99×10^{-5}	0.3843	4.59×10^{-6}	0.6512	3.07×10^{-6}	0.8084
2064	vault	3.21×10^{-6}	0.0247	5.02×10^{-7}	0.0711	3.88×10^{-7}	0.1023
mesh	adjoint	4.50×10^{-5}	0.3587	2.01×10^{-6}	0.6141	1.18×10^{-6}	0.8147
8256	vault	2.34×10^{-6}	0.0186	2.15×10^{-7}	0.0660	1.50×10^{-7}	0.1035

Table 2.1.: Topological perturbation: comparative results of the variations obtained with the adjoint method and with the vault method.

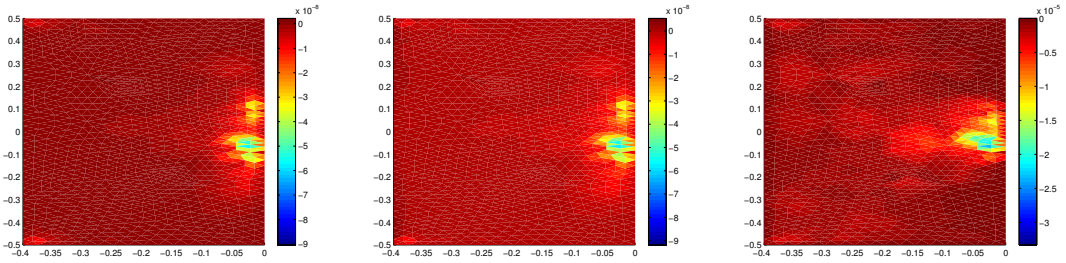


Figure 2.3.: From left to right, actual variation, the estimations obtained with the numerical vault and with the adjoint method.

	L^1 norm		L^2 norm		L^∞ norm	
method	difference	relative err.	difference	relative err.	difference	relative err.
adjoint	1.75×10^{-3}	953.17	9.7249×10^{-5}	463.06	3.32×10^{-5}	367.67
vault	2.00×10^{-7}	0.1091	2.0570×10^{-8}	0.0979	5.41×10^{-9}	0.0599

Table 2.2.: Mesh perturbation: comparative results of the variations obtained with the adjoint method and with the vault method.

variation of the cost function for each element when this element is deleted. In Table 2.1, the differences in the L^1, L^2 and L^∞ norms and the relative errors in the L^1, L^2 and L^∞ norms are given for all the results obtained with the numerical vault and with the adjoint method. One can note that the results obtained with the numerical vault are nearer to the actual variation than the estimations obtained with the adjoint method.

The second experiment is relative to mesh perturbation, the aim being to approximate the variation of the cost function when the mesh is modified. To perform the mesh modifications, at each node, one node is moved towards the node on the left in the first layer of elements. The nodes on the boundary of the domain are not modified. The actual variation of the cost function, estimations obtained with the adjoint method and

with the numerical vault are computed on a mesh made of 2064 elements. Figure 2.3 shows the variation of the cost function for each node when this node is moved, the most interesting points are the legends of the plots. In Table 2.2, the differences in the L^1, L^2 and L^∞ norms and the relative errors in the L^1, L^2 and L^∞ norms are given for all the results obtained with the numerical vault and with the adjoint method. One can note that the numerical vault outperforms the adjoint method.

The last experiments concern continuous variations of stiffness perturbations. Figures 2.4 and 2.5 present different experiments which aim at comparing the estimations obtained by the vault method and by the adjoint method with the actual variation. The mesh is made of 8256 elements. For each experiment, the locations of the perturbations are given on the left image and the associated graphs are on the right. In the curves, the abscissa is the inhomogeneity stiffness varying from 0 to 2.5 with the material stiffness equal to 1 and the ordinate is the variation of the cost function. Different cases are considered by varying the location, the number and the size of the perturbations. In Figure 2.4, the beam has an homogeneous material property, a rectangular shape and some perturbations introduced. The perturbations are inside the beam or at the extremity or at random locations with two different sizes. In Figure 2.5, the beam has a predefined shape. In each graph, the abscissa is the material property of the perturbation and the ordinate is the computed result of the cost function with the different methods. The legend is the following: blue-circle for the exact variation, red-asterisk for the estimation obtained by numerical vault and green-square for the estimation obtained by the adjoint method. One can note that the estimation by the adjoint method is an approximation of first order and the vault method is more precise.

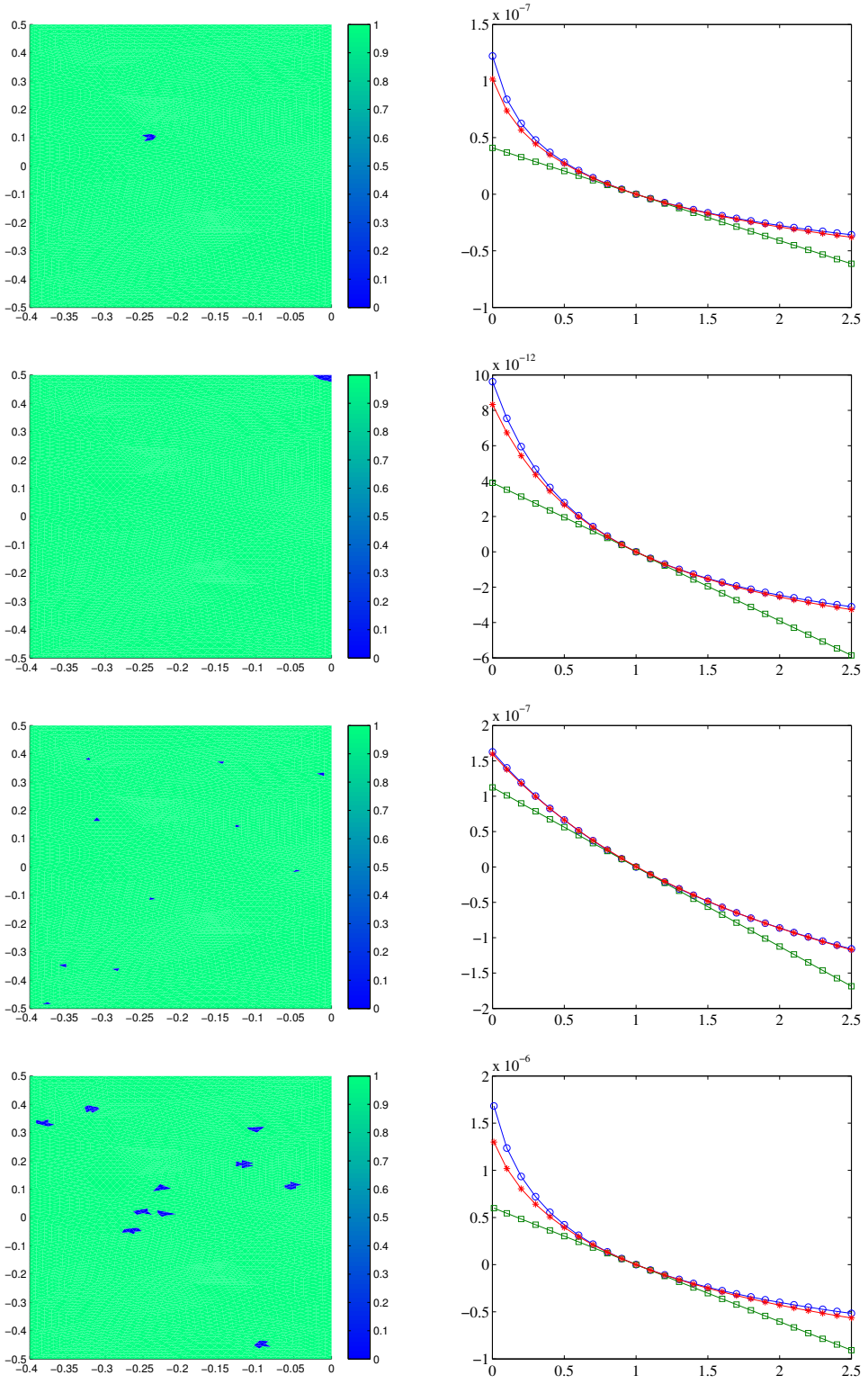


Figure 2.4.: From left to right, the perturbations and the associated curves with blue-circle for the exact variation, red-asterisk for the estimation obtained by the numerical vault and green-square for the estimation obtained by the adjoint method. The abscissa is the inhomogeneity stiffness varying from 0 to 2.5 with the material stiffness equal to 1 and the ordinate is the variation of the cost function.

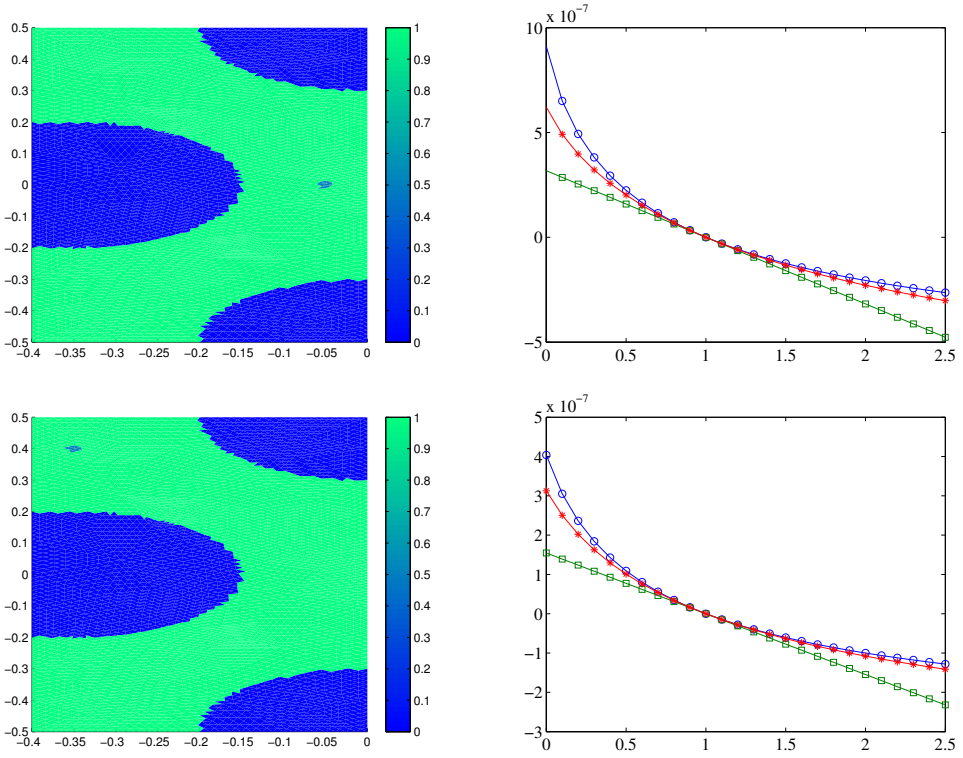


Figure 2.5.: From left to right, the perturbations and the associated curves with blue-circle for the exact variation, red-asterisk for the estimation obtained by the numerical vault and green-square for the estimation obtained by the adjoint method. The abscissa is the inhomogeneity stiffness varying from 0 to 2.5 with the material stiffness equal to 1 and the ordinate is the variation of the cost function.

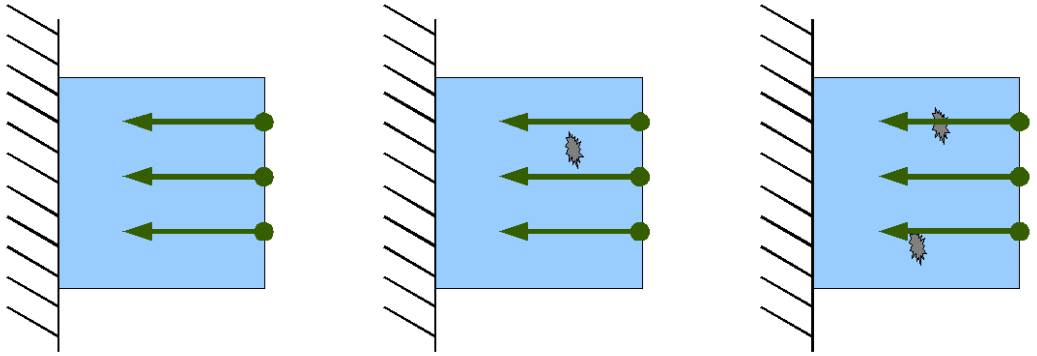


Figure 2.6.: From left to right, the experiment without inclusion, two experiments with inclusions.

2.6. Elastography

2.6.1. Problem statement

The aim of this section is to detect small inclusions in elastography [87, 86, 46, 44] which is a technique giving the directional displacement of a material under motion. This medical imaging technique relies on the fact that elastic properties of tissues provide medical information that allows to detect or classify tumors which tend to be stiffer than the surrounding soft tissue.

This imaging experiment is only an example, other applications such as localization of conductivity inhomogeneities can also be considered [12, 59].

We model an experiment as follows: the domain Ω is a rectangle filled with an elastic material, the left vertical side is fixed and the right vertical side is moved by a small amount μ in the left direction, see Figure 2.6. The other sides are free. In order to ensure uniqueness of the resulting displacement, we also prevent vertical translations.

Let Ω be a rectangular bounded domain of $\mathbb{R}^2$ and Γ be its boundary, composed of two parts Γ_1 and Γ_2. The points of the rectangle are submitted to a vertical displacement u solution of the following equation:

$$\begin{cases} -\nabla.\sigma(u) = 0 & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_1, \\ \sigma(u)n = \mu & \text{on } \Gamma_2, \end{cases} \tag{2.25}$$

with

$$\phi(u) = \frac{1}{2}(Du + Du^T),$$
$$\sigma(u) = hH_0\phi(u),$$

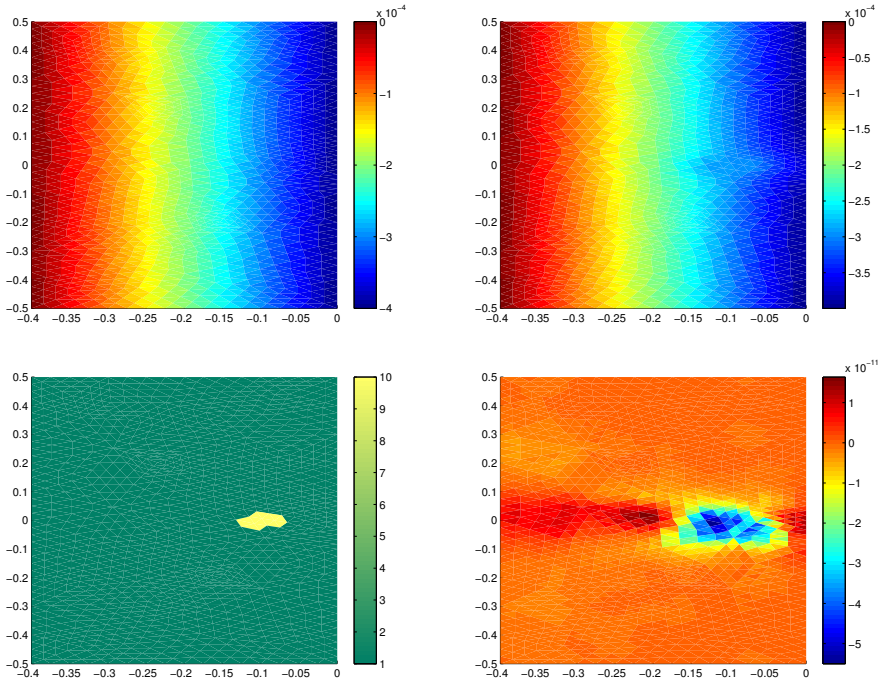


Figure 2.7.: From left to right, in the first row, the horizontal displacements for the uniform material and the perturbed material. In the second row, the inclusion and the detection obtained with the numerical vault.

where $\sigma(u)$ is a stress distribution, H_0 is the Hooke tensor and $h(x) = 1$ and σ represents the material stiffness. Unlike the previous illustration with the topology optimization, this time ε represents a stiff material property compared to the main part of the model, see Figure 2.6. Stiff material inhomogeneities are more difficult to detect than soft ones.

The optimization problem is to minimize the following cost function

$$J(h) = \int_{\Omega} |u_h - u_{obs}|^2 \, dx. \tag{2.26}$$

2.6.2. Numerical results

Two experiments are presented in this section to establish that an algorithm using the numerical vault to compute an approximation of the variations of the cost function is able to recover the localization of some material inclusions. The stiffness of the inclusion is supposed to be known.

In the first experiment, there is only one inclusion. The algorithm computes the estimation of the cost function obtained by the numerical vault for each element of the mesh. The considered element is supposed to be the inhomogeneity, the computations are only local. The domain D is made of four layers of elements around the considered element.

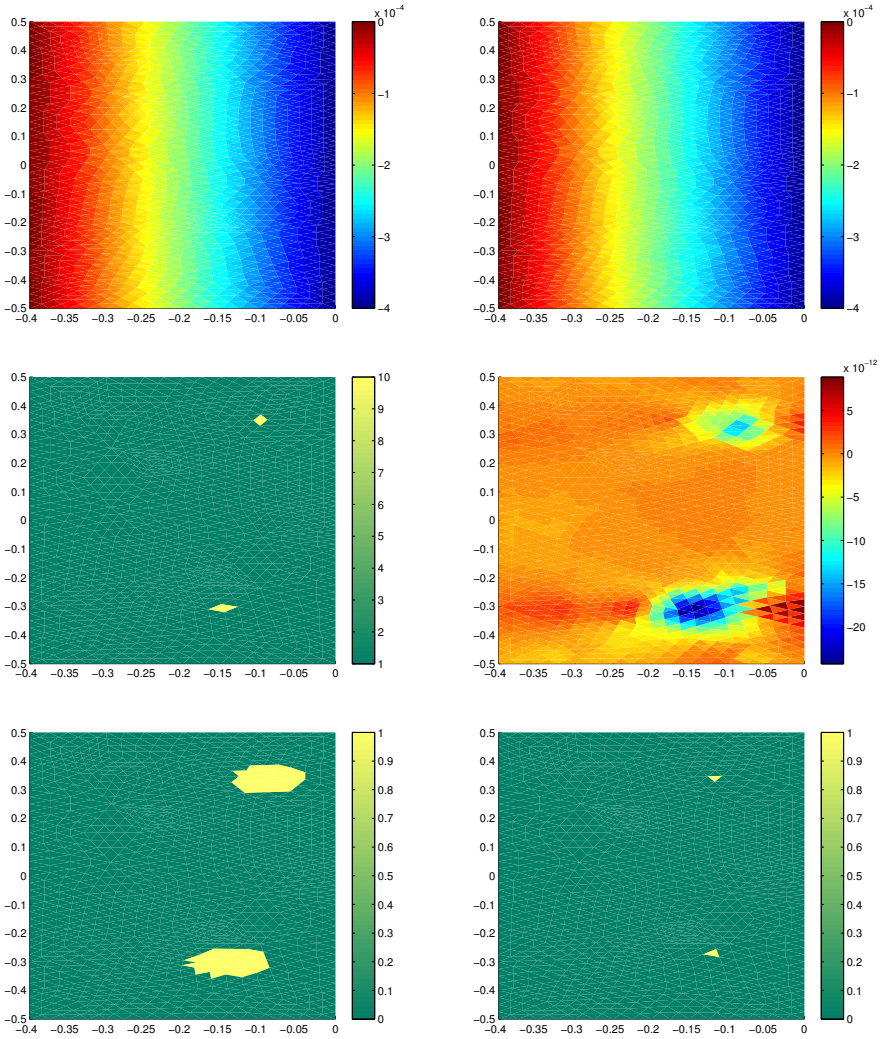


Figure 2.8.: From left to right, in the first row, the horizontal displacements for the uniform material and the perturbed material. In the second row, the inclusions and the detection obtained with numerical vault by looking for one inclusion. In the third row, two initial detection zones and the two localization results with the numerical vault by looking for two inclusions.

Figure 2.7 presents the horizontal displacements without and with the inclusion, the inclusion and the location result. The detection obtained with the numerical vault allows to detect the correct location of the inclusion and gives already an idea of its shape.

In the second experiment, there are two inclusions. The algorithm is the same as previously so it is only looking for one inclusion. Figure 2.8 presents the horizontal displacements without and with the inclusions, the inclusions considered and the location result. The detection obtained with the numerical vault by looking for one inclusion is not fully accurate but allows us to localize two inclusions. Another set of computations is performed by creating a perturbation in each localization zone. The small size of localization zones reduces the complexity of our computations. The result is neatly improved and gives better localizations.

In both cases, the detection can be improved by using the numerical vault to test the elements surrounding the localization. It may permit to obtain a better localization and gives an idea of the inclusion shape. A mesh refinement or mesh displacement, followed by some computations with the numerical vault will also generate a better detection.

2.7. Image restoration

2.7.1. Problem statement

The aim of this section is to illustrate Section 2.4 by reconstructing a perturbed solution using the numerical vault. A classical way to restore an image u from its noisy version v defined in a domain $\Omega \subset \mathbb{R}^2$ is to solve the following PDE problem

$$\begin{cases} -\nabla.(c\nabla u) + u = v \text{ in } \Omega, \\ \partial_n u = 0 \text{ on } \partial\Omega, \end{cases} \tag{2.27}$$

where c is a positive constant or a tensor [16] and ∂_n denotes the normal derivative to $\partial\Omega$.

Topological gradient has been applied to the restoration problem in order to determine a diffusion coefficient c preserving the edges [27, 21]. In this section, the estimation obtained by the numerical vault is employed.

At a given point $x_0 \in \Omega$, we insert a small insulating crack $\sigma_\varepsilon = x_0 + \varepsilon\sigma(\mathbf{n})$ where $\sigma(\mathbf{n})$ is a straight crack and $\mathbf{n}$ is a unit vector normal to the crack. We now consider $\Omega_\sigma = \Omega \backslash \sigma_\varepsilon$ the perturbed domain created by inserting this crack. The perturbed solution $u^\varepsilon \in \mathrm{H}^1(\Omega_\sigma)$ satisfies

$$\begin{cases} -\nabla.(c\nabla u^\varepsilon) + u^\varepsilon = v & \text{in } \Omega, \\ \partial_n u^\varepsilon = 0 & \text{on } \partial\Omega, \\ \partial_n u^\varepsilon = 0 & \text{on } \sigma_\varepsilon. \end{cases} \tag{2.28}$$

The corresponding variational formulation is given by

$$\begin{cases} \text{find } u^\varepsilon \in \mathrm{H}^1(\Omega_\sigma) \text{ such that} \\ a^\varepsilon(u^\varepsilon, w) = \ell^\varepsilon(w) \quad \forall w \in \mathrm{H}^1(\Omega_\sigma), \end{cases} \tag{2.29}$$

where a^ε is the bilinear form, defined on $\mathrm{H}^1(\Omega_\sigma) \times \mathrm{H}^1(\Omega_\sigma)$ by

$$a^\varepsilon(u, w) = \int_{\Omega_\sigma} c\nabla u \nabla w + u\, w\, dx,$$

and ℓ^ε is the linear form, defined on $\mathrm{L}^2(\Omega_\sigma)$ by

$$\ell^\varepsilon(w) = \int_{\Omega_\sigma} v\ w\ dx.$$

The domain Ω is decomposed in two parts. A part D containing the perturbation and its complementary $\Omega_0 = \Omega \backslash \overline{D}$. So the forms a^ε and l^ε are decomposed in the following way:

$$a^\varepsilon(u,w) = a_{\Omega_0}(u,w) + a_D^\varepsilon(u,w) = \int_{\Omega_0} c\nabla u \nabla w + u\ w\ dx + \int_D c\nabla u \nabla w + u\ w\ dx,$$

$$\ell^\varepsilon(w) = \ell_{\Omega_0}(w) + \ell_D^\varepsilon(w) = \int_{\Omega_0} v\ w\ dx + \int_D v\ w\ dx,$$

where a_{Ω_0} and l_{Ω_0} are independent of ε.

The local perturbed solution $u_{u^0}^\varepsilon \in \mathrm{H}^1(D)$ satisfies

$$\begin{cases} -\nabla.(c\nabla u_{u^0}^\varepsilon) + u_{u^0}^\varepsilon = v & \text{in } D, \\ \partial_n u_{u^0}^\varepsilon = 0 & \text{on } \sigma_\varepsilon, \\ u_{u^0}^\varepsilon = u^0 & \text{on } \partial D. \end{cases} \tag{2.30}$$

Let $\tilde{u}^\varepsilon$ be the update of u^0

$$\tilde{u}^\varepsilon = \begin{cases} u_{u^0}^\varepsilon & \text{in } D, \\ u^0 & \text{in } \Omega_0. \end{cases} \tag{2.31}$$

The function v^ε satisfies

$$\begin{cases} -\nabla.(c\nabla \tilde{u}^\varepsilon) + \tilde{u}^\varepsilon = v^\varepsilon & \text{in } \Omega_\varepsilon, \\ \partial_n \tilde{u}^\varepsilon = 0 & \text{on } \sigma_\varepsilon, \\ \partial_n \tilde{u}^\varepsilon = 0 & \text{on } \partial\Omega_\varepsilon. \end{cases} \tag{2.32}$$

The function δu^ε satisfies

$$\begin{cases} -\nabla.(c\nabla \delta u^\varepsilon) + \delta u^\varepsilon = v - v^\varepsilon & \text{in } \Omega, \\ \partial_n \delta u^\varepsilon = 0 & \text{on } \partial\Omega. \end{cases} \tag{2.33}$$

The approximation of u^ε is $\tilde{\tilde{u}}^\varepsilon = \tilde{u}^\varepsilon + \delta u^\varepsilon$.

2.7.2. Topological gradient, solving the perturbed problem

The function $u^0 \in \mathrm{H}^1(\Omega)$ is the solution of the following direct problem:

$$\begin{cases} -\nabla \cdot (c\nabla u^0) + u^0 = v & \text{in } \Omega, \\ \partial_n u^0 = 0 & \text{on } \partial\Omega, \end{cases} \tag{2.34}$$

where ∂_n denotes the normal derivative to $\partial\Omega$.

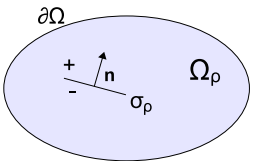


Figure 2.9.: Cracked domain.

The edge detection method consists in looking for a crack σ such that the energy $j(\varepsilon) = J_\varepsilon(u^\varepsilon) = \frac{1}{2}\int_{\Omega_\varepsilon} |\nabla u^\varepsilon|^2$ is as small as possible [27]. The function u^ε is the solution of Equation 2.28. This amounts to state that the energy outside the edges is as small as possible.

The cost function j has the following asymptotic expansion

$$j(\varepsilon) - j(0) = \varepsilon^2 g(\mathbf{x}_0, \mathbf{n}) + o(\varepsilon^2), \tag{2.35}$$

where the topological gradient g is given by

$$g(\mathbf{x}_0, \mathbf{n}) = -\pi(\nabla u^0(\mathbf{x}_0) \cdot \mathbf{n})(\nabla p^0(\mathbf{x}_0) \cdot \mathbf{n}) - \pi|\nabla u^0(\mathbf{x}_0) \cdot \mathbf{n}|^2.$$

The function p^0 is the solution to the adjoint problem:

$$\begin{cases} -\nabla \cdot (c\nabla p^0) + p^0 = -\partial_u J_0(u^0) & \text{in } \Omega, \\ \partial_n p^0 = 0 & \text{on } \partial\Omega. \end{cases} \tag{2.36}$$

The functions u^0 and p^0 are calculated in the initial domain without cracks. The topological gradient can be written $g(\mathbf{x}, \mathbf{n}) = \mathbf{n}^T M(\mathbf{x})\mathbf{n}$, where $M(\mathbf{x})$ is the 2×2 symmetric matrix defined by

$$M(\mathbf{x}) = -\pi c \frac{\nabla u^0(\mathbf{x})\nabla p^0(\mathbf{x})^T + \nabla p^0(\mathbf{x})\nabla u^0(\mathbf{x})^T}{2} - \pi \nabla u^0(\mathbf{x})\nabla u^0(\mathbf{x})^T. \tag{2.37}$$

For a given $\mathbf{x}$, $g(\mathbf{x}, \mathbf{n})$ takes its minimal value when $\mathbf{n}$ is the eigenvector associated to the lowest eigenvalue $\lambda_{min}(\mathbf{x})$ of $M(\mathbf{x})$. This value is the topological gradient associated to the optimal orientation of the crack σ_ε at the location $\mathbf{x}$.

The edges are located at points $\mathbf{x}$ where $\lambda_{min}(\mathbf{x})$ is the most negative and their orientation is given by the corresponding eigenvector.

Algorithm 5 Algorithm for image restoration with numerical vault

Input: noisy image v, diffusion coefficient c_0, $\varepsilon > 0$, threshold $\delta < 0$.
Output: restored image u.

1: Initialization: $c = c_0$.
2: Computation of u^0 and p^0, solutions of direct and adjoint problems, see equations (2.34) and (2.36).
3: Computation of the tensor $M(\mathbf{x})$ using equation (2.37).
4: Computation of the smallest eigenvalue $\lambda_{min}(\mathbf{x})$ of $M(\mathbf{x})$ at each point of the domain.
5: Set $c(\mathbf{x}) = \begin{cases} \varepsilon & \text{if } \lambda_{min}(\mathbf{x}) < \delta, \\ c_0 & \text{otherwise.} \end{cases}$
6: Computation of $u^{\varepsilon}_{u^0}$ solutions of direct problems in D around $\{\mathbf{x} \in \Omega, \lambda_{min}(\mathbf{x}) < \delta\}$.
7: Set $\tilde{u}^{\varepsilon} = \begin{cases} u^{\varepsilon}_{u^0} & \text{in } D, \\ u^0 & \text{in } \Omega_0. \end{cases}$
8: Computation of δu^{ε}, solution of direct problem, see Equation (2.33).
9: Set $\tilde{\tilde{u}}^{\varepsilon} = \tilde{u}^{\varepsilon} + \delta u^{\varepsilon}$.

2.7.3. Numerical results in image restoration

The Peak-Signal-to-Noise Ratio (PSNR) and the Structural SIMilarity (SSIM) [112] are used to quantify the restorations. The PSNR formula is the following

$$PSNR = 10 \log_{10} \left(\frac{255^2 mn}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2} \right),$$

where I and K are the images of dimension $m \times n$ which are compared. The SSIM follows this equation:

$$SSIM = \frac{(2\mu_I \mu_K + c_1)(2\sigma_{I,K} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)},$$

where μ_I is the average of I, μ_K is the average of K, σ_I^2 is the variance of I, σ_K^2 is the variance of K, $\sigma_{I,K}^2$ is the covariance of I and K, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ two variables to stabilize the division by a weak denominator (where L is the dynamic range, usually $2^{\#\text{bits per pixel}} - 1$, and $k_1 = 0.01$, $k_2 = 0.03$ by default)

Figure 2.10 presents the set of gray-level images: Barbara, Boat, Hill, Man and Lena (512×512); House and Peppers (256×256); Girl (462×357) used for the tests. The domain D is made of one layer of elements around the edge.

Table 2.3 presents the Barbara restoration results from images perturbed by additive Gaussian noises of standard deviation $\sigma = 5$, 15 and 25. The first approximation (2.31) (FA) is not sufficient. The second approximation (SA) has similar results. Figure 2.11 shows the noisy image and illustrates the similarity between the two restorations.

Figure 2.10.: Set of gray-level images used for tests.

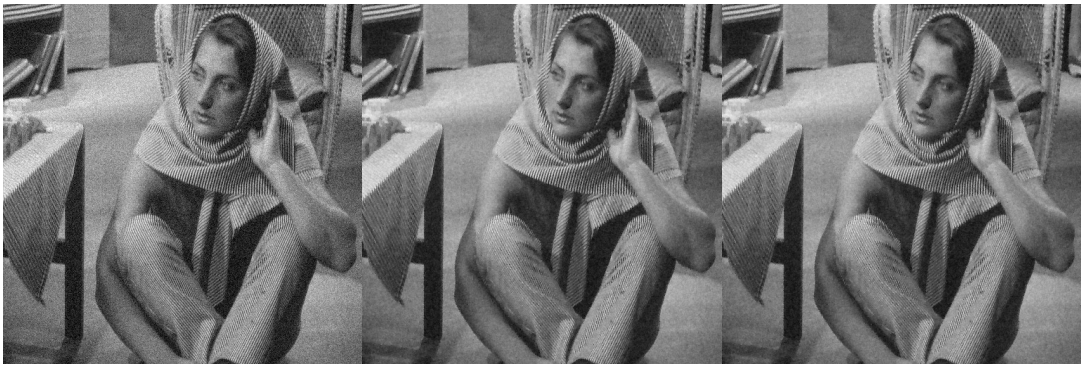


Figure 2.11.: From left to right, the Barbara noisy image with $\sigma = 15$, the reconstructions with respectively the isotropic diffusion and the numerical vault.

Barbara	$\sigma = 5$		$\sigma = 15$		$\sigma = 25$	
512×512	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Noisy	34.13	0.881	24.62	0.560	20.25	0.386
ID	35.78	0.942	27.92	0.760	25.32	0.638
FA	34.87	0.933	27.20	0.747	24.93	0.630
SA	35.74	0.938	27.87	0.758	25.28	0.638

Table 2.3.: Comparative results of the restoration using the isotropic diffusion (ID), the first approximation (2.31) (FA) and the second approximation (SA) obtained with Algorithm 2.7.2.

Table 2.4 presents the other image restoration results from images perturbed by additive Gaussian noises of standard deviation $\sigma = 5$, 15 and 25. The restorations giving by the direct equation and by the numerical vault are nearly equivalent. Figure 2.12 gives another visual results, the noisy image and illustrates the similarity between the two restorations for the Girl image.

		$\sigma = 5$		$\sigma = 15$		$\sigma = 25$	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Boat	Noisy	34.18	0.885	24.63	0.537	20.29	0.348
	ID	36.14	0.926	29.97	0.797	27.56	0.709
	SA	36.14	0.926	29.80	0.795	27.40	0.705
Hill	Noisy	34.13	0.888	24.63	0.536	20.27	0.328
	ID	36.17	0.930	30.32	0.794	28.25	0.711
	SA	36.17	0.930	30.22	0.792	28.16	0.709
Lena	Noisy	34.13	0.844	24.62	0.451	20.21	0.272
	ID	37.24	0.931	31.42	0.819	29.14	0.754
	SA	37.21	0.931	31.32	0.818	29.03	0.752
Man	Noisy	34.13	0.878	24.60	0.522	20.25	0.332
	ID	36.44	0.937	30.27	0.807	28.07	0.729
	SA	36.43	0.937	30.15	0.805	27.91	0.726
House	Noisy	34.20	0.839	24.64	0.447	20.19	0.278
	ID	37.34	0.926	31.59	0.823	28.95	0.753
	SA	37.24	0.925	31.24	0.820	28.52	0.747
Peppers	Noisy	34.14	0.870	24.60	0.523	20.35	0.352
	ID	37.48	0.954	30.93	0.859	28.10	0.786
	SA	37.38	0.954	30.69	0.857	27.90	0.782
Girl	Noisy	34.17	0.815	25.12	0.437	20.81	0.271
	ID	37.72	0.944	31.04	0.786	28.22	0.698
	SA	37.59	0.943	30.91	0.785	28.07	0.695

Table 2.4.: Comparative results of the restoration using the isotropic diffusion (ID) and the second approximation (SA) obtained with Algorithm 2.7.2.

PCG iter.	$\sigma = 5$		$\sigma = 15$		$\sigma = 25$	
Image	ID	SA	ID	SA	ID	SA
Barbara	10	8	10	7	9	5
Man	9	8	9	5	10	4

Table 2.5.: Comparative results of number of iteration steps of PCG using the isotropic diffusion (ID) and the second approximation (SA) obtained with Algorithm 2.7.2.

The numerical vault approach is faster than the classical method. For the reconstruction with the isotropic diffusion (ID), Equation (2.27) needs to be solved with a diffusion coefficient c that depends on the topological gradient. This resolution is made with a preconditioned conjugate gradient (PCG) method preconditioned by a DCT solver and a constant diffusion coefficient c (see [20] for details). The method with the numerical vault uses the same tools to compute the solution of Equation (2.33) with a constant diffusion coefficient c. The computations to obtain the first update are only local, see Equation (2.30), and are negligible compared to the problem size. Table 2.5 presents the number of iterations of PCG using the isotropic diffusion (ID) and the second approximation obtained with the algorithm (SA) from images perturbed by additive Gaussian noises of standard deviation $\sigma = 5$, 15 and 25. For the same tolerance, in term of PCG iterations, the resolution with the numerical vault is faster than the classical method.

Figure 2.12.: From left to right, the Girl noisy image with $\sigma = 15$, the reconstructions obtained with respectively the isotropic diffusion and the numerical vault.

2.8. Conclusion

In this work, the adjoint method has been extended with a new method named the numerical vault. The theoretical sections justify that the numerical vault is able to capture the variation of the cost function in a better way than the previous adjoint method. They also demonstrate that the vault method can be applied to update the solution of a singularly perturbed problem.

The numerical applications on mesh perturbation and topological and continuous variations show the superiority of the results obtained using the vault method compared to those obtained using the adjoint method. The curves presented in the numerical applications have illustrated that the numerical vault captures higher order of the variation compared to the adjoint method. As a result, inclusions in elastography could be successfully localized. To solve these problems, only a very simple direct solution and some local computations have been performed. Major advantages of the method are

that the numerical vault is non invasive and can be used with a parallel computing implementation.

The second interest is to offer a new method to resolve direct equations, avoiding the resolution of the full direct perturbed equation, which can be costly in computing time, when only local zones are affected by modifications of the equation operator. In the image applications, the purpose was not to obtain good restoration results - they can be improved by using an anisotropic diffusion scheme [71] - but to demonstrate that the approximation created with the numerical vault is equivalent to the direct solution.

The numerical vault is a universal tool that can be applied to many different linear problems as long as the hypothesis holds. Research in nonlinear applications is under consideration.

3. Dipolar source localization in electroencephalography

3.1. Introduction

The brain is made of billions of neurons that make trillions of synapses to form the human neural circuits. The neuron processes and transmits information by bioelectrical and chemical signaling. The ElectroEncephaloGraphy (EEG) and the MagnetoEncephaloGraphy (MEG) are able to detect electrical or more precisely bioelectrical changes. These changes are consequences of ion movements between the inner and the outer cell. Magnetic Resonance Imaging (MRI) is able to detect the neurotransmitters, endogenous chemicals which transmit signals from a neuron to a target cell across a synapse.

In previous research, during my master thesis, concerning the localization of dipolar sources in electroencephalography, I developed a new method to localize dipolar source in two dimensions. Theoretical and numerical results are recalled in this chapter. Within the present research, I have extended this method to the case of electroencephalography in three dimensions.

Whereas the localization methods in electroencephalography use classically iterative algorithms and some a priori assumptions, the present study attemps at developing a non iterative approach without a priori. Two cost functions have been studied: one on the boundary which is the classical misfit between the data and the estimated solution and the other on the domain which is related to the Kohn-Vogelius criterion. The second cost function has as its purpose the incorporation of more information. Our approach is exhaustive, it determines the best dipolar source, in the case of one source, or the best combination of dipolar sources for more than one.

The next Section 3.2 presents the clinical contexts: brain, the interest of electroencephalography, differences with magnetoencephalography. The physical equations of the problem are introduced in Section 3.3. The different ways to model the forward problem are presented in Section 3.4. The choice of the Finite Element Method (FEM) and the creation of our dipolar source are also explained. A comparison with Dipole Simulator, a free program for the simulation of EEG and MEG activity is also provided. In Section 3.5, the state of the art in dipolar source localization is briefly given together with our mathematical study of the inverse problem in two and three dimensions. The

numerical results in two and three dimensions are presented in Sections 3.6 and 3.7 respectively.

3.2. Clinical contexts

3.2.1. Neurons and brain

Even though neurons have different shapes and sizes, whatever their localization in the nervous system, they share the same structure, schematized in Figure 3.1. The cell body called the soma contains the cell nucleus. The dendrites are the receptors of a neuron. The axon, the second type of protoplasmic protrusion that extrudes from the cell body, is longer than the dendrites and conducts the electrical impulses away from the neuron's cell body or soma. An axon terminal is separated from a neighboring neuron by a small gap called a synapse. The synapses at the end of the axon transmit the information with neurotransmitters in the case of a chemical synapse or an electrical current in the case of an electrical synapse.

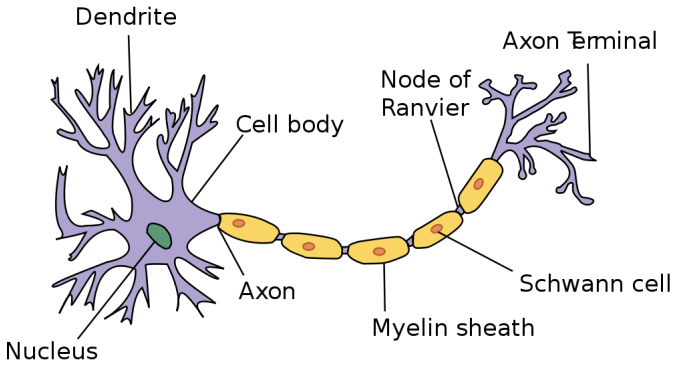


Figure 3.1.: Neuron structure.

The system can be schematized with a telegraphist model:

- ➢ receiving antennas: dendrites,
- ➢ a source of energy: soma with the cell nucleus,
- ➢ a communication line: axon,
- ➢ transmitting antennas: axon terminals.

The study of cerebral activity reveals the functional activity of the brain and the first maps of the cerebral cortex date from the beginning of the 20th century. The German neurologist Korbinian Brodmann based his work on the cytoarchitecture, the organization of cells in the brain. Broadmann's areas, see Figure 3.2, are still relevant today.

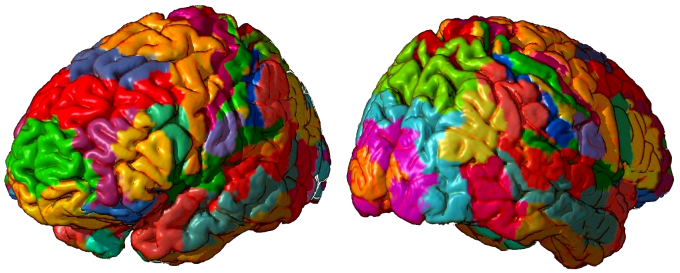


Figure 3.2.: Brodmann's areas.

The electrical potentials generated by single neurons cannot be detected with surface electrodes. EEG and MEG activities therefore reflect the summation of the synchronous activity of groups of widely inter-connected parallel neurons. The cortical macrocolumns of the cerebral cortex are considered as the producers of most EEG signal because the neurons are well-aligned and have an orientation perpendicular to the surface. A set of active pyramidal cells can be modeled electrically as a dipole layer. Due to the fact that voltage fields fall off with the square of the distance, the activity from deep sources is more difficult to detect than currents near the skull. The deep sources are often simulated by multipolar sources.

3.2.2. Measurement of the brain electrical activity

Electroencephalography (EEG)

In 1875, the English physician, Richard Caton $(1842 - 1926)$ discovered, thanks to a galvanometer, the presence of electrical currents in the brain of animals. The German physiologist and psychiatrist Hans Berger $(1873 - 1941)$ recorded the first human EEG in 1924. He also invented the electroencephalogram. This technique is useful in neurological diseases like epilepsy. Since 1975, it is also used to study the sleep stages. Despite technological innovations, the principle has remained the same until now.

The developement of MRI lead to a decrease of interest for the information given by the EEG. Nowadays, progress in imaging techniques is so great that one can ask oneself what the interest of electroencephalography is as it does not seem to have evolved.

This noninvasive imaging technique is able to detect some brain disorders. It is less expensive than a lot of other techniques, and can easily be repeated. The temporal resolution is of the order of milliseconds. It is particularly relevant when the cerebral anomalies are not linked to morphological modifications, like disorders of attentiveness, hallucinations and epilepsy. This technique is used in cognitive science, cognitive psychology, and psychophysiological research.

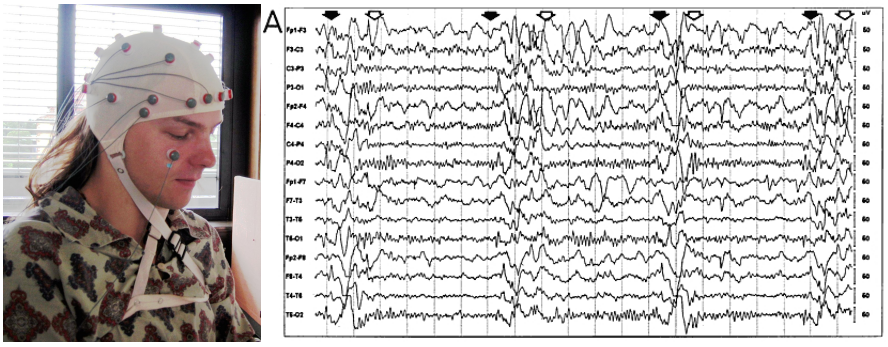


Figure 3.3.: From left to right, research participant with 32 electrode EEG cap and EEG signals.

Usually, to perform the electroencephalography, several electrodes are positioned on the patient's scalp. A conductive gel or paste is interposed between the electrode and the scalp. Different configurations are possible: when there is only a limited number of selective zones of interest, only a few electrodes are positioned to detect the activation in these zones, whereas if a larger number of electrodes on the whole surface of the head is necessary, the electrodes are integrated within an international configuration in caps or nets, see Figure 3.3. To give an example, some international configurations are made of 64, 128 or 256 electrodes. An EEG voltage signal represents a difference between the voltages of two electrodes. The signals are displayed on paper or recorded and displayed on a monitor, see Figure 3.3.

Advantages:

- ➢ Good temporal resolution.
- ➢ Hardware costs are significantly lower in comparison to other methods.

Limitations:

- ➢ Spatial resolution may be degraded due to the fact that the electrical potential can be spread out as a result of the low conductivity of the skull.
- ➢ Technological progress has a low influence on its evolution.

Magnetoencephalography (MEG)

In 1968, the physicist David Cohen registered for the first time the magnetic field produced by bioelectrical phenomena occurring naturally in the brain. The main difficulty of the MEG development comes from the fact that typical scalp magnetic fields are of the order of 10 femtotesla (fT) for cortical activity; this is of the order of a 10 billionth of the earth's magnetic field. James E. Zimmerman's invention, the Superconducting Quantum Interference Device (SQUID), overcame this difficulty.

Figure 3.4.: From left to right, research participant with MEG instrument and an entrance to a magnetically shielded room, showing the separate shielding layers.

Recent set ups are made of 50 to 200 SQUIDS captors. Based on superconducting loops, they are able to measure extremely weak magnetic fields. The sensors use low-temperature electronics cooled by liquid helium at 4 Kelvin. MEG fields are measured inside a magnetically shielded room for protection against magnetic background noise made of higher-frequency electromagnetic perturbations.

Advantages:

- ➢ Good temporal resolution, of the order of milliseconds.
- ➢ Good spatial resolution.

Limitations:

- ➢ Captures only the radial component.
- ➢ The cap cannot be adapted to the head of the patient.
- ➢ High cost of equipment and recurring cost (liquid helium).

Remark 5. Figure 3.5 illustrates the fundamental difference between the MEG and the EEG. Magnetic fields are less distorted than electric fields by the skull and scalp, which results in a better spatial resolution of the MEG. Whereas scalp EEG is sensitive to both tangential and radial components of a current source in a spherical volume conductor, MEG detects only its tangential components.

Remark 6. The white matter consists of axons, grouped in bundles. The conductivity along the nerve bundle is nine times greater than perpendicular to the nerve bundle. It is often assumed that the conductivity tangential to the skull surface is ten times larger than the radial conductivity [61]. Concerning the skull, other authors have different opinions and consider that the skull should be described by three isotropic layers [102].

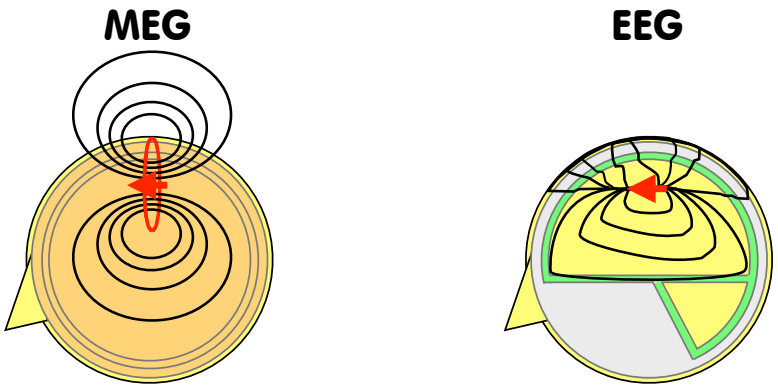


Figure 3.5.: From left to right, magnetic fields and electric fields.

3.3. Physics of the problem

3.3.1. Maxwell's equations

The Maxwell's equations are the following:

$$\nabla.E = \frac{\rho}{\epsilon_0} \qquad \text{(Gauss's law for electricity)}, \tag{3.1}$$

$$\nabla.B = 0 \qquad \text{(Gauss's law for magnetism)}, \tag{3.2}$$

$$rot(E) = -\frac{\partial B}{\partial t} \qquad \text{(Faraday's law)}, \tag{3.3}$$

$$rot(B) = \mu_0 \left(J + \epsilon_0 \frac{\partial E}{\partial t} \right) \qquad \text{(Ampère's law with Maxwell's correction)}, \tag{3.4}$$

where

- E is the electric field,
- B is the magnetic field,
- J is the total current density,
- ρ is the charge density,
- $\epsilon_0 \approx 8,85.10^{-12}\ F.m^{-1}$ is the electric permeability of free space,
- $\mu_0 = 4\pi.10^{-7}\ H.m^{-1}$ is the magnetic permeability of free space.

The permeability of the tissue in the head is the same as the one of free space. In a passive nonmagnetic medium, J is the sum of ohmic volume current and the polarization current:

$$J = \sigma E + \frac{\partial P}{\partial t}, \tag{3.5}$$

where

- σ is the conductivity,
- $P = (\epsilon - \epsilon_0)E$ is the polarization,
- ϵ_0 is the permeability of the material.

3.3.2. Quasistatic approximation

To simplify the study of these equations, we consider the quasistatic approximation which consists in negleting the time derivatives of the fields E and B. The explanation of this simplification can be found in [66].

In EEG and MEG, the frequencies of the phenomena of interest are under 100 Hz. We consider an electromagnetic phenomenon of frequency f.

$$E = E_0(r)exp(i2\pi ft). \tag{3.6}$$

By combining Equations (3.4) and (3.5), we obtain the following equation:

$$rot(B) = \mu_0 \left(\sigma E + (\epsilon - \epsilon_0)\frac{\partial E}{\partial t} + \epsilon_0 \frac{\partial E}{\partial t} \right) = \mu_0 \left(\sigma E + \epsilon \frac{\partial E}{\partial t} \right). \tag{3.7}$$

For the quasistatic approximation to be valid, it is necessary that:

$$\left| \epsilon \frac{\partial E}{\partial t} \right| << |\sigma E| .$$

This equation is equivalent to $2\pi f\epsilon/\sigma << 1$. With the parameters $\sigma = 0.3\ \Omega^{-1}m^{-1}$, $\epsilon = 10^5\epsilon_0$ and $f = 100$ Hz, this condition: $2\pi f\epsilon/\sigma = 2.10^{-3} << 1$ is satisfied.

By combining Equations (3.3) and (3.7), we obtain the following equation:

$$rot(rot(E)) = -\frac{\partial}{\partial t}(rotB) = -\mu_0 \frac{\partial}{\partial t} \left(\sigma E + \epsilon \frac{\partial E}{\partial t} \right) = -\mu_0 i2\pi f \left(\sigma + i2\pi f\epsilon \right) E.$$

The solutions of this equation have spatial changes with a characteristic length scale $|\mu_0 i2\pi f \left(\sigma + i2\pi f\epsilon \right)|^{-1/2}$. With the above parameters, we obtain 65 meters which is

much longer than the diameter of the head. This implies that the contribution of $\frac{\partial B}{\partial t}$ to E is small. The quasistatic approximation is therefore justified.

In the quasistatic approximation since the curl of E is null, the electric field can be expressed as the derivative of the electrical potential V.

$$E = -\nabla V. \tag{3.8}$$

Equation (3.4) becomes:

$$rot(B) = \mu_0 J. \tag{3.9}$$

3.3.3. Forward problem in EEG

Within the quasistatic approximation, Equation (3.5) has to be modified to take into account that the brain is not a passive medium. Let J^p be the primary currents generated by the neuronal activity. The equation is the following:

$$J(r) = \sigma(r)E(r) + J^p(r). \tag{3.10}$$

Equation (3.8) implies that:

$$J = J^p - \sigma \nabla V.$$

Equation (3.9) implies that:

$$rot(B) = \mu_0 J = \mu_0 J^p - \mu_0 \sigma \nabla V.$$

Using the divergence operation and Equation (3.2), we finally obtain the following equation

$$-\nabla.(\sigma \nabla V) = \nabla.(J^p). \tag{3.11}$$

We recognize a Poisson's equation which links the conductivity σ, the electric potential V and the primary currents J^p. The forward problem consists in finding V in Equation (3.11) with σ and J^p given.

3.4. Model of the forward problem

In this thesis, a three-shell concentric spherical model and the Finite Element Method have been chosen to model the forward problem.

3.4.1. The different approaches

To compute the electric potential at the surface of the head, a spherical model is used which contains three concentric layers; each one with different conductivity values representing the brain, the scalp and the skull. This model is the most commonly used, however more layers could be introduced. Thanks to the anatomic MRI, more realistic models can also be formulated. These models can be adapted to the specific patient's head, they are more accurate because they take into account the head geometry, and that the thickness and the curve of the skull influence the results. However, they render the computation more complex and increase the computing time. There are three different approaches to solve the forward problem:

- Boundary Element Method (BEM)
- Finite Element Method (FEM)
- Finite Difference Method (FDM)

Hybrid methods of BEM and FEM can be found in the litterature [68] and can also be applied to EEG and MEG.

All these methods face intrinsic difficulties:

- the conductivity of the different head components are subject to many studies. This problem is not solved yet due to the absence of method to measure the conductivity in vivo. The parameters are therefore approximated and variations are observed using different methods as stated in [60].
- it is impossible to identify the skull bone surface in MRI due to the lack of water in this part. In general, an interpolation is made from the cortex and scalp ones.
- creation of "good" meshes. Most of these meshing schemes start by overlaying a simple model on top of the voxelized images then it is necessary to perform refinements and improve boundaries based on the needs of the simulation. Although carrying out refinement to an existing mesh is a relatively straightforward task, it becomes much more challenging when maintaining good mesh quality, i.e. controlling the shape and size of the elements.

Table 3.1 gives a comparison of the different methods for solving Poisson's equation with a realistic head model [60]. The iFDM and aFDM are both Finite Difference Methods; they only differ by the fact that the aFDM takes into account the anisotropy.

Methods	BEM	FEM	iFDM	aFDM
Position of computational points	surface	volume	volume	volume
Free choice of computational points	yes	yes	no	no
System matrix	full	sparse	sparse	sparse
Solvers	direct	iterative	iterative	iterative
Number of compartments	small	large	large	large
Complex mesh	yes	yes	no	no
Handles anisotropy	no	yes	no	yes

Table 3.1.: Comparison of the different methods for solving direct problem.

3.4.2. Finite Element Method

The Finite Element Method is a widely used technique to approximate and discretize the partial derivative equations. During the middle of the nineties, this method has been applied to realistic head models. The equation to solve is Equation (3.11) with a boundary condition.

$$\begin{cases} -\nabla.(\sigma\nabla V) = \nabla.(J^p) \ \text{ in } \Omega, \\ \dfrac{\partial V}{\partial n} = 0 \ \text{ on } \partial\Omega. \end{cases} \tag{3.12}$$

In FEM, the mesh is not decomposed into three layers of different conductivity, instead the conductivity distribution σ is defined in Ω which represents the head. Outside the head, the conductivity follows the equation $\sigma(x) = 0$.

The problem is formulated in a variational form. The aim is to determine the energy formulation corresponding to Equation (3.12). This energy is:

$$E(V) = \frac{1}{2}\int_\Omega \sigma|\nabla V|^2 - \int_\Omega \nabla.(J^p)V. \tag{3.13}$$

This minimization problem is discretized by the Galerkin's method. The unknown is discretized in a subspace of functions of finite dimension. Let $\varphi_1, \ldots, \varphi_n$ be the basis functions, n is the dimension of the subspace. In this basis, V has the following decomposition:

$$V = \sum_{i=1}^{n} V_i\varphi_i.$$

By inserting this decomposition in Equation (3.13), an approximation of the problem is given by the following linear system:

$$K\mathbf{v} = \mathbf{b}, \tag{3.14}$$

where $K_{ij} = \int_\Omega \sigma \nabla\varphi_i \nabla\varphi_j$, $b_i = \int_\Omega \nabla.(J^p)\varphi_i$, and $v_i = V_i$.

The matrix K is called the stiffness matrix. The formula for K_{ij} allows to establish two fundamental properties of the stiffness matrices. First it is symmetric and the small support of the functions φ_i is the reason why the matrix K is sparse. That is why a mesh of the head does not increase too much the computing time required to perform the resolution of the sytem (3.14).

The advantages of the finite element method are the ability to adapt the mesh to the geometry of the head and to use a variable conductivity. It possesses a fix cost in the computing time when the stiffness matrix is assembled. The drawback of this method is linked to the difficulty of creating a "good" tetraedric mesh from anatomical MRI.

The meshes, used for the forward and inverse problems in EEG, have been generated with Gmsh, a three-dimensional finite element mesh generator developed by Christophe Geuzaine and Jean-François Remacle [50, 51]. The computations in MATLAB are performed with GetFEM++, a generic C++ finite element library developed by Yves Renard and Julien Pommier [96].

3.4.3. Numerical creation of dipolar sources

Dipole Simulator

During this thesis, EEG signals were necessary. The doctors from INSERM advice us to use Dipole Simulator a well-known and used free program for the simulation of EEG and MEG activities. This program is written by Patrick Berg and is based on spatio-temporal dipole models [30].

With this program, you can

- ➢ Simulate both EEG and MEG.
- ➢ See maps resulting from a dipole anywhere within the head.
- ➢ Add up to 20 dipoles to a model.
- ➢ Generate independent waveforms of each dipole (source waveform). You can also read in source waveforms from an ASCII file (BESA swf).
- ➢ See the surface data resulting from the model, as reference-free, average reference, Laplacian (CSD) reference, or using any electrode/sensor as reference.

Linked electrode pairs can also be defined as reference.

- ➢ Place a cursor anywhere in the time interval and view maps of the data at that time point.
- ➢ Specify the parameters of the spherical head model.
- ➢ Add coherent noise to the data defined in terms of rms level and level of alpha activity.
- ➢ Learn about filters by displaying the effects of applying various types of filter to your simulated data.
- ➢ Save and read the models you have generated. Read BESA par and bsa models.
- ➢ Save the generated data in BESA avr (ASCII) format.

The EEG data are generated using a spherical four-shell head model of radius 85 mm. Table 3.2 gives the default parameters. CSF means CerebroSpinal Fluid. In the numerical tests, Dipole Simulator was configured for a spherical three-shell head model, the cerebrospinal fluid conductivity was change into 0.33 and consider as the brain.

Layer	Thickness	Conductivity
Scalp	6 mm	0.33 S/m
Bone	7mm	0.0042 S/m
CSF	1 mm	1.0 S/m
Brain	71 mm	0.33 S/m

Table 3.2.: Defaul parameters of the spherical model in Dipole Simulator.

The noise is modelized with 200 random sources with random locations and orientations. The locations are coherent with the brain activity. The user specifies the proportion to the dipole simulated data.

Figure 3.6 presents the typical Dipole Simulator window. At the left, there is the heads frame, the six heads help editing dipole location and orientation, data maps can be displayed. In the middle, there is the dipole maps, topography of single sources are seen and the input source waveforms can be edited here. At the right, there is the output data waveform display at each sensor. The scrolbars allow to adjust the noise. At the bottom, there is a map area which displays a sequence of map.

Our method of creation

Altough Dipole Simulator provides EEG data for numerical experimentations, the simulation of a dipolar source was necessary. In [60], the creations are based on the derivative

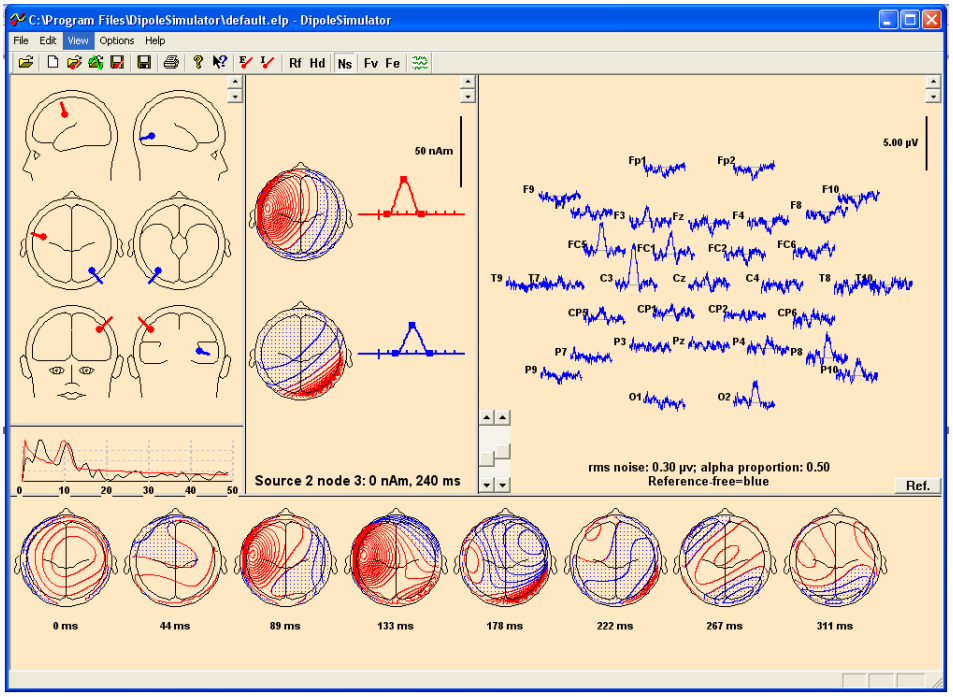


Figure 3.6.: Dipole Simulator window.

of the Dirac function.

$$\mathbf{d}.\nabla\delta_{x_0} = \frac{\delta_{x_0+\mathbf{d}/2} - \delta_{x_0-\mathbf{d}/2}}{\mathbf{d}}.$$

One possibility is to use the mesh nodes, the other possibility is to use the gravity center of the finite elements. These two approaches need a fine mesh and the dipole orientation is constrained by the mesh. To overcome this difficulty, some works use linear combinations.

The method during the present thesis is based on one finite element. There are two avantages:

- ➢ the orientation of the dipole is not constrained by the mesh,
- ➢ it assures a better approximation of the dipolar source because the function is based on only one element.

The aim is to approximate numerically the linear form l such as for all the test functions φ, $l(\varphi) = \boldsymbol{d}.\nabla\varphi(x_0)$. This form is approximated by considering the element E of the mesh containing x_0 (in the numerical experimentations x_0 is the barycenter of E). For all φ_j, we have the following equation

$$l(\varphi_j) = \frac{1}{|E|}\int_E \mathbf{d}.\nabla\varphi_j,$$

where $|E|$ is the area of E. This quantity is nonzero for the φ_j of the nodes of the element E.

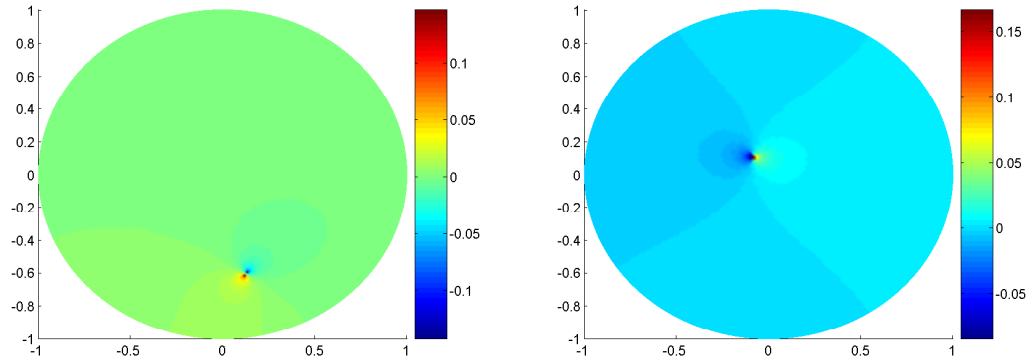


Figure 3.7.: Two creations of one dipolar source.

3.5. Inverse problem

3.5.1. State of the art

The inverse problem is ill-posed as different source configurations can generate the same distribution of potentials fields on the scalp, for a review see [47]. There are two types of parameters to determine:

- ➢ parameters with a linear dependency: amplitude of the sources,
- ➢ parameters with a non-linear dependence: locations and orientations.

The choice of the mathematical method depends on the number of dipoles assumed in the model and whether one or more of dipole position(s), magnitude(s) and orientation(s) is/are kept fixed and which, if any, of these are assumed to be known. Moreover some or all parameters can be fixed in time or time-varying [83]. For example, in [100], the following four dipole models with different dipole moment constraints are discussed:

- ➢ constant unknown dipole moment,
- ➢ fixed known dipole moment orientation and variable moment magnitude,
- ➢ fixed unknown dipole moment orientation, variable moment magnitude,
- ➢ variable dipole moment orientation and magnitude.

This section presents a simple review of the main resolution methods in EEG, for extended reviews see [1, 54]. These resolution methods comprise two categories:

- ➢ parametric methods
- ➢ non-parametric methods

The parametric methods are also referred to as equivalent current dipole methods or concentrated source or spatio-temporal dipole fit models. In this approach, a search is made for the best dipole locations, orientations and magnitudes. The models range in complexity from a single dipole to multiple dipoles and from a spherical head model to a realistic head model. The number of dipoles is inferred a priori, a choice which can be guided by the type of experiment, a visual interpretation of the EEG data, ... Time can also be taken into consideration.

There are iterative methods such as the multiple-signal classification algorithm (MUSIC) [81], the recursive extension (R-MUSIC) [79], the recursively applied and projected extension (RAP-MUSIC) [80] and beamforming approaches [70, 110]. There are direct methods such as algebraic methods [22, 23]. One can note that some analytical methods which work correctly in two dimensions are not optimal in three dimensions.

The non-parametric methods are also referred to as imaging methods, distributed source models or distributed inverse solutions. In these models several dipole sources with fixed locations and most of the time fixed orientations are distributed in the whole brain volume or on the cortical surface. The orientation is assumed a priori perpendicular to the cortex due to the brain structure. These methods estimate the amplitude of the sources.

The most known methods are low-resolution electrical tomography (LORETA) [92], the standardized extension (sLORETA) [91], local autoregressive average (LAURA) [41].

On the one hand, the complexity of parametric models depends on the a priori chosen number of dipoles. Since in this case a search is made for dipole position, orientation and magnitude which intervene non-linearly in the equations, parametric approaches present a non-linear problem.

On the other hand, in non parametric techniques, the dipole location is not estimated, and such techniques present a linear problem. The inconvenient is the need to use regularization methods to choose between all possible solutions.

Some hybrid methods, which combine the two approaches, have been proposed to merge their advantages [25]. The last generation of MEG instruments allows the simultaneous recording of brain magnetic fields and electrical potentials on the scalp. Some methods are specially developed to fusion the different information available and avoid the possible drawback of each method as for example the estimation of deeper or radial sources with MEG [26].

3.5.2. Cost function on the boundary

The cost function

The cost function to minimize is the following:

$$j(f) = J(u_f) = \frac{1}{2}\int_{\partial\Omega} |u_f - u_{mes}|^2, \tag{3.15}$$

where u_f is the solution of the following Poisson's equation with the source term f:

$$\begin{cases} -\triangle u = f \ \text{ in } \Omega, \\ \partial_n u = 0 \ \text{ on } \partial\Omega, \\ \int_\Omega u = 0. \end{cases} \tag{3.16}$$

The problem is well posed when $\int_\Omega f = 0$.

The associated variational problem is:

$$\begin{cases} \text{find } u \in H^1(\Omega) \text{ such as} \\ a(u,v) = l^f(v), \ \forall v \in H^1(\Omega), \end{cases} \tag{3.17}$$

with $\forall u, v \in H^1(\Omega)$

$$a(u,v) = \int_\Omega \nabla u \nabla v,$$

$$l^f(v) = \int_\Omega fv.$$

The associated Lagrangian is

$$\mathcal{L}(f,u,p) = J(u) + a(u,v) - l^f(v).$$

We have

$$j(f) = \mathcal{L}(f, u_f, p).$$

$$\partial_f j(f)\delta f = \partial_f \mathcal{L}(f,u_f,p)\delta f + \partial_u \mathcal{L}(f,u_f,p)\partial_f u_f \delta f.$$

The equation $\partial_u \mathcal{L}(f,u_f,p) = 0$ gives the following adjoint equation of solution p_f:

$$\begin{cases} -\triangle p_f = 0 \ \text{ in } \Omega, \\ \partial_n p_f = u_f - u_{mes} \ \text{ on } \partial\Omega. \end{cases} \tag{3.18}$$

We have

$$\partial_f j(f)\delta f = -\int_\Omega \delta f p_{u_f} \tag{3.19}$$

We are looking for a dipolar source, let $f = 0$ in Ω and formally $\delta f = \varepsilon \mathbf{d}.\nabla\delta_{x_0}$ is the derivative of a Dirac function in the direction $\mathbf{d}$ in Ω. The general idea is to use the extension of the adjoint method, presented in Chapter 2, to find the development of $\partial_f j(f)$. We will study the cases in two and three dimensions.

Three dimension case

The elementary solution of the Laplacian, called G_0, which satisfies the following equation:

$$-\triangle G_0 = \delta_{x_0},$$

is explicitly given by

$$G_0(x) = \frac{1}{4\pi}\frac{1}{|x - x_0|}.$$

We are looking for an analytical solution G_1 which verifies

$$\begin{aligned} -\triangle G_1 &= \mathbf{d}.\nabla\delta_{x_0}, \\ -\triangle G_1 &= \mathbf{d}.\nabla(-\triangle G_0). \end{aligned}$$

where $\mathbf{d} = (d_1, d_2, d_3)$. This solution is

$$G_1 = \mathbf{d}.\nabla G_0 = -\frac{1}{4\pi}\frac{\mathbf{d}.(x - x_0)}{|x - x_0|^3},$$

where $\nabla G_0 = (\partial_1 G_0(x), \partial_2 G_0(x), \partial_3 G_0(x))$, with $i \in \{1, 2, 3\}$,

$$\partial_i G_0(x) = -\frac{1}{4\pi}\frac{(x_i - x_{0i})}{|x - x_0|^3}.$$

In order to use a correct variational formulation, the domain truncation is used. The study is performed in the complementary of a small domain D containing the singularity. We consider a domain $\Omega \subset \mathbb{R}^3$. Let $\Omega_0 = \Omega \diagdown D$ where D is the sphere of center $x_0 = (x_{01}, x_{02}, x_{03})$ and of radius R, containing the singularity.

Let σ represents the perturbed case, it depends on ε, $\mathbf{d}$ and x_0.

$$\begin{cases} -\triangle u = \varepsilon\mathbf{d}.\nabla\delta_{x_0} \text{ in } D, \\ \quad u = \psi_\sigma \text{ on } \partial D. \end{cases} \tag{3.20}$$

The truncated perturbed problem is the following:

$$\begin{cases} -\triangle u_\sigma = 0 \text{ in } \Omega_0, \\ \partial_n u_\sigma = 0 \text{ on } \partial\Omega, \\ \quad u_\sigma = \psi_\sigma \text{ on } \partial D. \end{cases} \tag{3.21}$$

In the non perturbed case, the solution is $u_0 = 0$ and p_0 is the solution of the associated adjoint equation.

We are looking for a variational formulation of the truncated problem. A second way to demonstrate the following results is to use a Green-Riemann formula.

$$a(u, v) = a_{\Omega_0}(u, v) + a_D(u, v) = \int_{\Omega_0} \nabla u \nabla v + \int_D \nabla u \nabla v.$$

The following relation have a meaning for $v \in D(\overline{\Omega})$, the space of distribution functions,

$$\int_D \nabla u \nabla v = -\int_D u \triangle v + \int_\Sigma u \partial_n v.$$

If $v = p_0$ then $a_D(u, p_0) = \int_\Sigma u \partial_n p_0$.

Let u_ψ be the solution ot the following variational problem:

$$\begin{cases} \text{Find } u \in H^1(D) \text{ solution of} \\ a_D(u, v) = 0 \quad \forall v \in H_0^1(D), \\ u = \psi \text{ on } \Sigma. \end{cases}$$

Let a_Σ be the following bilinear form:

$$a_\Sigma : H^{1/2}(\Sigma) \times H^{1/2}(\Sigma) \to \mathbb{R},$$
$$(\phi, \psi) \mapsto a_\Sigma(\phi, \psi) = a_D(u_\phi, u_\psi) = -\int_D u_\phi \triangle u_\psi + \int_\Sigma u_\phi \partial_n u_\psi.$$

The following equations have a meaning for $v \in D(\overline{\Omega})$

$$\begin{aligned} l^\sigma(v) &= \int_\Omega \varepsilon \mathbf{d}.\nabla \delta_{x_0}\ v, \\ l^\sigma_{\Omega_0}(v) + l^\sigma_D(v) &= \int_{\Omega_0} \varepsilon \mathbf{d}.\nabla \delta_{x_0}\ v + \int_D \varepsilon \mathbf{d}.\nabla \delta_{x_0}\ v, \\ l^\sigma_{\Omega_0}(v) &= 0, \\ l^\sigma_D(v) &= \int_D \varepsilon \mathbf{d}.\nabla \delta_{x_0}\ v. \end{aligned}$$

Let G_R be the following function:

$$G_R(x) = -\frac{1}{4\pi}\frac{\mathbf{d}.(x - x_0)}{|x - x_0|^3} + \frac{1}{4\pi}\frac{\mathbf{d}.(x - x_0)}{R^3}.$$

The function εG_R is the solution of the following problem:

$$\begin{cases} -\triangle u = \varepsilon \mathbf{d}.\nabla \delta_{x_0} \text{ in } D, \\ u = 0 \text{ on } \partial D. \end{cases} \tag{3.22}$$

$$\begin{aligned} & l^\sigma_D(v) = \int_D \varepsilon \mathbf{d}.\nabla \delta_{x_0}\ v, \\ \Leftrightarrow\ & l^\sigma_D(v) = -\varepsilon \int_D \triangle G_R\ v, \\ \Leftrightarrow\ & l^\sigma_D(v) = \varepsilon \int_D \nabla G_R\ \nabla v - \varepsilon \int_\Sigma \partial_n G_R\ v, \\ \Leftrightarrow\ & l^\sigma_D(v) = -\varepsilon \int_D G_R\ \triangle v - \varepsilon \int_\Sigma \partial_n G_R\ v. \end{aligned}$$

The previous formula is continuous on $H^1(\Omega)$ and by density, it gives a meaning to the linear form defined by the dipolar source term.

$$\begin{aligned} l_\Sigma^\sigma : H^{1/2}(\Sigma) &\to \mathbb{R} \\ \phi &\mapsto l_\Sigma^\sigma(\phi) = l_D^\sigma(u_\phi) = -\varepsilon \int_D G_R \, \triangle u_\phi - \varepsilon \int_\Sigma \partial_n G_R \, u_\phi \end{aligned}$$

If $v = p_0$ then $l_D^\sigma(p_0) = -\varepsilon \int_\Sigma \partial_n G_R \, p_0$.

The variation of the problem associated with the truncated problem is

$$a_{\Omega_0}(u, v) + a_\Sigma(u, v) = l_\Sigma^\sigma(v), \qquad \forall v \in H^1(\Omega_0). \tag{3.23}$$

The Lagrangian of the truncated problem is

$$\mathcal{L}_{\Omega_0}(\sigma, u, v) = J(u) + a_{\Omega_0}(u, v) + a_\Sigma(u, v) - l_\Sigma^\sigma(v).$$

The function J has the following expansion:

$$\begin{aligned} J(u_\sigma) - J(u_0) &= \partial_u J(u_0; u_\sigma - u_0) + \frac{1}{2}\partial_u^2 J(u_0; u_\sigma - u_0, u_\sigma - u_0), \\ \Leftrightarrow J(u_\sigma) - J(u_0) &= \partial_u J(u_0; u_\sigma - u_0) + \frac{1}{2}\|u_\sigma - u_0\|^2_{L^2(\partial\Omega)}, \\ \Leftrightarrow J(u_\sigma) - J(u_0) &= \partial_u J(u_0; u_\sigma - u_0) + \frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)}. \end{aligned}$$

Thanks to the adjoint equation, we have:

$$a_{\Omega_0}(u_\sigma - u_0, p_0) + a_\Sigma(u_\sigma - u_0, p_0) = -\partial_u J(u_0; u_\sigma - u_0).$$

where p_0 is the solution of the adjoint equation without perturbation.

The variation of the cost function is the following:

$$\begin{aligned} & j(\sigma) - j(0) \\ =& \mathcal{L}_{\Omega_0}(\sigma, u_\sigma, p_0) - \mathcal{L}_{\Omega_0}(0, u_0, p_0) \\ =& J(u_\sigma) - J(u_0) + a_{\Omega_0}(u_\sigma - u_0, p_0) + a_\Sigma(u_\sigma - u_0, p_0) - l_\Sigma^\sigma(p_0) \\ =& \partial_u J(u_0; u_\sigma - u_0) + a_{\Omega_0}(u_\sigma - u_0, p_0) + a_\Sigma(u_\sigma - u_0, p_0) - l_\Sigma^\sigma(p_0) + \frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)} \\ =& - l_\Sigma^\sigma(p_0) + \frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)} \\ =& \varepsilon \int_\Sigma \partial_n G_R \, p_0 + \frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)}. \end{aligned}$$

$$\nabla G_R = (\partial_1 G_R(x), \partial_2 G_R(x), \partial_3 G_R(x)),$$

where

$$\partial_i G_R(x) = \frac{1}{4\pi}\left(\frac{d_i}{R^3} - \frac{d_i}{|x-x_0|^3} + \frac{\sum_{j=1}^{3} 2d_j(x_j - x_{0j})(x_i - x_{0i})}{|x-x_0|^4}\right),$$

in polar coordinates, $\mathbf{n} = (\cos(\theta)\sin(\phi), \sin(\theta)\sin(\phi), \cos(\phi))$ and after simplification, we obtain

$$\partial_n G_R(R,\theta,\phi) = \frac{1}{\pi}\left(\frac{\cos(\theta)\sin(\phi)d_1 + \sin(\theta)\sin(\phi)d_2 + \cos(\phi)d_3}{R^3}\right).$$

$$\begin{aligned}\int_\Sigma p_0 \partial_n G_R d\sigma &= \int_0^\pi \int_0^{2\pi} p_0(R,\theta,\phi)\partial_n G_R(R,\theta,\phi)R^2 d\theta d\phi \\ &= \int_0^\pi \int_0^{2\pi} p_0(R,\theta,\phi)\frac{\cos(\theta)\sin(\phi)d_1 + \sin(\theta)\sin(\phi)d_2 + \cos(\phi)d_3}{\pi R} d\theta d\phi.\end{aligned}$$

The following relations $\int_0^\pi \int_0^{2\pi} \cos(\theta)\sin(\phi)d\theta d\phi = 0$, $\int_0^\pi \int_0^{2\pi} \sin(\theta)\sin(\phi)d\theta d\phi = 0$ and $\int_0^\pi \int_0^{2\pi} \cos(\phi)d\theta d\phi = 0$ allow to modify the previous equation:

$$\begin{aligned}&\int_\Sigma p_0 \partial_n G_R d\sigma = \\ &\frac{1}{\pi}\int_0^\pi \int_0^{2\pi} \frac{p_0(R,\theta,\phi) - p_0(0)}{R}\left(\cos(\theta)\sin(\phi)d_1 + \sin(\theta)\sin(\phi)d_2 + \cos(\phi)d_3\right) d\theta d\phi.\end{aligned}$$

When R approaches 0, the limit of $\frac{p_0(R,\theta,\phi)-p_0(0)}{R}$ is $\partial_r p_0(0) = n.\nabla p_0(x_0)$ where

$$\begin{aligned}\int_\Sigma p_0 \partial_n G_R d\sigma &= \frac{d_1\partial_1 p_0(x_0)}{\pi}\int_0^\pi \int_0^{2\pi} \cos^2(\theta)\sin^2(\phi)d\theta d\phi \\ &+ \frac{d_2\partial_2 p_0(x_0)}{\pi}\int_0^\pi \int_0^{2\pi} \sin^2(\theta)\sin^2(\phi)d\theta d\phi \\ &+ \frac{d_3\partial_3 p_0(x_0)}{\pi}\int_0^\pi \int_0^{2\pi} \cos^2(\phi)d\theta d\phi \\ &= \mathbf{d}.\nabla p_0(x_0).\end{aligned}$$

The variation of the cost function has the following expansion:

$$j(\sigma) - j(0) = \varepsilon \mathbf{d}.\nabla p_0(x_0) + \frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)}.$$

Remark 7. A formal calculus from Equation (3.19) would have given the following result:

$$j(\sigma) - j(0) = \varepsilon \mathbf{d}.\nabla p_0(x_0) + o(\varepsilon^2)$$

without the term $\frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)}$ which will be useful in the numerical part.

Two dimension case

In two dimensions, the variation of the cost function remains the same:

$$j(\sigma) - j(0) = \varepsilon \mathbf{d}.\nabla p_0(x_0) + \frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)}.$$

The previous calculus can be adapted with the following functions:

$$G_0(x) = -\frac{1}{2\pi} log(|x - x_0|),$$

$$G_1(x) = -\frac{1}{2\pi}\frac{\mathbf{d}.(x - x_0)}{|x - x_0|^2},$$

$$G_R = -\frac{1}{2\pi}\frac{\mathbf{d}.(x - x_0)}{|x - x_0|^2} + \frac{1}{2\pi}\frac{\mathbf{d}.(x - x_0)}{R^2}.$$

3.5.3. Cost function on the domain

The cost function

The study of a cost function defined on all the domain can provide more information and numerical stability. There are both Dirichlet and Neumann conditions on the boundary, two direct problems can be defined in all the domain. The cost function over the entire domain is the difference between the two functions. This idea was firstly proposed by Kohn and Vogelius [69].

Let σ denoted the perturbed case, it depends on ε, $\mathbf{d}$ and x_0. Since the boundary conditions on $\partial\Omega$ are known both in the Dirichlet and the Neumann cases in a part of, we can define two forward problems: the Dirichlet problem 3.24 and the Neumann problem 3.25.

$$\begin{cases} -\triangle u_D^\sigma = \varepsilon \mathbf{d}.\nabla \delta_{x_0} \ \text{ in } \Omega, \\ \quad u_D^\sigma = u_{mes} \ \text{ on } \Omega. \end{cases} \tag{3.24}$$

$$\begin{cases} -\triangle u_N^\sigma = \varepsilon \mathbf{d}.\nabla \delta_{x_0} \ \text{ in } \Omega, \\ \partial_n u_N^\sigma = 0 \ \text{ on } \partial\Omega, \\ \displaystyle\int_\Omega u_N^\sigma = 0. \end{cases} \tag{3.25}$$

These two problems have the following variational definitions:

$$\begin{cases} \text{Find } u_D^\sigma \in \tilde{u} + H_0^1(\Omega) \text{ such as} \\ a(u_D^\sigma, v) = l^\sigma(v), \quad \forall v \in H_0^1(\Omega), \end{cases}$$

where $\tilde{u}$ is a lifting of u_{mes} in all the domain and

$$\begin{cases} \text{Find } u_N^\sigma \in H^1(\Omega) \text{ such as} \\ a(u_N^\sigma, v) = l^\sigma(v), \quad \forall v \in H^1(\Omega), \end{cases}$$

where $\forall u, v \in H^1(\Omega)$,

$$a(u,v) = \int_\Omega \nabla u \nabla v,$$
$$l^\sigma(v) = \varepsilon \int_\Omega \mathbf{d}.\nabla \delta_{x_0} v.$$

The dipolar source is found when there is no difference between the two functions, the cost function is the following:

$$j(\sigma) = J(u_D^\sigma, u_N^\sigma) = \frac{1}{2}\|u_D^\sigma - u_N^\sigma\|^2_{L^2(\Omega)}. \tag{3.26}$$

It is related to the Kohn-Vogelius criterion [69].

Three dimension case

By definition of J, we have

$$\begin{aligned} \partial_1 J(u_D^\sigma, u_N^\sigma) &= u_D^\sigma - u_N^\sigma, \\ \partial_2 J(u_D^\sigma, u_N^\sigma) &= -u_D^\sigma + u_N^\sigma, \end{aligned} \tag{3.27}$$

and the adjoint equations are given by the following equations:

$$\begin{aligned} a(\psi, p_D^\sigma) &= -\partial_1 J(u_D^\sigma, u_N^\sigma; \psi), \quad \forall \psi \in H_0^1(\Omega), \\ a(\psi, p_N^\sigma) &= -\partial_2 J(u_D^\sigma, u_N^\sigma; \psi), \quad \forall \psi \in H^1(\Omega). \end{aligned} \tag{3.28}$$

To compute the variation of the cost function, we need to solve numerically the two direct problems having solutions u_D^0 and u_N^0.

$$\begin{cases} \triangle u_D^0 = 0 \ \text{ in } \Omega, \\ u_D^0 = u_{mes} \ \text{ on } \partial\Omega, \end{cases} \tag{3.29}$$

$$\begin{cases} \triangle u_N^0 = 0 \ \text{ in } \Omega, \\ \partial_n u_N^0 = 0 \ \text{ on } \partial\Omega, \\ \displaystyle\int_\Omega u_N^0 = 0, \end{cases} \tag{3.30}$$

and the two adjoint problems

$$\begin{cases} -\triangle p_D^0 = -u_D^0 \ \text{ in } \Omega, \\ p_D^0 = 0 \ \text{ on } \partial\Omega, \end{cases} \tag{3.31}$$

$$\begin{cases} -\triangle p_N^0 = u_D^0 - \overline{u_D} \text{ in } \Omega, \\ \partial_n p_N^0 = 0 \text{ on } \partial\Omega, \\ \int_\Omega p_N^0 = 0, \end{cases} \tag{3.32}$$

with $\overline{u_D} = \frac{1}{mes(\Omega)} \int_\Omega u_D^0 dx$.

The asymptotic expansion of the cost is the following

$$\begin{aligned} & j(\sigma) - j(0) \\ =& J(u_D^\sigma, u_N^\sigma) - J(u_D^0, u_N^0) \\ =& \partial_1 J(u_D, u_N; u_D^\sigma - u_D^0) + \partial_2 J(u_D, u_N; u_N^\sigma - u_N^0) + \frac{1}{2}\|u_D^\sigma - u_N^\sigma\|^2_{L^2(\Omega)} \\ =& - a(u_D^\sigma - u_D^0, p_D^0) - a(u_N^\sigma - u_N^0, p_N^0) + \frac{1}{2}\|u_D^\sigma - u_N^\sigma\|^2_{L^2(\Omega)} \\ =& - a(u_D^\sigma, p_D^0) + a(u_D^0, p_D^0) - a(u_N^\sigma, p_N^0) + a(u_N^0, p_N^0) + \frac{1}{2}\|u_D^\sigma - u_N^\sigma\|^2_{L^2(\Omega)} \\ =& - l_\sigma(p_D^0) - l_\sigma(p_N^0) + \frac{1}{2}\|u_D^\sigma - u_N^\sigma\|^2_{L^2(\Omega)} \\ =& \varepsilon \mathbf{d}.\nabla(p_D^0(x_0) + p_N^0(x_0)) + \frac{1}{2}\|u_D^\sigma - u_N^\sigma\|^2_{L^2(\Omega)}. \end{aligned}$$

Quadratic formula

Let us define $\mathbf{d}_\varepsilon = \varepsilon \mathbf{d} = (\alpha, \beta, \gamma)$, $u_D^\sigma = \alpha u_D^x + \beta u_D^y + \gamma u_D^z$, $u_N^\sigma = \alpha u_N^x + \beta u_N^y + \gamma u_N^z$. To shorten the writing, we define $u^e = u_D^e - u_N^e$ for $e = x$, y or z and $p_0 = p_D^0 + p_N^0$. Let φ be the following quadratic form:

$$\varphi(\mathbf{d}_\varepsilon, \mathbf{x}_0) = j(\omega) - j(0) = (\alpha, \beta, \gamma).\nabla p_0(\mathbf{x}_0) + \frac{1}{2}\int_\Omega (\alpha u^x + \beta u^y + \gamma u^z)^2 \tag{3.33}$$

For more than one dipolar source, we have:

$$\varphi(\mathbf{d}_i, x_i) = \sum_{i=1}^m (\alpha_i, \beta_i, \gamma_i).\nabla p_0(\mathbf{x}_i) + \frac{1}{2}\int_\Omega \left(\sum_{i=1}^m (\alpha_i u_i^x + \beta_i u_i^y + \gamma_i u_i^z)\right)^2. \tag{3.34}$$

If we want to find $\mathbf{d}_\varepsilon$ which minimize the quadratic form (3.33) with a dipolar source at the position x_1, we have to solve the following system:

$$Q_1.\mathbf{d}_\varepsilon = -\nabla p_0(\mathbf{x}_0),$$

where Q_1 is the following matrix:

$$Q_1 = \begin{pmatrix} \int_\Omega u^x u^x & \int_\Omega u^x u^y & \int_\Omega u^x u^z \\ \int_\Omega u^x u^y & \int_\Omega u^y u^y & \int_\Omega u^y u^z \\ \int_\Omega u^x u^z & \int_\Omega u^y u^z & \int_\Omega u^z u^z \end{pmatrix}.$$

If we want to find $\mathbf{d}_1$ and $\mathbf{d}_2$ which minimize the quadratic form 3.34 with a dipolar source at the position x_1 and a dipolar source at the position x_2, we have to solve the following problem:

$$Q_2.\begin{pmatrix}\mathbf{d}_1\\ \mathbf{d}_2\end{pmatrix} = -\begin{pmatrix}\nabla p_0(\mathbf{x}_1)\\ \nabla p_0(\mathbf{x}_2)\end{pmatrix},$$

where

$$Q_2 = \begin{pmatrix}
\int_\Omega u_1^x u_1^x & \int_\Omega u_1^x u_1^y & \int_\Omega u_1^x u_1^z & \int_\Omega u_1^x u_2^x & \int_\Omega u_1^x u_2^y & \int_\Omega u_1^x u_2^z \\
\int_\Omega u_1^x u_1^y & \int_\Omega u_1^y u_1^y & \int_\Omega u_1^y u_1^z & \int_\Omega u_1^y u_2^x & \int_\Omega u_1^y u_2^y & \int_\Omega u_1^y u_2^z \\
\int_\Omega u_1^x u_1^z & \int_\Omega u_1^y u_1^z & \int_\Omega u_1^z u_1^z & \int_\Omega u_1^z u_2^x & \int_\Omega u_1^z u_2^y & \int_\Omega u_1^z u_2^z \\
\int_\Omega u_2^x u_1^x & \int_\Omega u_2^x u_1^y & \int_\Omega u_2^x u_1^z & \int_\Omega u_2^x u_2^x & \int_\Omega u_2^x u_2^y & \int_\Omega u_2^x u_2^z \\
\int_\Omega u_2^x u_1^y & \int_\Omega u_2^y u_1^y & \int_\Omega u_2^y u_1^z & \int_\Omega u_2^y u_2^x & \int_\Omega u_2^y u_2^y & \int_\Omega u_2^y u_2^z \\
\int_\Omega u_2^x u_1^z & \int_\Omega u_2^y u_1^z & \int_\Omega u_2^z u_1^z & \int_\Omega u_2^z u_2^x & \int_\Omega u_2^z u_2^y & \int_\Omega u_2^z u_2^z
\end{pmatrix}.$$

The matrices Q are symmetric.

3.6. Numerical results in two dimensions

For all the numerical tests, the data on the boundary is generated by a dipolar source on a fine mesh of 32768 elements. The dipolar sources are positioned within the disk centred in the origin and of radius 0.85. When many tests are performed, the direction is random. The amplitude is equal to 0.01. The noise is an additive Gaussian noise with a variance equals to a fraction of the maximum of the data. The localization is made on a mesh of 8192 elements for the first algorithm and 2048 elements for the rest.

3.6.1. Algorithms based on the boundary dependant cost function

The variation of the cost function is:

$$j(\sigma) - j(0) = \varepsilon \mathbf{d}.\nabla p_0(x_0) + \frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)}$$

If we assume that σ is the correct solution, we have $j(\sigma) = 0$, $j(0) = \frac{1}{2}\|u_{mes}\|^2_{L^2(\partial\Omega)}$, $\frac{1}{2}\|u_\sigma\|^2_{L^2(\partial\Omega)} = \frac{1}{2}\|u_{mes}\|^2_{L^2(\partial\Omega)}$. The aim is to find the location x_0 which minimizes $\varepsilon \mathbf{d}.\nabla p_0(x_0) + \|u_{mes}\|^2_{L^2(\partial\Omega)}$

The first test has been performed by supposing known the amplitude and the direction of the dipolar source. The aim is to find the minimum of the function $|\varepsilon \mathbf{d}.\nabla p_0(x) + \|u_{mes}\|^2_{L^2(\partial\Omega)}|$. The problem is that the adjoint function p_0 is harmonic, this implies that

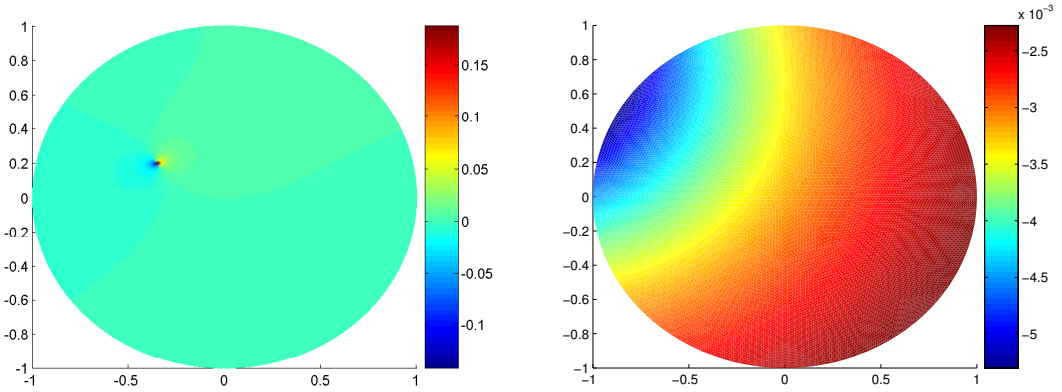


Figure 3.8.: From left to right, the dipole, the function $\mathbf{d}.\nabla p_0$.

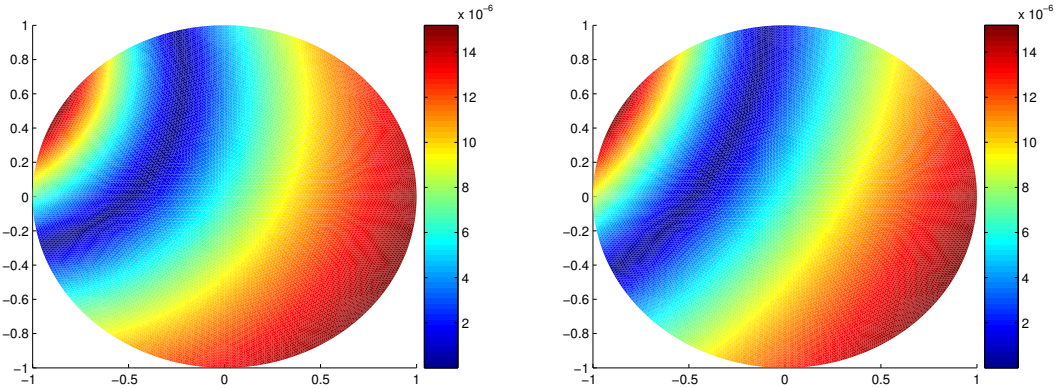


Figure 3.9.: Contour line of potential solutions.

$\mathbf{d}.\nabla p_0$ is also harmonic, see Figure 3.8. We obtain a contour line of possible solutions with this approach, see Figure 3.9 left. The location of the dipolar source is a point belonging to this contour line and the farthest of the boundary.

The second test has been performed by supposing known the amplitude and by using the direction of steepest descent. The function to minimize becomes:

$$\left| -\varepsilon \frac{\nabla p_0(x)}{\sqrt{\nabla p_0(x)^2}} . \nabla p_0(x) + \|u_{mes}\|^2_{L^2(\partial\Omega)} \right| .$$

The problem stays the same. One more time, a contour line of possible solutions is obtained with this method, see Figure 3.9 right.

The following approach comes from the fact that the minimum of the harmonic function

$$-\varepsilon \frac{\nabla p_0(x)}{\sqrt{\nabla p_0(x)^2}} . \nabla p_0(x)$$

Creation region	$0.95 \geqslant r > 0.66$	$0.66 \geqslant r > 0.33$	$0.33 \geqslant r$
Relative orientation error	0.0567	0.0126	0.0033

Table 3.3.: Relative orientation error for 500 cases in three different regions.

on the boundary of the disk gives a similar orientation of our dipolar source. This result has been checked empirically. Table 3.3 gives the relative error between the estimated orientation and the real orientation on the mean of 500 experiments in three different regions. The variable r is the distance of the disk origin to the dipole location.

By taking into account this estimated direction and looking for the location which could have this direction, a cone of potential solutions is obtained, see Figure 3.10. The following equation allows to obtain an estimation of ε :

$$\varepsilon = \frac{\|u_{mes}\|^2_{L^2(\partial\Omega)}}{\mathbf{d}.\nabla p_0(x)}$$

Then, in each location of the cone, each of associated u_σ is computed with the estimated ε and the estimated direction. The solution corresponds to the configuration which gives the nearest replica of the data. The resulting dipolar source is given in Figure 3.10.

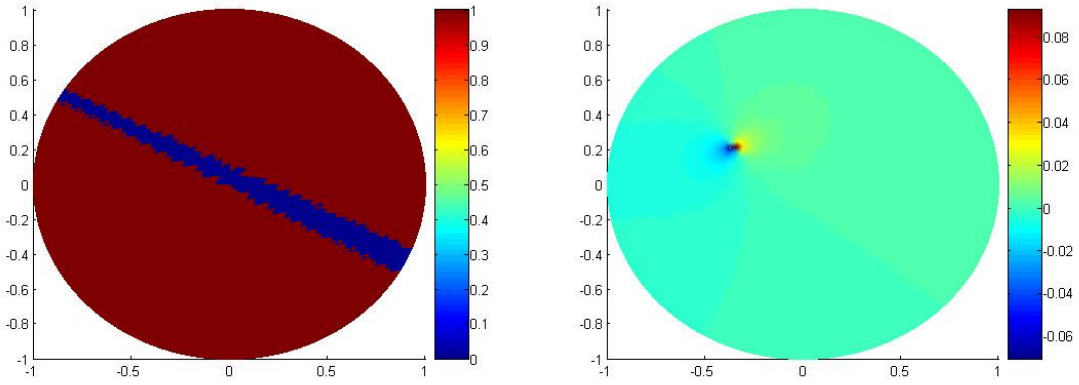


Figure 3.10.: Cone of potential solutions and the final location.

Up to now, only one dipole is sought with the assumption that u_0 is a non perturbed state. The same reasoning can be made by supposing that $n-1$ dipolar sources are known and one dipolar source is sought as in Figure 3.11.

3.6.2. Algorithms based on the decomposition of the dipolar source

In this section, we compare the results of three different algorithms which are all based on the decomposition of the dipolar source into three components. The first algorithm

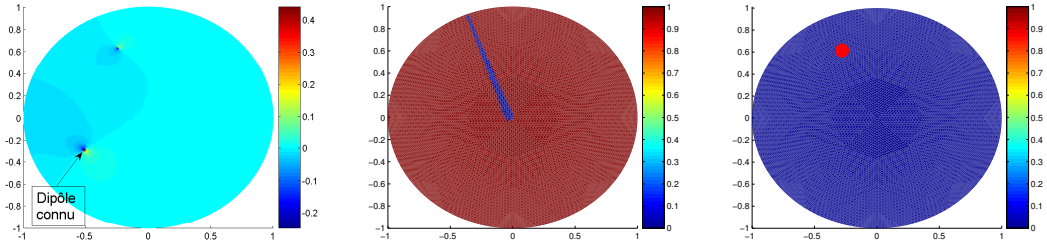


Figure 3.11.: From left to right, known and unknown dipolar sources, cone of the potential solutions, localisation of the unknown dipolar source.

is based on the resolution of the quadratic form given by the cost function on the boundary. (The study to obtain it is similar to Section 3.5.3) The second one is based on the resolution of the quadratic form given by the cost function on the domain, see Equation 3.33. The third algorithm uses the following approach: the first step is to create, for each finite element, some data V_x and V_y containing the information on the boundary created by unit dipoles in the abscissa direction and in the ordinate direction respecively; the second step is, for each finite element, to project the data on the boundary V_{mes} on the plane generated by V_x and V_y.

5% of noise	$\leqslant 2 * lcar$	2 à $4 * lcar$	4 à $6 * lcar$	$> 6 * lcar$	orientation error
1^{st} method	50	40	10	0	0.069
2^{nd} method	86	14	0	0	0.021
3^{rd} method	92	8	0	0	0.025
10% of noise	$\leqslant 2 * lcar$	2 à $4 * lcar$	4 à $6 * lcar$	$> 6 * lcar$	orientation error
1^{st} method	41	40	17	2	0.079
2^{nd} method	73	26	0	0	0.040
3^{rd} method	77	23	0	0	0.039
20% of noise	$\leqslant 2 * lcar$	2 à $4 * lcar$	4 à $6 * lcar$	$> 6 * lcar$	orientation error
1^{st} method	39	47	10	4	0.101
2^{nd} method	45	46	9	0	0.070
3^{rd} method	51	43	5	1	0.070

Table 3.4.: Dipolar source localization results.

Table 3.4 gives the results of 100 tests realized with 5%, 10% and 20% of noise. The error on the localization is expressed as a fraction of the characteristic length and the L^2 relative error is used to quantify the quality of the dipole orientation. Let $lcar$ be the characteristic length, it is the radius of the finite element if we consider it like a disk. The mesh is made of $8\,092$ elements, the characteristic length $lcar$ is equal to 0.0221 and the domain Ω is a disk of radius 1. In the best of cases, the localization error is of the order of the length of a mesh element. The localization results are given as a function of this length $lcar$.

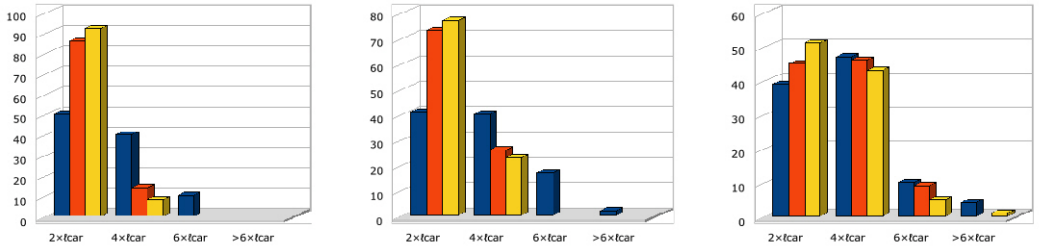


Figure 3.12.: From left to right, graphics with 5%, 10% and 20% of noise.

The following results are obtained with the cost function simular to the Kohn-Vogelius criterion. In the case of two dipolar sources, if they are separated by a certain distance, two zones of interest are obtained by looking for only one dipolar source. In Figure 3.13, the two dipolar sources are represented on the left and the result obtained by looking only for one dipolar source is shown on the right, there are two zones of interest. However in Figure 3.14 if there is a small distance between the two, only one dipolar source is detected. It is a drawback of this approach.

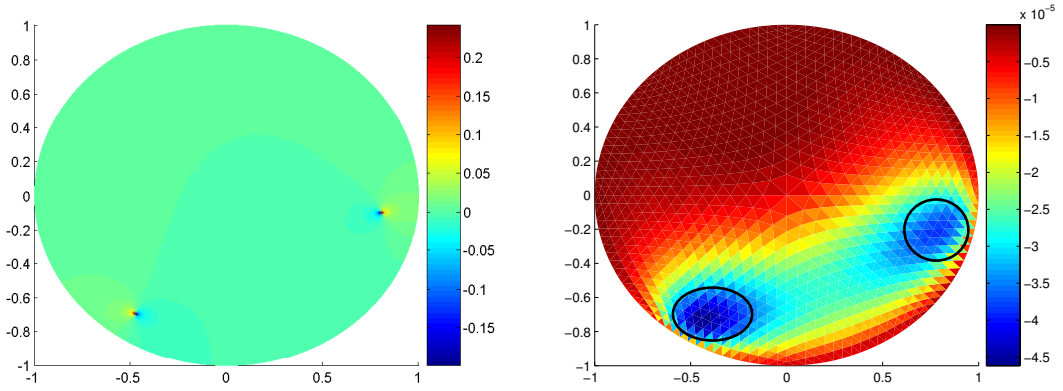


Figure 3.13.: From left to right, two dipolar sources, good result by only looking for one source.

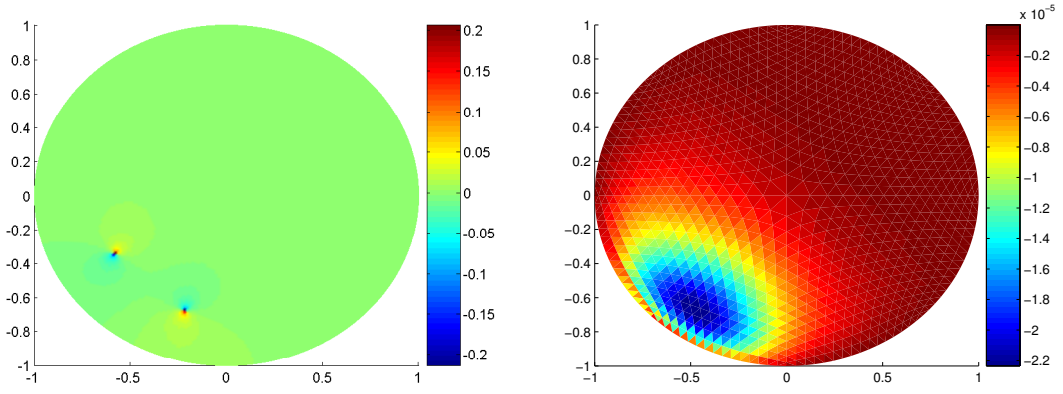


Figure 3.14.: From left to right, two dipolar sources, bad result by only looking for one source.

3.7. Numerical results in three dimensions

3.7.1. Numerical considerations

In Dipole Simulator, the data is generated with the standard configuration of 81 captors. Different captors configurations were tested in correspondence with the ones provided by Dipole Simulator. This one was chosen due to its better coverage zone. The dipolar sources were chosen within the brain.

Before solving the inverse problem, the components of the matrices Q can be computed and stocked. Once the components are computed for the interactions of two dipolar sources, we have the components required to compute matrices for more than two dipolar sources. This stage is time-costly but afterwards there is no need to repeat it and the resolution of the inverse problem is fast. We use a hierarchical method, the algorithm looks for dipolar sources on a mesh then the search is made with a finer mesh only in the interest zones obtained with the previous one.

Our head model is spherical and made of three layers with the same conductivities and dimensions than those in Dipole Simulator. The CSF in Dipole Simulator is considered as the brain by adjusting its conductivity to 0.33. The noise is the one provided in Dipole Simulator, and we vary the rms noise.

There has been some debate over the relative accuracy of EEG or MEG based source localization, for a review see [72]. The precision of EEG is in the order of 1 to 3 cm for the spherical head model and in the order of 1 cm for a realistic head model [48]. The simulated data used for our experiments come from a spherical model which fits our mesh, we suppose that the result should be in the order of 1 cm.

The zone where the potential is known is not limited to a mesh point, it is a zone of 1 cm to represent the captor. A hierarchical method is used with two different meshes, one with locations with a spacing of 20 mm and one with a spacing of 10 mm. The time is taken into consideration during the process. A time window is taken around a peak in the EEG data, the dipole is supposed to have a fixed location and orientation.

3.7.2. Results

Table 3.5 presents the results for one dipolar source with one amplitude phase (a positive peak) and two amplitude phases (positive and negative peaks). The 20 cases have an alpha proportion of 0.50. The results show that one dipolar source can be located even with noise.

The MATLAB code takes nearly 30 seconds to provide a result with one dipolar source.

One amplitude phase			Two amplitude phases	
location error (mm)	orientation error	rms noise (μv)	location error (mm)	orientation error
$7,06$	$0,081$	0	$7,06$	$0,153$
$7,75$	$0,114$	10	$7,55$	$0,198$
$9,00$	$0,202$	20	$8,64$	$0,253$

Table 3.5.: Results of 10 tests with one amplitude phase and 10 tests with two amplitude phases.

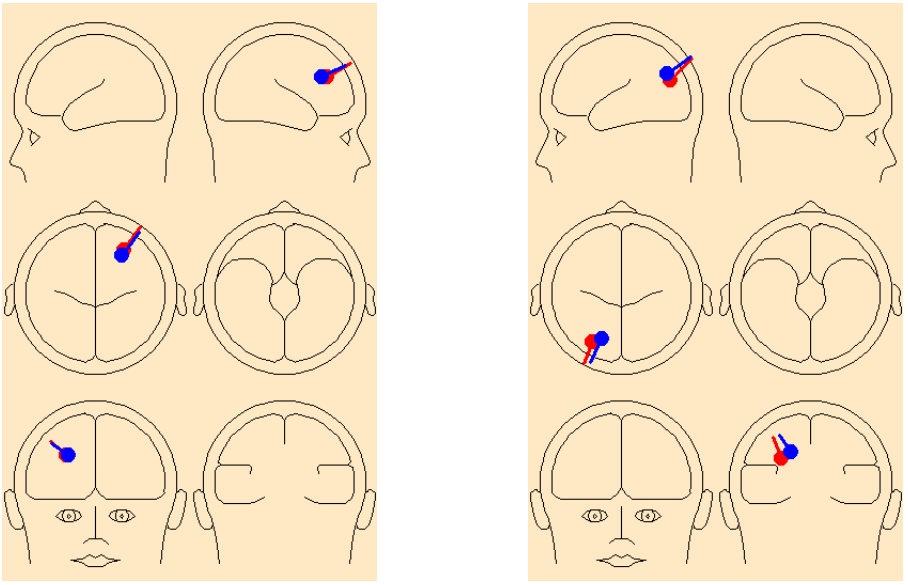


Figure 3.15.: Two results with one dipolar source and a rms noise of $20\mu v$.

Figure 3.15 presents two examples with a rms noise of $20\mu v$. The original dipolar sources were represented in red and the results in blue with the Dipole Simulator graphic interface. The left test has the location $(31,46,28)$ and the orientation $(0.50,0.74,0.45)$, the result has the location $(29,40,29)$ and the orientation $(0.54,0.75,0.37)$. The location error is 6.4 mm and the orientation error is 0.09. The right test has the location $(37,4,14)$ and the orientation $(0.18,0.74,0.65)$, the result has the location $(29,9,10)$ and the orientation $(0.22,0.79,0.56)$. The location error is 8.8 mm and the orientation error is 0.075.

It is to be reckoned that some cases of two dipolar sources may be problematic. The main problems arise with deep sources and symmetric sources. The last case can be solved by supposing symmetric a priori. Table 3.6 whows that when two dipolar sources are not deep and not symmetric, they can be detected even with noise.

rms noise (μv)	location error (mm)	orientation error
0	$10,52$	$0,257$
10	$9,32$	$0,308$
20	$13,83$	$0,346$

Table 3.6.: Results of 10 tests with one amplitude phase.

The MATLAB code takes nearly 350 seconds to provide a result with two dipolar sources. Figure 3.16 presents two examples with a rms noise of $20\mu v$. The original dipolar sources are represented in red and green and the results in blue and brown with the Dipole Simulator graphic interface. The left test have the locations $(-37,42,26)$ and $(34,-46,21)$; and respectively the orientations $(-0.69,-0.72,0.00)$ and $(0.67,0.74,-0.09)$, the results have the locations $(-41,41,21)$ and $(40,-42,20)$; and respectively the orientation $(-0.70,-0.71,0.03)$ and $(0.73,0.67,-0.10)$. The location error are 6.4 mm and 7.3 mm and the orientation error are 0.03 and 0.09. The right test have the locations $(21,-4,21)$ and $(-52,-29,-7)$; and respectively the orientations $(0.64,0.74,0.19)$ and $(-0.87,-0.50,0.00)$, the results have the locations $(39,-10,21)$ and $(-49,-32,-1)$; and respectively the orientation $(0.39,0.88,0.26)$ and $(-0.77,-0.62,-0.16)$. The location error are 18.9 mm and 7.3 mm and the orientation error are 0.29 and 0.22.

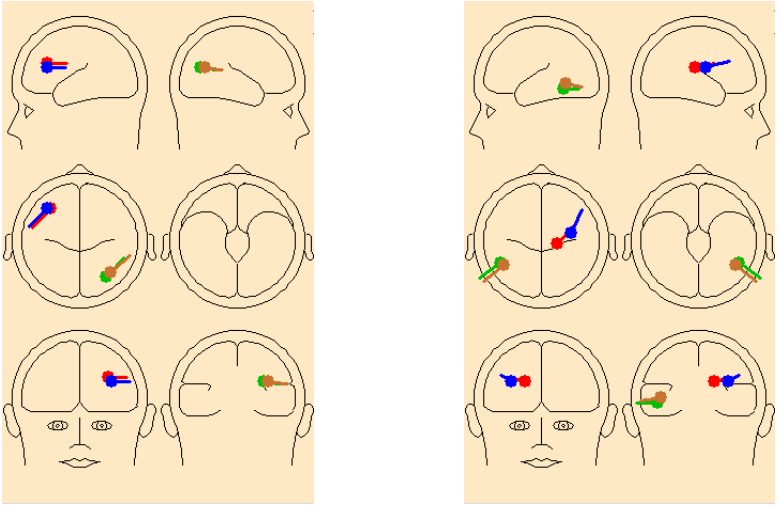


Figure 3.16.: Two results with two dipolar sources and a rms noise of $20\mu v$.

Figure 3.17 presents the same test with two symmetric dipolar sources treated in two different ways with a rms noise of $20\mu v$. The original dipolar sources are represented in red and green and the results in blue and brown with the Dipole Simulator graphic interface. The left test presents the results without a priori. The dipolar sources provided by the algorithm have wrong locations and orientations. The right test present the results with a priori on the symmetry. The dipolar sources have nearly the right location and only a part of the orientation. A priori on the orientation or a better Dirichlet condition could provide better results.

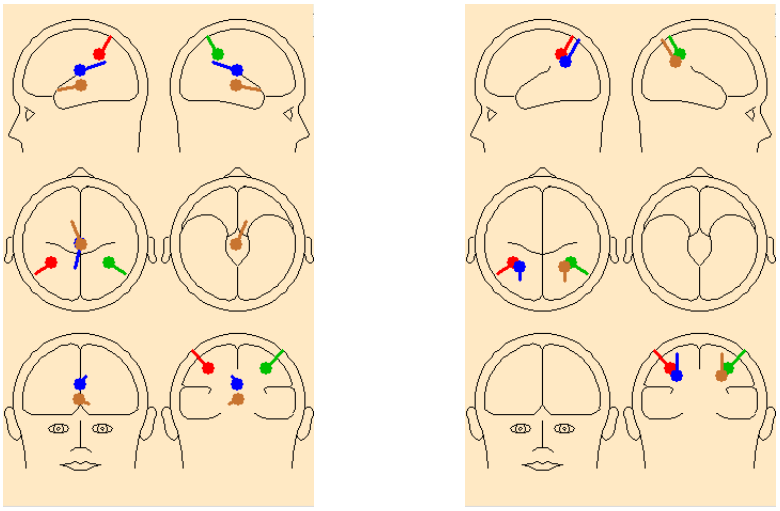


Figure 3.17.: Results with two symmetric dipolar sources and a rms noise of $20\mu v$.

3.8. Conclusion and further work

The objective of this chapter was to develop a non iterative approach without a priori to localize dipolar sources in electroencephalography.

In the theoretical parts, two cost functions have been studied: one on the boundary which is the classical misfit between the data and the estimated solution and the other on the domain which is related to the Kohn-Vogelius criterion. The developements of the cost functions provide their sensibility when a dipolar source is created. To perform the theoretical study, I use the extension of the adjoint method, presented in Chapter 2, to obtain a differentiable problem. The expansion highlight the importance of the derivative of order 2 which could not have been observed by a formal derivation of the cost functions.

The first algorithm presented in Section 3.6.1 is based on the cost function on the boundary. The result of the cost function expansion raises a problem due to the solution of the adjoint problem which is an harmonic function. The cost function on the domain has the same problem. The presented method can find a dipolar source by knowing all the other dipolar sources if they exist. This approach combined with a Particle Swarm Optimization algorithm, like in [99], could be used to detect the dipolar sources.

In Section 3.6.2, other approaches are based on the decomposition of the dipole in three components. A quadratic formula is obtained for the two cost functions. In two dimensions, these approaches give good results even in presence of noise. A hierarchical method can be used to improve the speed and avoid to compute for all elements. The comparative study in two dimensions illustrates that the approach related to the Kohn-Vogelius criterion gives interesting results. It is the one used in three dimensions.

The three dimension study shows that the proposed method is able to find a dipolar source location. Some problems begin to appear with two dipolar sources. Two dipolar sources cannot be detected when they are deep. When some symmetry occurs, a single dipolar source could be detected instead of two but a priori on the symmetry allows us to detect the two dipolar sources.

Using spherical splines to improve the Dirichlet condition could provide better results. Even with this improvement, the absence of a priori seems to be non reachable. Imposing reliable biophysical and psychological constraints has given superior results [54] in other methods, it could also improve our actual results by decreasing the numbers of possible configurations. The three-dimensional mesh could be finer with less locations. A parallelization could be implemented.

Conclusion and perspectives

This chapter analyzes the work done during this thesis where topology optimization techniques have been used to complete different research projects. It also presents future research lines to extend these results.

The image processing chapter presented a general approach to image processing using topological asymptotic analysis to perform edge detection. Many applications with new problems, demosaicing and super-resolution, were presented. I emphasized the importance of using all the information provided by the topological gradient to perform the anisotropic diffusion. The results are compared to existing algorithms using topological gradient and reference methods. The reconstructions show better or similar quality in terms of PSNR and SSIM. In inpainting applications, our method is better suited for geometry reconstruction. When texture is available, it should be combined with texture reconstruction algorithms. With Roberto Mecca, an Italian PhD student, we propose a general reconstruction algorithm which uses the nonlocal properties of fractional differentiation. This algorithm continues two previous works: the fractional derivative implementation from [24] and the edge detection by topological gradient from [27]. Compared to integer order derivative, better results are obtained and our algorithm is faster than the one proposed by [24].

With fractional derivatives, the information provided by the topological gradient is better suited for the integer order 1. The results are really better for this order and they are similar for other orders. The use of a topological gradient adapted to direct and adjoint equations with fractional orders should increase the efficiency. Future research should include other differential operators. When there is some self-similarity in the image, patch methods are more efficient. Non-local information could be combined at our approach by using anisotropic diffusion in the patches domain.

The numerical vault chapter presented an extension of the adjoint method. The theoretical and numerical parts justified and illustrated that the numerical vault captures higher order of the variation compared to the adjoint method. Another application of the vault method is the update of a solution of a singularly perturbed problem. In this case, the interest of the numerical vault is to avoid the resolution of the full direct perturbed equation, which can be costly in computing time, when only local zones are affected by modifications of the equation operator. It is a universal tool that can be applied to many linear problems as long as the hypothesis holds. Some examples in topological and continuous variations, mesh perturbations, elastography and image processing have been treated. Major advantages of the method are that the numerical vault is non invasive and can be used in different kinds of problems with a parallel computing implementation.

From the theoretical point of view, this method could be seen as an improvement of the domain truncation method, where the variation of the Dirichlet to Neumann operator is needed [75, 49, 14]. Here we just need to study the variation of a local solution. Our method could be a tool in theoretical investigations. The method is very simple and has great potential for applications. It could be used with multiscale algorithms: a finer mesh around the perturbation and a coarse mesh for the complementary. In addition to this, given some further development, it should find interesting applications in the area of semilinear equations.

The electroencephalography chapter presented new approaches to locate dipolar sources in two dimensions and three dimensions. The theoretical part used the extension of the adjoint method to find the developement of two cost functions. The study in two dimensions leads to choose a cost function on all the domain related to the Kohn-Vogelius criterion. The data has been simulated by Dipole Simulator, 3D meshes have been created with Gmsh and the MATLAB toolbox Getfem++ has been used for the implementation. In three dimensions, good results were obtained with one dipole, however difficulties appeared with two dipoles.

The absence of assumptions, which was one of the objectives, seems to be unattainable for the moment. In three dimensions, the use of spherical splines should provide a better Dirichlet boundary condition. Some assumptions on the dipolar sources should narrow down the possibilities. The quadratic formula with adjoint information is new, the question of its efficiency compared to other methods remains to be studied.

Appendices A and B presented numerical studies to improve diffusion schemes in image processing and to extend the denoising algorithm of Bai and Feng to color image. These appendices illustrate my efforts to improve my code implementation or other researchers' algorithms. Appendices C and D presented current research with physicians, physicists and biologists to perform segmentation of 3-D images or 1-D signals with the topological gradient method. These appendices illustrate the interests to confront with real data, to interact with people from other fields and to propose improvements in order to reach the objectives.

Future research includes the comparison of the difference element schemes with finite element mesh for image processing, the coupling of our method with level set methods for 3-D segmentation, the study of other differential operators and the resolution length for the topological gradient method in 1-D signals.

In a nutshell, some work has been done and some work still remains to be done. It would be wonderful if this thesis inspired its readers to pursue this work or to adapt it to other applications.

A. A comparative study of schemes for image processing

This appendix presents a comparative study of the schemes used in this thesis for resolving image processing problems. The first three sections present successively a scheme in each section: Scheme 1, Scheme 2 and Hymann-Shashkov's scheme. Numerical results and comparisons are given in the last section.

A.1. Scheme 1

Scheme 1 was used to restore images in the code that was adopted at the beginning of this work on image processing.

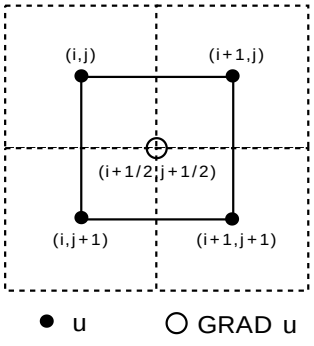


Figure A.1.: Stencil for operator GRAD.

$$(\nabla u)_{i+\frac{1}{2},j+\frac{1}{2}} = \left(\frac{u_{i+1,j} - u_{i,j} + u_{i+1,j+1} - u_{i,j+1}}{2}, \frac{u_{i,j+1} - u_{i,j} + u_{i+1,j+1} - u_{i+1,j}}{2} \right)$$

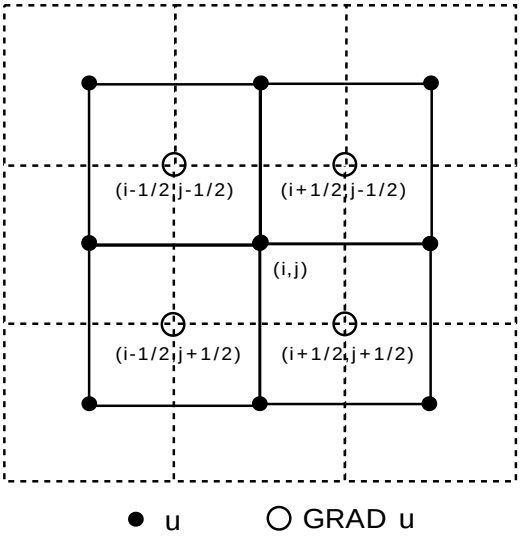


Figure A.2.: Stencil for operator DIV.

$$(\nabla.(g^1,g^2))_{i,j} = \begin{cases} \dfrac{g^1_{i+\frac{1}{2},j-\frac{1}{2}} - g^1_{i-\frac{1}{2},j-\frac{1}{2}} + g^1_{i+\frac{1}{2},j+\frac{1}{2}} - g^1_{i-\frac{1}{2},j+\frac{1}{2}}}{2} & \text{if } 1 < i < N \\ \dfrac{g^1_{i+\frac{1}{2},j-\frac{1}{2}} + g^1_{i+\frac{1}{2},j+\frac{1}{2}}}{2} & \text{if } i = 1 \\ \dfrac{-g^1_{i-\frac{1}{2},j-\frac{1}{2}} - g^1_{i-\frac{1}{2},j+\frac{1}{2}}}{2} & \text{if } i = N \end{cases}$$

$$+ \begin{cases} \dfrac{g^2_{i-\frac{1}{2},j+\frac{1}{2}} - g^2_{i-\frac{1}{2},j-\frac{1}{2}} + g^2_{i+\frac{1}{2},j+\frac{1}{2}} - g^2_{i+\frac{1}{2},j-\frac{1}{2}}}{2} & \text{if } 1 < j < N \\ \dfrac{g^2_{i-\frac{1}{2},j+\frac{1}{2}} + g^2_{i+\frac{1}{2},j+\frac{1}{2}}}{2} & \text{if } j = 1 \\ \dfrac{-g^2_{i-\frac{1}{2},j-\frac{1}{2}} - g^2_{i+\frac{1}{2},j-\frac{1}{2}}}{2} & \text{if } j = N \end{cases}$$

A.2. Scheme 2

The second discretization comes from Aubert and al's paper [15].

For pixels (i,j), where $i = 1,\ldots,m$ and $j = 1,\ldots,n$, the gradient ∇u is a vector given by

$$(\nabla u)_{i,j} = \left((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j}\right)$$

where

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

and

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

For pixels (i, j), where $i = 1, \ldots, m$ and $j = 1, \ldots, n$, the discrete version of the divergence operator is the following

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

A.3. Hymann-Shashkov's scheme

Scheme 3 comes from Hyman and Shashkov's paper [65]. In this section, the nodes are the image pixels. The image size is $m \times n$.

$$\text{GRAD} : HN \to HL$$

where HN is the space of discrete scalars given by their values in the nodes and HL is the space of discrete vector functions defined by their orthogonal projections onto directions of the edges of the cell. The stencil for operator $GRAD$ is shown in Figure A.3.

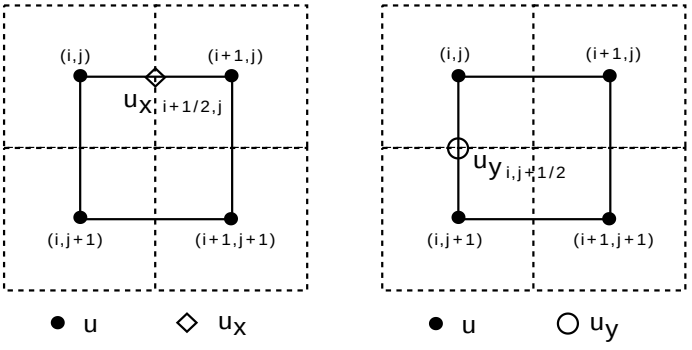


Figure A.3.: Stencil for operator GRAD.

GRAD $u = (u_x, u_y)$ has two components defined on the correponding edges. For $L_x(i+1/2, j)$ edges, where $i = 2, \ldots, m-1$ and $j = 2, \ldots, n$, u_x is given by

$$(u_x)_{i+1/2,j} = u_{i+1,j} - u_{i-1,j}.$$

For $L_y(i, j+1/2)$ edges, where $i = 2, \ldots, m$ and $j = 2, \ldots, n-1$, u_y is given by

$$(u_y)_{i,j+1/2} = u_{i,j+1} - u_{i,j-1}.$$

They constructed an operator $\overline{\text{DIV}}$ defined as the negative adjoint of GRAD. The stencil for operator $\overline{\text{DIV}}$ is shown in Figure A.4.

$$\overline{\text{DIV}} = -\text{GRAD}^*.$$

$$\overline{\text{DIV}} : HL \to HN.$$

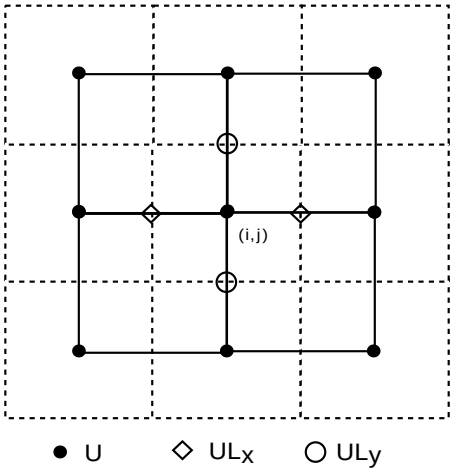


Figure A.4.: Stencil for operator $\overline{DIV}$.

For nodes (i, j), where $i = 2, \ldots, m-1$ and $j = 2, \ldots, n-1$, $\overline{\text{DIV}}$ can be expressed

$$\left(\overline{\text{DIV}}\,(u_x, u_y)\right)_{i,j} = (u_x)_{i+1/2,j} - (u_x)_{i-1/2,j} + (u_y)_{i,j+1/2} - (u_y)_{i,j-1/2}\,. \qquad \text{(A.1)}$$

A.4. Numerical results

Our experiments used 8-bits images with an intensity in the interval $[0, 255]$. For the tests and the comparative study, a set of gray-level images has been chosen, which can be seen in Figure A.5: Barbara, Boat, Hill, Man and Lena (512×512); House and Peppers (256×256); Girl (462×357). To quantify the results, PSNR and SSIM indicators

Figure A.5.: Set of gray-level images used for tests.

presented in Section 1.3.2 have been used. The results are obtained with Algorithm 1 to perform the isotropic restoration and Algorithm 2 to perform the anisotropic restoration. It is important to note that for the tests, the diffusion coefficient c in the step 6 of Algorithm 2 has been changed to focus only on the anisotropic diffusion. The diffusion coefficient in the anisotropic case is defined has followed:

$$c(x) = \begin{cases} P(x) \begin{pmatrix} \epsilon & 0 \\ 0 & c_0 \end{pmatrix} P^{-1}(x) & \text{if } x \text{ is in a valley and } \lambda_{min}(x) < \delta, \\ c_0 I_d & \text{otherwise.} \end{cases}$$

For the Hyman-Shashkov scheme, the information on the gradient is fragmented on two different grids. Only this information is used after considering a straight vertical edge at each location of the partial derivative in x and a straight horizontal edge at each location of the partial derivative in y. The diffusion is isotropic but with this scheme a better aspect is obtained.

The first experiment is the image restoration with isotropic diffusion. Two different noise levels have been chosen and the results are presented in Tables A.1, A.2, A.3 and A.4. The parameters c_0 and δ were chosen to optimize the results in term of PSNR in Tables A.1 and A.3 and in term of SSIM in Tables A.2 and A.4. It can be seen that the Hyman-Shaskov's scheme outperforms the two others schemes.

The second experiment is the image restoration with anisotropic diffusion. As in the previous experiment, two different noise levels have been chosen and the results are presented in Tables A.5, A.6, A.7 and A.8. The parameters c_0 and δ were chosen to optimize the results in term of PSNR in Tables A.5 and A.7 and in term of SSIM in Tables A.6 and A.8. Once again the isotropic diffusion is presented for the third scheme because this scheme is not compatible with anisotropic diffusion. It is clear that Scheme 1 outperforms the two others schemes.

$\sigma = 10$	Noisy		Scheme 1		Scheme 2		Scheme 3	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	28.15	0.701	30.40	0.852	30.50	0.856	**30.83**	0.851
Boat	28.14	0.692	32.19	0.852	32.13	0.855	**32.46**	0.861
Hill	28.12	0.694	32.35	0.854	32.28	0.854	**32.55**	0.861
Lena	28.14	0.614	33.46	0.870	33.44	0.870	**33.73**	0.879
Man	28.14	0.680	32.37	0.861	32.33	0.865	**32.55**	0.872
House	28.12	0.604	33.56	0.865	33.52	0.865	**33.97**	0.877
Peppers	28.12	0.669	33.20	0.856	33.18	0.901	**33.46**	0.913
Girl	28.52	0.588	33.37	0.856	33.37	0.861	**33.53**	0.858

Table A.1.: Comparative results of the best PSNR and the associated SSIM for the isotropic restoration using the three schemes with $\sigma = 10$.

$\sigma = 10$	Noisy		Scheme 1		Scheme 2		Scheme 3	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	28.15	0.701	30.12	0.856	30.19	0.861	30.14	**0.863**
Boat	28.14	0.692	32.02	0.858	31.91	0.857	32.37	**0.864**
Hill	28.12	0.694	32.17	0.856	32.23	0.855	32.48	**0.862**
Lena	28.14	0.614	33.11	0.878	33.11	0.878	33.40	**0.884**
Man	28.14	0.680	32.17	0.871	32.11	0.870	32.45	**0.877**
House	28.12	0.604	33.31	0.874	33.30	0.874	33.80	**0.880**
Peppers	28.12	0.669	32.71	0.916	32.72	0.916	33.03	**0.921**
Girl	28.52	0.588	32.93	0.872	32.87	0.869	33.12	**0.877**

Table A.2.: Comparative results of the best SSIM and the associated PSNR for the isotropic restoration using the three schemes with $\sigma = 10$.

$\sigma = 20$	Noisy		Scheme 1		Scheme 2		Scheme 3	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	22.14	0.460	26.41	0.674	26.31	0.686	**26.53**	0.700
Boat	22.19	0.428	28.65	0.745	28.59	0.749	**28.98**	0.770
Hill	22.14	0.411	29.11	0.747	29.06	0.744	**29.33**	0.760
Lena	22.10	0.343	30.06	0.781	30.03	0.786	**30.27**	0.803
Man	22.14	0.411	29.04	0.758	28.99	0.761	**29.20**	0.781
House	22.14	0.347	30.23	0.788	30.16	0.788	**30.75**	0.813
Peppers	22.74	0.339	29.34	0.822	29.30	0.827	**29.57**	0.838
Girl	22.20	0.423	29.46	0.733	29.46	0.734	**29.67**	0.755

Table A.3.: Comparative results of the best PSNR and the associated SSIM for the isotropic restoration using the three schemes with $\sigma = 20$.

$\sigma = 20$	Noisy		Scheme 1		Scheme 2		Scheme 3	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	22.14	0.460	25.62	0.712	25.78	0.714	25.87	**0.728**
Boat	22.19	0.428	28.46	0.759	28.41	0.760	28.94	**0.771**
Hill	22.14	0.411	29.05	0.750	28.98	0.749	29.33	**0.760**
Lena	22.10	0.343	29.49	0.807	29.44	0.807	29.91	**0.814**
Man	22.14	0.411	28.85	0.773	28.78	0.773	29.17	**0.782**
House	22.14	0.347	29.48	0.811	29.48	0.811	30.34	**0.822**
Peppers	22.74	0.339	28.64	**0.850**	28.68	**0.850**	29.54	0.848
Girl	22.20	0.423	28.78	0.777	28.80	**0.778**	29.65	0.763

Table A.4.: Comparative results of the best SSIM and the associated PSNR for the isotropic restoration using the three schemes with $\sigma = 20$.

$\sigma = 10$	Noisy		Scheme 1		Scheme 2		Scheme 3	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	28.15	0.701	30.96	0.838	**31.06**	0.855	30.83	0.851
Boat	28.14	0.692	**32.64**	0.863	32.50	0.861	32.46	0.861
Hill	28.12	0.694	**32.74**	0.863	32.63	0.861	32.55	0.861
Lena	28.14	0.614	**34.21**	0.884	33.98	0.877	33.73	0.879
Man	28.14	0.680	**32.76**	0.874	32.63	0.872	32.55	0.872
House	28.12	0.604	**34.42**	0.878	34.12	0.873	33.97	0.877
Peppers	28.12	0.669	**34.05**	0.914	33.73	0.909	33.46	0.913
Girl	28.52	0.588	**33.70**	0.859	33.50	0.851	33.53	0.858

Table A.5.: Comparative results of the best PSNR and the associated SSIM for the anisotropic restoration using the three schemes with $\sigma = 10$.

$\sigma = 10$	Noisy		Scheme 1		Scheme 2		Scheme 3	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	28.15	0.701	30.21	0.859	30.32	**0.864**	30.14	0.863
Boat	28.14	0.692	32.58	**0.866**	32.35	0.864	32.37	0.864
Hill	28.12	0.694	32.72	**0.865**	32.63	0.861	32.48	0.862
Lena	28.14	0.614	33.99	**0.887**	33.79	0.886	33.40	0.884
Man	28.14	0.680	32.66	**0.880**	32.49	0.877	32.45	0.877
House	28.12	0.604	34.28	**0.881**	33.88	0.879	33.80	0.880
Peppers	28.12	0.669	33.61	**0.925**	33.03	0.922	33.03	0.921
Girl	28.52	0.588	33.03	**0.874**	32.85	0.872	33.12	0.877

Table A.6.: Comparative results of the best SSIM and the associated PSNR for the anisotropic restoration using the three schemes with $\sigma = 10$.

$\sigma = 20$	Noisy		Scheme 1		Scheme 2		Scheme 3	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	22.14	0.460	26.69	0.700	**26.73**	0.707	26.53	0.700
Boat	22.19	0.428	**29.33**	0.771	29.21	0.770	28.98	0.770
Hill	22.14	0.411	**29.62**	0.767	29.52	0.761	29.33	0.760
Lena	22.10	0.343	**31.03**	0.816	30.77	0.814	30.27	0.803
Man	22.14	0.411	**29.59**	0.790	29.44	0.783	29.20	0.781
House	22.14	0.347	**31.61**	0.823	31.19	0.809	30.75	0.813
Peppers	22.74	0.339	**30.44**	0.851	30.13	0.843	29.57	0.838
Girl	22.20	0.423	**30.20**	0.770	29.98	0.751	29.67	0.755

Table A.7.: Comparative results of the best PSNR and the associated SSIM for the anisotropic restoration using the three schemes with $\sigma = 20$.

$\sigma = 20$	Noisy		Scheme 1		Scheme 2		Scheme 3	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	22.14	0.460	26.03	0.732	26.03	**0.733**	25.87	0.728
Boat	22.19	0.428	29.28	**0.780**	29.03	0.776	28.94	0.771
Hill	22.14	0.411	29.62	**0.767**	29.52	0.761	29.33	0.760
Lena	22.10	0.343	30.93	**0.823**	30.61	0.821	29.91	0.814
Man	22.14	0.411	29.55	**0.792**	28.78	0.773	29.17	0.782
House	22.14	0.347	31.54	**0.828**	31.05	0.824	30.34	0.822
Peppers	22.74	0.339	30.19	**0.869**	29.62	0.864	29.54	0.848
Girl	22.20	0.423	29.80	**0.788**	29.47	0.784	29.65	0.763

Table A.8.: Comparative results of the best SSIM and the associated PSNR for the anisotropic restoration using the three schemes with $\sigma = 20$.

A.5. Conclusion

In terms of quality, Schemes 1 and 2 give equivalent results with the isotropic diffusion and allow to perform anisotropic diffusion. In the case of anisotropic diffusion, the best results are obtained with Scheme 1. The Hyman-Shaskov's scheme is an isotropic scheme which is better than the two others when isotropic diffusion is considered. In the case of anisotropic diffusion, the Hyman-Shaskov's scheme is less efficient than Schemes 1 and 2.

For segmentation or texture removal, see Section 1.4.2, the more interesting scheme is the Hyman-Shaskov's one because it is more suited. The anisotropic diffusion could generate some artefacts around the edges.

B. Extension of Bai and Feng's restoration algorithm to color images

This appendix presents an extension to color images of Bai and Feng's restoration algorithm [24]. These authors used fractional derivatives to restore gray-level images. In Section 1.7, the implementation of the fractional derivatives provided by Bai and Feng is mixed with topological gradient information. In this appendix, there is no topology optimization.

The algorithm presented by Bai and Feng is an extension of Perona and Malik's scheme. It is iterative and use the fractional gradient of the image. A brief summary of the implementation of fractional derivatives is provided in Section 1.7.1. They solve an Euler-Lagrange equation (1.43) through the following gradient descent procedure:

$$\frac{\partial u}{\partial t} = -D_x^{\alpha *}(c(|D_x^\alpha u|^2)D_x^\alpha u) - D_x^{\alpha *}(c(|D_y^\alpha u|^2)D_y^\alpha u). \tag{B.1}$$

In image processing, the grey-level restoration algorithm of many methods is extend by taking into account the fact that the gradients of the different image channels are nearly the same. Table B.1 presents the reconstructions for different orders α of the red channel performed with Bai and Feng's algorithm by taking into account the fractional gradient of the original red channel and the fractional gradient of the original blue channel. Figure B.1 provides the noisy image and the two restorations with $\alpha = 1.5$. The results have nearly the same quality when another channel is used.

Channel	$\alpha = 1$	$\alpha = 1.25$	$\alpha = 1.5$	$\alpha = 1.75$	$\alpha = 2$
Red	$35.09/936$	$35.42/937$	$35.18/932$	$35.04/927$	$34.71/922$
Blue	$34.66/934$	$35.09/935$	$34.94/931$	$34.86/926$	$34.56/921$

Table B.1.: Reconstructions of the noisy Parrot red channel ($\sigma = 15$) with successively the fractional gradient from the original red channel, and with the one from the original blue channel.

Section B.1 presents my algorithm for color image which used a mean of the fractional gradients. Section B.2 presents some results for gray-level images to emphasize the interest of particular fractional orders. Section B.3 presents the color image results provided by the color algorithm.

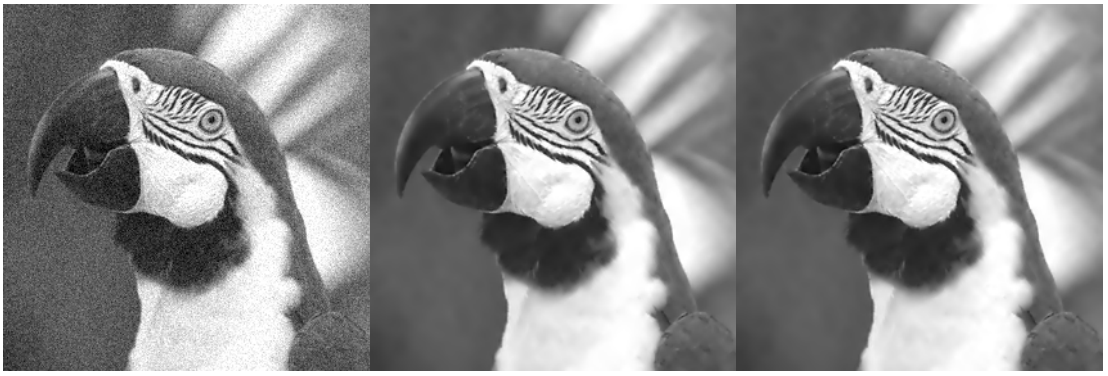


Figure B.1.: From left to right, the noisy red channel ($\sigma = 15$), reconstructions with the fractional gradient ($\alpha = 1.5$) from the original red channel (center) and from the original blue channel (right).

B.1. Algorithm for color image restoration

Algorithm 6 Algorithm for color image restoration

Input: $v = (v^1, v^2, v^3)$, Δt and k
Output: $u_n = (u_n^1, u_n^2, u_n^3)$

1: Let v be the input image and set $n = 1$, $u_n = v$, k, Δt, $t = k\Delta t$.
2: Compute $\hat{u}_n^l$ for $l = \{1, 2, 3\}$.
3: Compute the α-order partial differences $\tilde{D}_x^\alpha u_n^l$ and $\tilde{D}_y^\alpha u_n^l$ for $l = \{1, 2, 3\}$ using (1.50).
4: Compute in the spatial domain: $h_{xn}^l = c(\sum_{m=1}^3 |\tilde{D}^\alpha u_n^m|^2)\tilde{D}_x^\alpha u_n^l$ and $h_{yn}^l = c(\sum_{m=1}^3 |\tilde{D}^\alpha u_n^m|^2)\tilde{D}_y^\alpha u_n^l$ for $l = \{1, 2, 3\}$.
5: Compute for $m = x, y$, $K_m^* = \text{diag}(\text{conj}((1 - \exp(-i2\pi w_i/n))^\alpha \exp(i\pi\alpha w_i/n)))$.
6: Compute $\hat{g}_n^l = K_1^* \circ F(h_{xn}^l) + K_2^* \circ F(h_{yn}^l)$ for $l = \{1, 2, 3\}$.
7: Compute $\hat{u}_{n+1}^l = \hat{u}_n^l - \hat{g}_n^l \times \Delta t$ for $l = \{1, 2, 3\}$ and set $n = n + 1$
8: If $n = k$, compute the 2-D IDFT of $\hat{u}_n$, stop; else go to 3.

As in [24], the function $c(s) = 1/(1 + s)$ is used to obtain the diffusion coefficients. In this appendix, the step size Δt is always equal to $3 * 4^{-\alpha}$. Bai and Feng reflect the image symmetrically across the border in order to reduce discontinuities across the image border due to the periodization. The same principle is used in this appendix.

B.2. Numerical results for gray-level images

In this section, some numerical results obtained with the fractional-order diffusion algorithm in [24] are presented to emphasize the importance of fractional orders. Figure B.2 presents the set of gray-level images: Lena, Barbara, Hill and Boat (512×512); House and Peppers (256×256).

Figure B.2.: Set of gray-level images used for test.

Our experiments used 8-bits images with an intensity in the interval $[0, 255]$. Each image in Figure 2.10 is altered by an additive Gaussian noise of standard deviation $\sigma = 15$ and $\sigma = 25$ respectively. Table B.2 presents the results obtained with Bai and Feng's algorithm [24] on the noisy images. The PSNR and SSIM are given for each image. At the end of Table B.2, the mean of the number of iterations for each level of noise are given. One can note that when the order α is an integer, the results have a lower quality. The best results are obtained for a fractional-order α around 1.5 and 2.5. The number of iterations increases with the fractional-order. Figure B.3 (top) presents two noisy images the Boat and Hill images suffering respectively of a gaussian additive noise of standard deviation $\sigma = 15$ and $\sigma = 25$. Figure B.3 (middle) presents the reconstruction for a fractional-order $\alpha = 1.5$ and Figure B.3 (bottom) for $\alpha = 2.5$. The two reconstructions have a similar visual quality.

Images	σ	$\alpha = 1$	$\alpha = 1.25$	$\alpha = 1.5$	$\alpha = 1.75$	$\alpha = 2$
Lena	15	31.58/843	32.57/865	32.77/870	**32.81**/**870**	32.26/855
	25	29.19/791	30.26/822	30.49/**830**	**30.50**/**830**	29.74/797
Barbara	15	28.97/844	29.53/866	**29.54**/867	**29.54**/**868**	29.13/851
	25	25.99/756	26.55/784	**26.57**/**786**	26.52/785	26.00/749
Boat	15	29.82/792	30.53/811	**30.63**/**814**	30.61/813	30.07/795
	25	27.35/712	28.11/738	**28.22**/**743**	28.12/740	27.54/705
Hill	15	29.97/774	30.64/798	**30.65**/**800**	30.52/754	30.02/774
	25	27.72/683	28.44/714	**28.46**/**716**	28.39/712	27.75/677
House	15	32.057/844	**32.70**/853	32.60/**854**	32.52/852	31.91/837
	25	29.61/809	**30.32**/**822**	30.29/821	30.14/817	29.31/782
Peppers	15	30.81/873	31.95/901	32.11/906	**32.27**/**909**	31.52/892
	25	27.94/812	29.12/851	29.35/861	**29.46**/**864**	28.59/828
Iterations	15	272	387	536	751	1084
	25	723	1008	1407	1984	2964

Images	σ	$\alpha = 2$	$\alpha = 2.25$	$\alpha = 2.5$	$\alpha = 2.75$	$\alpha = 3$
Lena	15	32.26/855	32.93/871	**32.97**/**872**	32.89/869	32.61/864
	25	29.74/797	30.59/**830**	**30.60**/829	30.50/825	30.24/818
Barbara	15	29.13/851	29.68/870	29.70/**871**	**29.75**/869	29.48/862
	25	26.00/749	26.59/788	**26.68**/**789**	26.64/785	26.38/776
Boat	15	30.07/795	30.68/815	**30.70**/**818**	30.59/816	30.24/807
	25	27.54/705	28.25/743	**28.26**/**746**	28.15/743	27.87/734
Hill	15	30.02/774	30.58/798	**30.67**/**804**	30.60/803	30.32/794
	25	27.75/677	28.46/718	**28.51**/**721**	28.45/721	28.25/714
House	15	31.91/837	**32.49**/**850**	32.45/849	32.27/845	31.88/838
	25	29.31/782	**30.07**/**810**	30.03/808	29.88/801	29.56/791
Peppers	15	31.52/892	32.37/**909**	**32.39**/908	32.27/906	31.88/900
	25	28.59/828	**29.56**/**862**	29.51/860	29.38/855	29.01/846
Iterations	15	1084	1446	1979	2776	3904
	25	2964	3840	5299	7440	10535

Table B.2.: Image denoising with Bai and Feng's algorithm [24].

Figure B.3.: From top to bottom, Boat ($\sigma = 15$) and Hill ($\sigma = 25$) noisy images, restorations with $\alpha = 1.5$ and restorations with $\alpha = 2.5$.

B.3. Numerical results for color images

In this section, we compare the fractional-order diffusion algorithm [24] applied to each channel and our color algorithm, in order to emphasize the interest of a coupling between the channels. Figure B.4 presents the two color images: Parrot (286×286) and Flower (286×286).

Figure B.4.: Set of color images used for test.

The images of Figure B.4 are altered by an additive Gaussian noise of standard deviation $\sigma = 15$ and $\sigma = 25$. Table B.3 presents the results obtained with Bai and Feng's algorithm [24] on the noisy images by improving each channel separately and the results obtained with our algorithm. The PSNR and SSIM are given for each image. At the end of Table B.3, the means of the number of iterations for each noise level are given. One can note that when the order α is an integer, the results have a lower quality. The best results are obtained for a fractional-order α around 1.5 and 2.5. The number of iterations increases with the fractional-order. The coupled color algorithm clearly outperforms the absence of coupling.

Figures B.5 and B.6 present respectively on the top line the noisy Parrot images and the noisy Flower images. They are degraded with an additive Gaussian noise of standard deviation $\sigma = 15$ for the images on the left and $\sigma = 25$ for the images on the right. The second line of Figures B.5 and B.6 exhibit the reconstructions obtained by Bai and Feng's algorithm applied to each channel separately with the fractional-order $\alpha = 1.5$. The bottom line of Figures B.5 and B.6 shows the reconstructions obtained by Algorithm 6 with the fractional-order $\alpha = 1.5$. Quantitative estimators of the quality of these reconstructions are presented in Table B.3.

Images	coupled	$\alpha = 1$	$\alpha = 1.25$	$\alpha = 1.5$	$\alpha = 1.75$	$\alpha = 2$
Parrot	no	31.85/867	32.88/892	33.01/896	**33.05**/**897**	32.48/880
$\sigma = 15$	yes	33.03/892	33.82/907	33.90/909	**33.92**/**909**	33.60/904
Parrot	no	29.21/819	30.30/854	30.42/861	**30.47**/**861**	29.73/832
$\sigma = 25$	yes	30.56/853	31.42/875	**31.45**/**878**	31.44/877	31.11/869
Flower	no	29.21/857	30.18/881	30.39/885	**30.48**/ **886**	29.92/870
$\sigma = 15$	yes	30.82/895	31.65/909	31.86/913	**31.90**/**912**	31.49/906
Flower	no	26.47/771	27.37/804	27.55/810	**27.61**/**811**	26.96/778
$\sigma = 25$	yes	28.16/834	28.94/854	29.09/857	**29.11**/**858**	28.68/845
Iterations	no	272	396	544	755	1100
$\sigma = 15$	yes	228	365	458	639	878
Iterations	no	724	1004	1410	1975	3030
$\sigma = 25$	yes	578	836	1172	1644	2402

Images	coupled	$\alpha = 2$	$\alpha = 2.25$	$\alpha = 2.5$	$\alpha = 2.75$	$\alpha = 3$
Parrot	no	32.48/880	**33.07**/**895**	33.04/894	32.94/890	32.60/883
$\sigma = 15$	yes	33.60/904	**33.84**/**906**	33.72/903	33.59/900	33.25/892
Parrot	no	29.73/832	**30.42**/**853**	30.42/852	30.28/846	29.96/836
$\sigma = 25$	yes	31.11/869	**31.32**/**871**	31.18/865	30.92/855	30.63/847
Flower	no	29.92/870	29.94/874	**30.04**/**877**	30.00/875	29.63/866
$\sigma = 15$	yes	31.49/906	32.03/**914**	**32.04**/913	30.98/897	30.69/891
Flower	no	26.96/778	27.90/819	28.06/**824**	**28.07**/823	27.77/814
$\sigma = 25$	yes	28.68/845	**29.18**/**857**	29.15/855	28.67/843	28.74/844
Iterations	no	1100	1442	1988	2768	3847
$\sigma = 15$	yes	878	1222	1685	2328	3190
Iterations	no	3030	3794	5250	7394	10442
$\sigma = 25$	yes	2402	3242	4541	6200	8641

Table B.3.: Image denoising with Bai and Feng's algorithm [24] (no coupling) and Algorithm 6 (coupling).

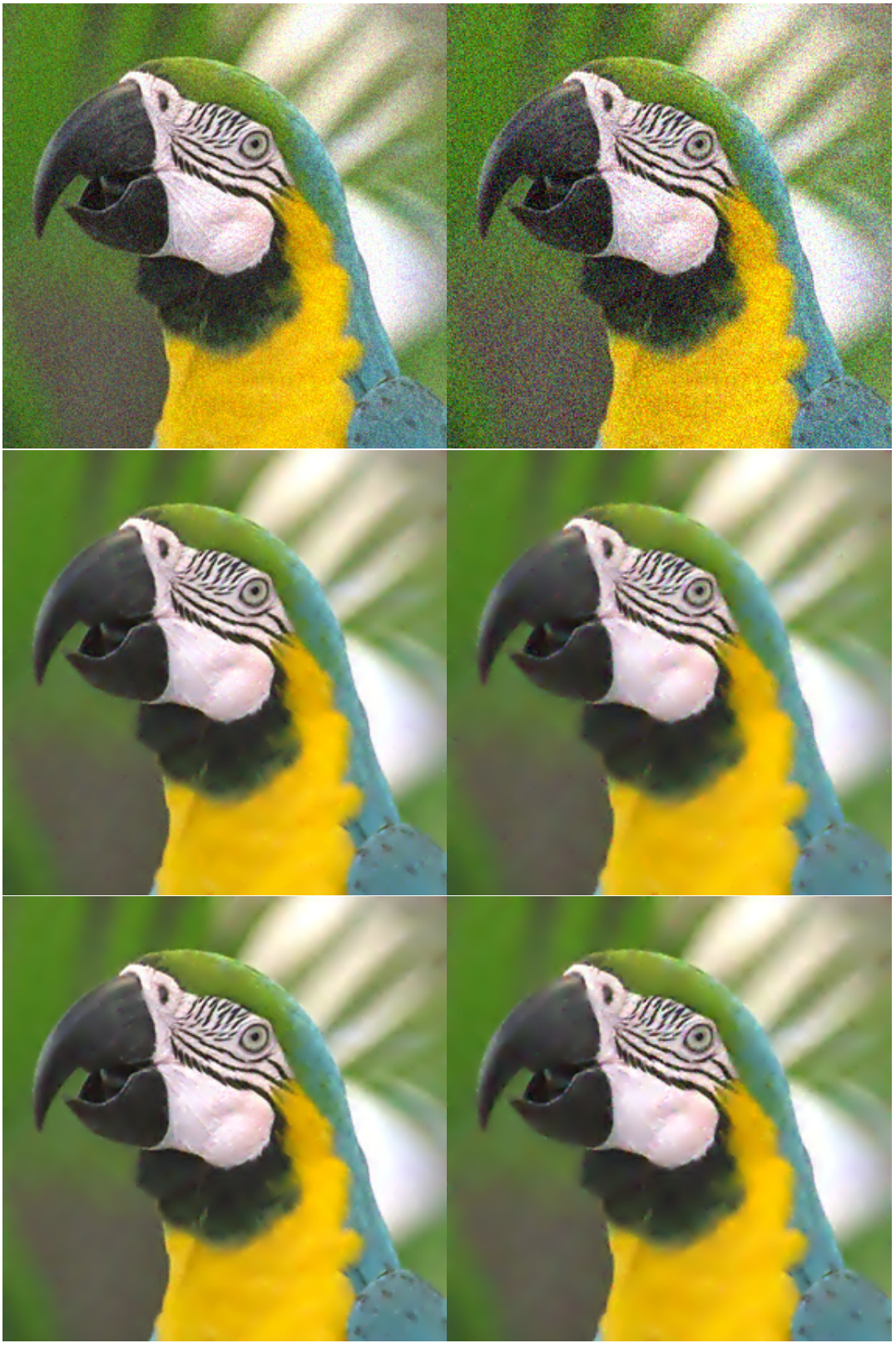


Figure B.5.: From top to bottom, Parrot ($\sigma = 15$) and Parrot ($\sigma = 25$) noisy images, channel by channel restorations with $\alpha = 1.5$ and coupled channel restorations with $\alpha = 1.5$.

Figure B.6.: From top to bottom, Flower ($\sigma = 15$) and Flower ($\sigma = 25$) noisy images, channel by channel restorations with $\alpha = 1.5$ and coupled channel restorations with $\alpha = 1.5$.

B.4. Conclusion

The numerical tests reported in this appendix demonstrate that the image fractional-order gradients are nearly the same for each channel. Better color image reconstruction can be obtained by coupling the image fractional-order gradients as it is done in the case of image integer-order gradients.

This allows to propose an extension of Bai and Feng's algorithm [24] for denoising color images. The proposed algorithm performs faster and better restorations as indicated by the substantial gains in term of PSNR and SSIM.

The numerical results with grey-level images and color images illustrate the importance of fractional-order compared to integer-order. Two zones of interests appear for the best fractional-order: between 1.5 and 1.75 and between 2.25 and 2.5.

In our experiment, the combination of the three channels is a mean but it could be a weigthed mean the weight depending on the noise level in each channel.

C. Segmentation of a human vertebrae in 3D

This appendix presents a project initiated with the Hospital of Purpan. Dr. Nicolas Sans uses SpineJack implants, sort of car jacks, to perform anatomical reduction and stabilization of a human vertebrae to correct a compression of the vertebrae. The SpineJack implants are used on patients and an impact study has already been performed. Doctor Sans contacted us because he would like to have independent information about the efficiency of these implants.

The aim is to segment the 3D images of a human vertebrae before and after an operation with topological gradient information, see the Chapter 1 and perform registration [45] in order to detect the difference between the two. This appendix presents only the segmentation protocol and the numerical results in respectively Sections C.1 and C.2.

Figure C.1.: Preoperation data.

C.1. Protocol

Figures C.1 and C.2 present slides of the vertebrae in respectively the preoperation dataset and the postoperation one.

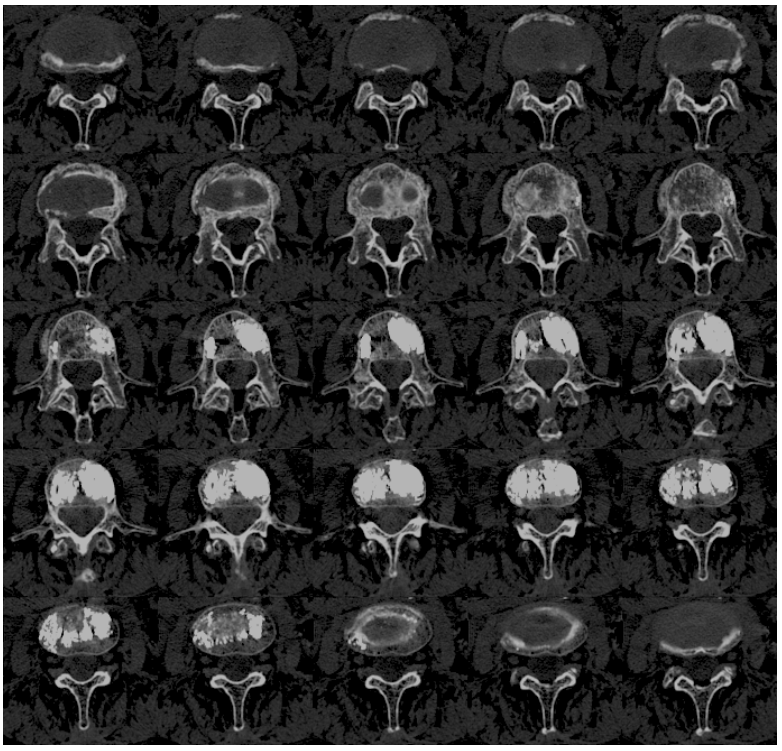


Figure C.2.: Postoperation data.

The datasets are coded with DICOM and they are in an image format. The acquisition lengths in x and y are of 0.75 mm and the image resolution is around 0.25 mm so an interpolation has been performed. The acquisition length in z is of 0.3 mm, it is more precise because the main interest resides in the evolution of the plateaus.

1. The vertebrae of interest is isolated from the rest of the data.

2. The images are reduced in the x and y axis in order to have back data similar to the ones taken by the instrument. The reduction is made in the Fourier space to keep the main features.

3. The segmentation in three dimensions is performed with the Hymann-Shaskov scheme, see Section A.3.

4. A threshold is used to detect the bones.

5. The vertebrae is filled to perform the registration.

The vertebrae has to be filled because the cement used in the operation to fix the SpineJack implants and the vertebrae is also detected and can perturb the registration. Although the bone is difficult to detect on some vertebrae sides, both plateaus are well detected.

C.2. Numerical results

Figures C.3 and C.4 present the results obtained with different orientations of the vertebrae in respectively the preoperation dataset and the postoperation one.

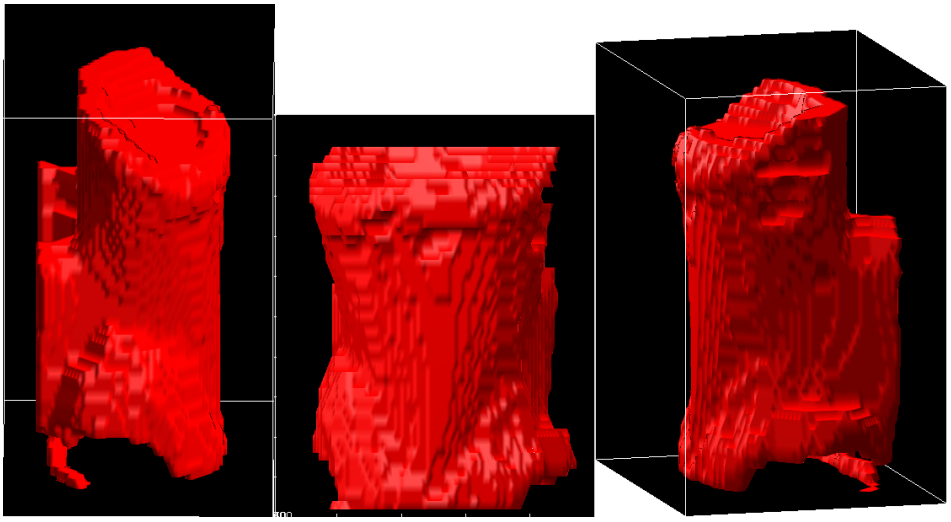


Figure C.3.: Segmentation result of the vertebrae in the preoperation dataset.

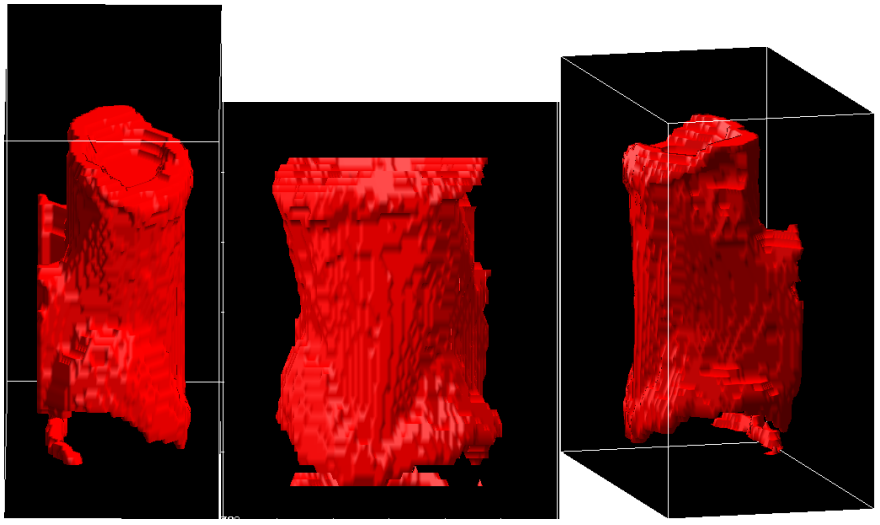


Figure C.4.: Segmentation result of the vertebrae in the postoperation dataset.

C.3. Conclusion and further improvements

The segmentation in 3D with topological gradient is operative. Further improvements could comprise

- using the data before the software convert them to an image format,
- using an automatic detection with a shape model of the positions in the dataset of the vertebraes,
- combining the segmentation procedure with level set methods.

D. Application of topological gradient to 1-D signals

This appendix presents a project initiated with the biologists Laurence Salomé and Catherine Tardin and the physicists Nicolas Destainville and Manoel Manghi of Université Paul Sabatier. They work on Tethered Particle Motion (TPM). The TPM is a promising technique used in biology and biophysics laboratories. This technique provides information relative to conformational changes of DNA molecules, e.g. upon looping or interaction with proteins, by tracking the Brownian motion of a particle probe tethered to a surface by a single DNA molecule and by detecting changes in its motion amplitude, see Figure D.1.

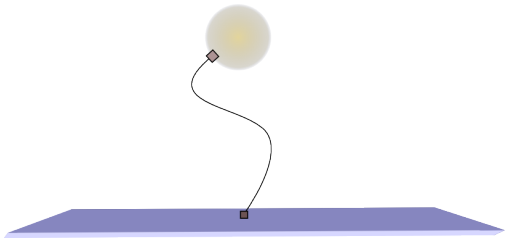


Figure D.1.: Scheme of a TPM experiment.

The diameter of the particle ranges from a few tens to several hundreds of nanometers. By measuring DNA end-to-end distance (or effective length), tracking by video microscopy the particle trajectory informs about DNA conformations in real time.

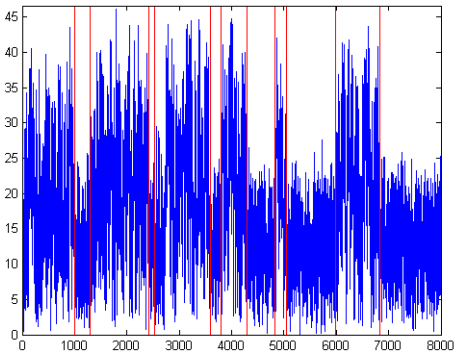


Figure D.2.: Example of signal in TPM. The abscissa is the time and the ordinate is the motion amplitude. The signal is in blue and the transitions are in red.

Figure D.2 presents an example of signal in TPM. The biologists and physicists are interested by the detection of the transitions. They need to know the precision of the detection method. The spatial resolution refers to the smallest number of samples needed to detect a transition. It is interesting to know the size of the smallest rectangular signal that can be detected.

Signal segmentation plays a significant role in recognition-oriented signal processing. The aim is to detect the abrupt changes in the signal and segment the signal accordingly. The topological derivatives have been succesfully used in image segmentation. A study of their applications to one-dimensional signals is performed in this appendix.

There is an interaction between the noise reduction and the spatial resolution. The level of noise influences the coefficients of the segmentation methods. The purpose is to compare the noise reduction and the spatial resolution of different methods. The smaller the spatial resolution, the better the method. For the theoretical study of this appendix, two types of signals are considered: a step signal and a rectangular signal, see Figure D.3. In order to simplify the analysis, an additive Gaussian noise is considered. Let s be an original signal, v the observed signal and n an additive Gaussian noise. The quantities are related by

$$v = s + n.$$

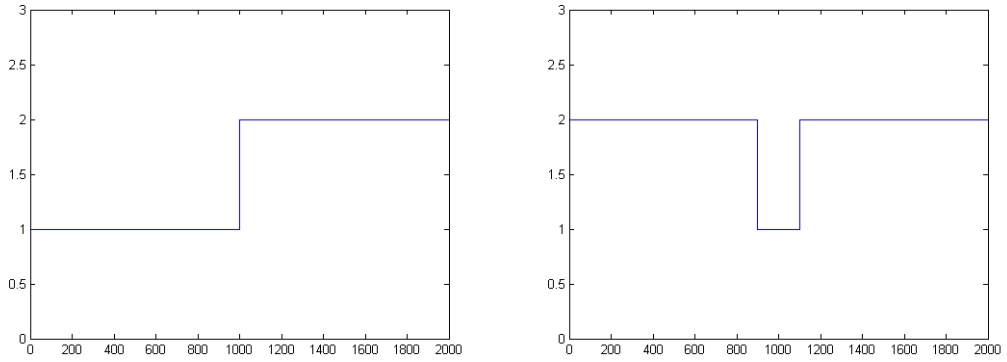


Figure D.3.: From left to right, the step signal and the rectangular signal.

Two different filtering processes are studied: a sliding window and a low pass filtering with the following diffusion equation:

$$-c^2u + u = v.$$

In Section D.1, a filtering with a sliding window is studied. In Section D.2, the application of a diffusion equation is studied. Section D.3 demonstrates that a better spatial resolution is obtained with the diffusion equation.

Section D.4 is dedicated to a topological asymptotic analysis. An example shows the interest of the topological gradient to detect abrupt changes. Section D.5 presents a concrete application to TPM.

D.1. Sliding window

D.1.1. Effect on the signal ?

Let L be the length of the sliding windows w. The convolution of the signal with the sliding window follows the next equation:

$$s * w(t) = \frac{1}{L} \sum_{-L/2}^{L/2} s(t+i) \tag{D.1}$$

Figure D.4 presents the convolution of the step signal with the sliding window and the convolution of the rectangular signal with the sliding window. With the rectangular signal, the form of the resulting signal depends on the length l_c of the rectangular signal, i. e. the number of samples between the two transitions.

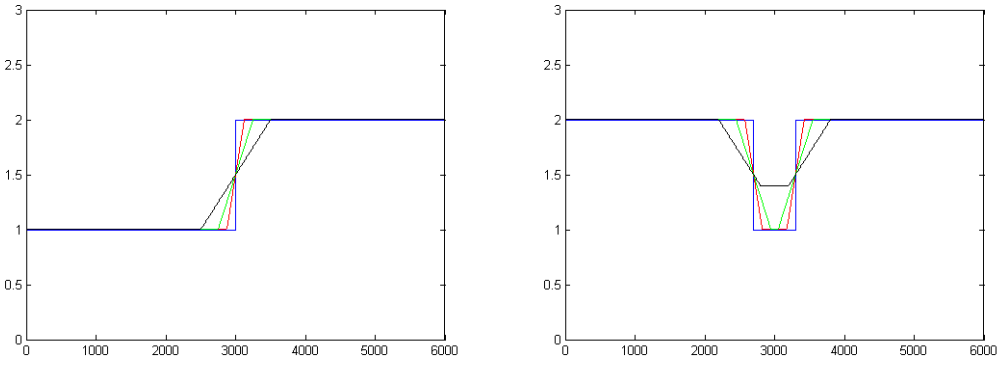


Figure D.4.: From left to right, the step signal and the rectangular signal after different convolutions with a sliding window.

D.1.2. Effect on the noise ?

If we assume that the noise n is an additive Gaussian noise of mean 0 and of variance σ^2 then $L * n$ is a Gaussian noise of mean 0 and of variance $\frac{\sigma^2}{L}$.

D.2. Diffusion equation

D.2.1. Effect on the signal ?

The diffusion equation

$$-c^2\varphi'' + \varphi = \delta \tag{D.2}$$

has the fundamental solution

$$\varphi(t) = K \exp\left(\frac{-|t|}{c}\right), \tag{D.3}$$

where K the normalization coefficient is defined as follow

$$\sum_{-\infty}^{+\infty} \varphi(t) = 1 \Leftrightarrow \frac{1}{K} = 2\sum_{t=0}^{+\infty} \exp\left(\frac{-t}{c}\right) - 1 \Leftrightarrow K = \frac{1-\exp\left(\frac{-1}{c}\right)}{1+\exp\left(\frac{-1}{c}\right)} = \tanh\left(\frac{1}{2c}\right). \tag{D.4}$$

Figure D.5 provides a representation of φ. The final signals are obtained by the convolution of the signals with the function φ, see Figure D.6 for the results without noise.

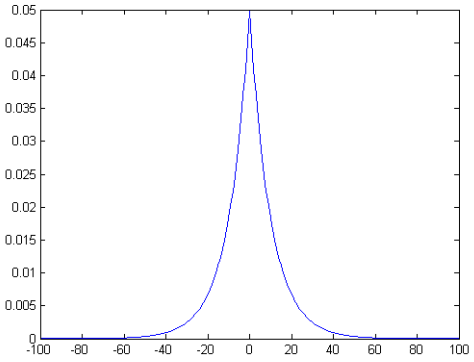


Figure D.5.: The function φ.

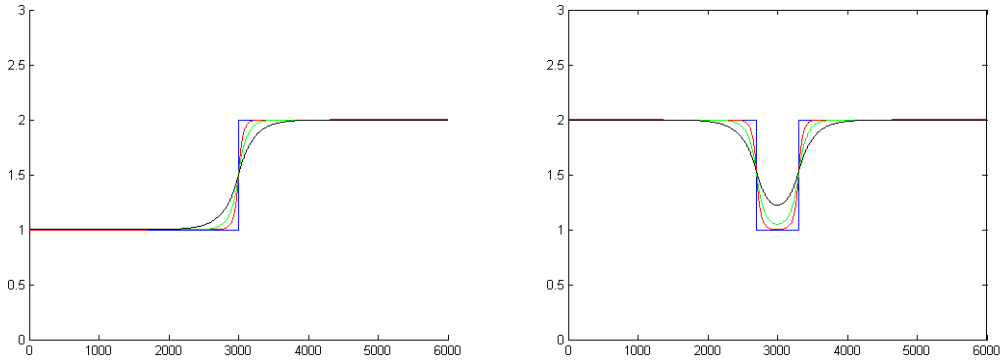


Figure D.6.: From left to right, the step signal and the rectangular signal after different convolutions with the function φ.

D.2.2. Effect on the noise ?

One can note that for a set of N samples following a Gaussian law $\mathcal{N}(0, \sigma^2)$, we have:

$$\frac{1}{N}\sum_{m=1}^{N}\left(\sum_{j=-\infty}^{+\infty}\varphi(j)n_{m-j}\right)^2 = \frac{1}{N}\sum_{m=1}^{N}\sum_{j=-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty}\varphi(j)\varphi(k)n_{m-j}n_{m-k}$$
$$= \sigma^2\sum_{j=-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty}\varphi(j)\varphi(k)\delta_{j,k}$$

Thanks to the previous relation, the variance of $n * \varphi$ is

$$\begin{aligned}\mathrm{Var}(n*\varphi) =& \sigma^2K^2\sum_{t=-\infty}^{+\infty}\exp\left(\frac{-2|t|}{c}\right) = \sigma^2K^2\left(2\sum_{t=0}^{+\infty}\exp\left(\frac{-2t}{c}\right)-1\right)\\ =& \sigma^2K^2\frac{1+\exp\left(-\frac{2}{c}\right)}{1-\exp\left(\frac{-2}{c}\right)} = \sigma^2\tanh^2\left(\frac{1}{2c}\right)\coth\left(\frac{1}{c}\right)\end{aligned}$$

D.2.3. Derivative of the direct solution with the step signal

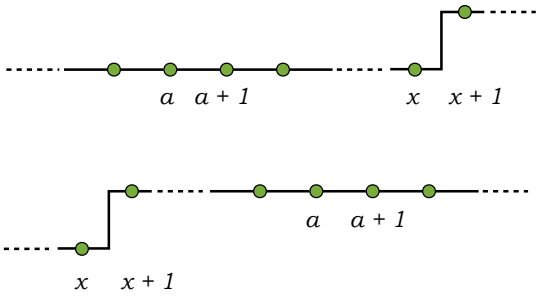


Figure D.7.: Two configurations for the derivative of the direct solution with the step signal.

The discrete derivative of the direct solution is the difference of $s * \varphi$ at position $a + 1$ and at position a. The result depends on the transition between positions x and $x + 1$. Let r_1 be the lower level of the signal and r_2 be the upper one.

If the derivative is calculated before the jump, we have the following equation:

$$s*\varphi(a+1) - s*\varphi(a) = S_1 + S_2 + S_3 = (r_2 - r_1)K\exp\left(\frac{a-x}{c}\right), \qquad \text{(D.5)}$$

where

$$
\begin{aligned}
S_1 &= \sum_{m=-\infty}^{a} s(m)\left(\varphi(a+1-m)-\varphi(a-m)\right), \\
\Leftrightarrow S_1 &= r_1 K\left(\exp\left(-\frac{a+1}{c}\right)-\exp\left(-\frac{a}{c}\right)\right)\sum_{m=-\infty}^{a}\exp\left(\frac{m}{c}\right), \\
\Leftrightarrow S_1 &= r_1 K\left(1-\exp\left(\frac{1}{c}\right)\right)\exp\left(-\frac{a+1}{c}\right)\left(\frac{-\exp\left(\frac{a+1}{c}\right)}{1-\exp\left(\frac{1}{c}\right)}\right), \\
\Leftrightarrow S_1 &= -r_1 K,
\end{aligned}
$$

and

$$
\begin{aligned}
S_2 &= \sum_{m=a+1}^{x} s(m)\left(\varphi(a+1-m)-\varphi(a-m)\right), \\
\Leftrightarrow S_2 &= r_1 K\left(\exp\left(\frac{a+1}{c}\right)-\exp\left(\frac{a}{c}\right)\right)\sum_{m=a+1}^{x}\exp\left(-\frac{m}{c}\right), \\
\Leftrightarrow S_2 &= r_1 K\left(1-\exp\left(-\frac{1}{c}\right)\right)\exp\left(\frac{a+1}{c}\right)\left(\frac{\exp\left(-\frac{a+1}{c}\right)-\exp\left(-\frac{x+1}{c}\right)}{1-\exp\left(\frac{1}{c}\right)}\right), \\
\Leftrightarrow S_2 &= r_1 K - r_1 K\exp\left(\frac{a-x}{c}\right),
\end{aligned}
$$

and

$$
\begin{aligned}
S_3 &= \sum_{m=x+1}^{\infty} s(m)\left(\varphi(a+1-m)-\varphi(a-m)\right), \\
\Leftrightarrow S_3 &= r_2 K\left(\exp\left(\frac{a+1}{c}\right)-\exp\left(\frac{a}{c}\right)\right)\sum_{m=x+1}^{\infty}\exp\left(-\frac{m}{c}\right), \\
\Leftrightarrow S_3 &= r_2 K\left(1-\exp\left(-\frac{1}{c}\right)\right)\exp\left(\frac{a+1}{c}\right)\left(\frac{\exp\left(-\frac{x+1}{c}\right)}{1-\exp\left(-\frac{1}{c}\right)}\right), \\
\Leftrightarrow S_3 &= r_2 K\exp\left(\frac{a-x}{c}\right).
\end{aligned}
$$

If the derivative is calculated after the jump, we have the following equation:

$$
s*\varphi(a+1)-s*\varphi(a)=(r_2-r_1)K\exp\left(\frac{x-a}{c}\right). \tag{D.6}
$$

D.2.4. Derivative of the direct solution with the rectangular signal

The rectangular signal can be seen as the sum of two step signals.

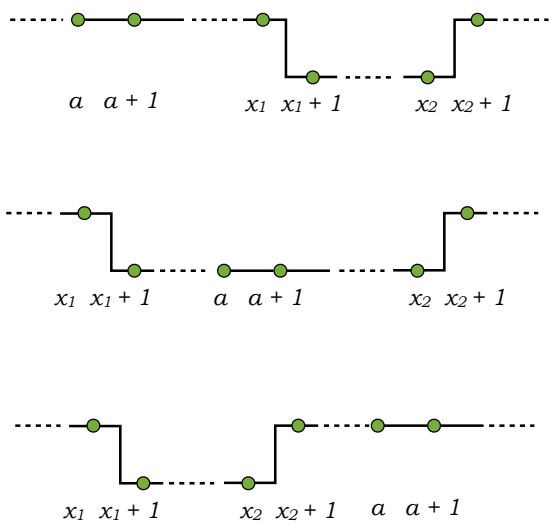


Figure D.8.: Three configurations for the derivative of the direct solution with the rectangular signal.

The discrete derivative of the direct solution is the difference of $s * \varphi$ at position $a + 1$ and at position a. The result depends on the first transition between positions x_1 and $x_1 + 1$ and the second transition between positions x_1 and $x_1 + 1$. Let r_1 be the lower level of the signal and r_2 be the upper one.

If the derivative is calculated before the first jump, we have the following equation:

$$s * \varphi(a+1) - s * \varphi(a) = K\left((r_1 - r_2)\exp\left(\frac{a - x_1}{c}\right) + (r_2 - r_1)\exp\left(\frac{a - x_2}{c}\right)\right). \tag{D.7}$$

If the derivative is calculated between the jumps, we have the following equation:

$$s * \varphi(a+1) - s * \varphi(a) = K\left((r_1 - r_2)\exp\left(\frac{x_1 - a}{c}\right) + (r_2 - r_1)\exp\left(\frac{a - x_2}{c}\right)\right). \tag{D.8}$$

If the derivative is calculated after the second jump, we have the following equation:

$$s * \varphi(a+1) - s * \varphi(a) = K\left((r_1 - r_2)\exp\left(\frac{x_1 - a}{c}\right) + (r_2 - r_1)\exp\left(\frac{x_2 - a}{c}\right)\right). \tag{D.9}$$

D.3. Comparison of the spatial resolutions

In this section, the standard deviation of the aditive Gaussian noise is equal to 0.5.

If we consider the width of the function at half its height as the resolution:

- with the sliding window, the spatial resolution is its length L,
- with the function φ, the spatial resolution is $2c\ln(2)$.

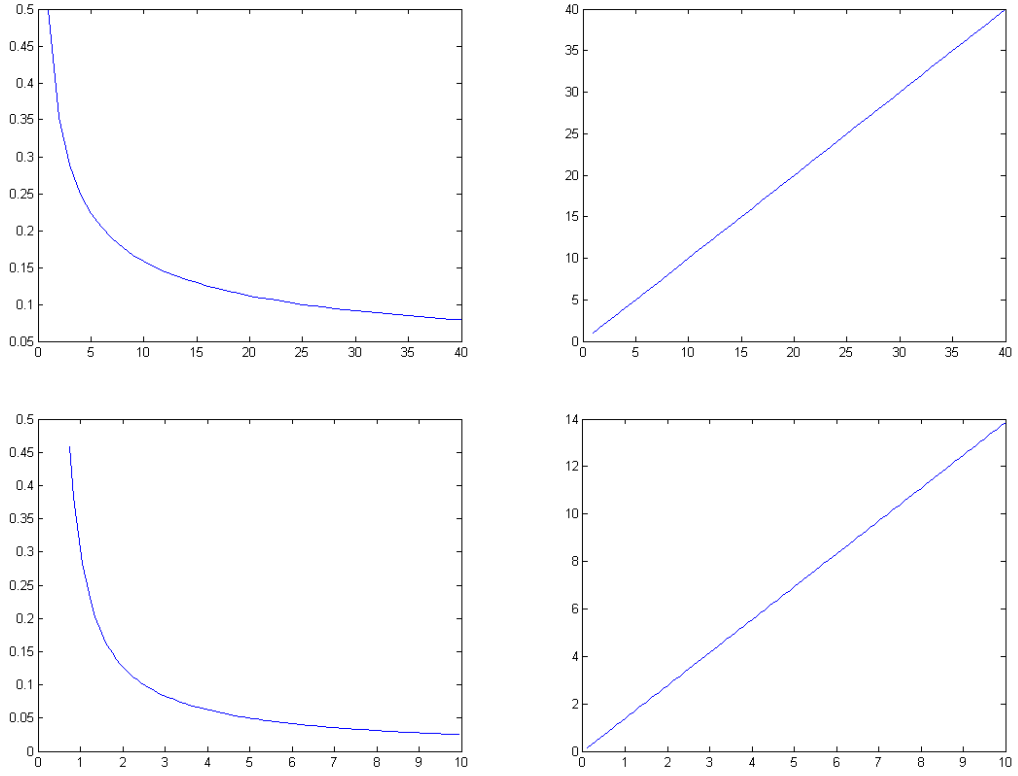


Figure D.9.: First line, from left to right, the evolution of the standard deviation and of the resolution with the length L of the sliding window. Second line, from left to right, the evolution of the standard deviation and of the resolution with the diffusion coefficient c of the diffusion equation.

The first line of Figure D.9 presents the evolution of the standard deviation and the resolution with the length L of the sliding window. The second line of Figure D.9 presents the evolution of the standard deviation and the resolution with the diffusion coefficient c of the diffusion equation. Figure D.10 compares the evolution of the standard deviation and the resolution with the length L of the sliding window. For the same decrease in standard deviation, the diffusion equation has a lower, and therefore better, resolution than the sliding window.

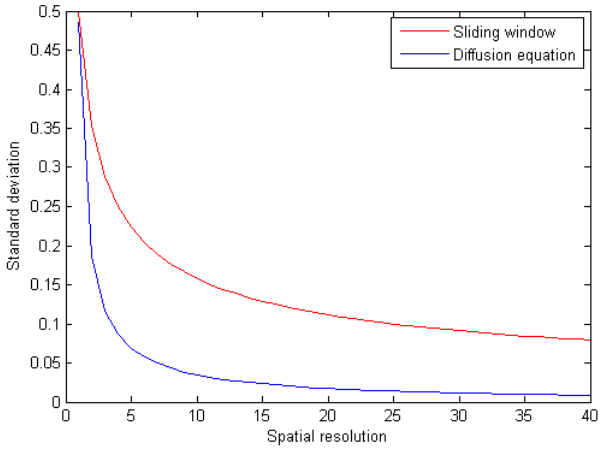


Figure D.10.: Comparison of the resolutions, the red and blue curves represent respectively the sliding window and the diffusion equation. The abscissa is the spatial resolution and the ordinate is the standard deviation.

D.4. Topological analysis

D.4.1. Cost function without noise

By using Equations D.5 and D.6, the cost function for the step signal without perturbation and in the absence of noise is the following:

$$
\begin{aligned}
& j(s*\varphi) = \sum_{i=-\infty}^{\infty} \left| s*\varphi(i+1) - s*\varphi(i) \right|^2, \\
\Leftrightarrow\ & j(s*\varphi) = (r_2-r_1)^2 K^2 \left(\sum_{i=-\infty}^{x-1} \exp\left(\frac{2(i-x)}{c}\right) + \sum_{i=x}^{\infty} \exp\left(\frac{2(x-i)}{c}\right) \right), \\
\Leftrightarrow\ & j(s*\varphi) = (r_2-r_1)^2 K^2 \left(\frac{-1}{1-\exp\left(\frac{2}{c}\right)} + \frac{1}{1-\exp\left(-\frac{2}{c}\right)} \right), \\
\Leftrightarrow\ & j(s*\varphi) = (r_2-r_1)^2 K^2 \frac{1}{\tanh\left(\frac{1}{c}\right)}.
\end{aligned}
\tag{D.10}
$$

By using Equations D.7, D.8 and D.9, the cost function for the crenel signal without perturbation and in the absence of noise is the following:

$$
\begin{aligned}
& j(s_m*\varphi) = \sum_{i=-\infty}^{\infty} \left| s_m*\varphi(i+1) - s_m*\varphi(i) \right|^2 = S_1 + S_2 + S_3, \\
\Leftrightarrow\ & j(s_m*\varphi) = 2(r_2-r_1)^2 K^2 \left(\frac{1-\exp\left(\frac{x_1-x_2}{c}\right)}{\tanh\left(\frac{1}{c}\right)} + (x_1-x_2)\exp\left(\frac{x_1-x_2}{c}\right) \right),
\end{aligned}
\tag{D.11}
$$

where

$$S_1 = \sum_{i=-\infty}^{x_1-1} \left| K\left((r_1-r_2)\exp\left(\frac{i-x_1}{c}\right)+(r_2-r_1)\exp\left(\frac{i-x_2}{c}\right)\right)\right|^2,$$

$$\Leftrightarrow S_1 = K^2(r_1-r_2)^2 \sum_{i=-\infty}^{x_1-1} \left(\exp\left(\frac{2(i-x_1)}{c}\right) + \exp\left(\frac{2(i-x_2)}{c}\right) - 2\exp\left(\frac{2i-(x_1+x_2)}{c}\right)\right),$$

$$\Leftrightarrow S_1 = K^2(r_1-r_2)^2 \left(-\frac{1}{1-\exp\left(\frac{2}{c}\right)} - \frac{\exp\left(\frac{2(x_1-x_2)}{c}\right)}{1-\exp\left(\frac{2}{c}\right)} + \frac{2\exp\left(\frac{(x_1-x_2)}{c}\right)}{1-\exp\left(\frac{2}{c}\right)}\right),$$

and

$$S_2 = \sum_{i=x_1}^{x_2-1} \left| K\left((r_1-r_2)\exp\left(\frac{x_1-i}{c}\right)+(r_2-r_1)\exp\left(\frac{i-x_2}{c}\right)\right)\right|^2$$

$$\Leftrightarrow S_2 = K^2(r_1-r_2)^2 \sum_{i=x_1}^{x_2-1} \left(\exp\left(\frac{2(x_1-i)}{c}\right) + \exp\left(\frac{2(i-x_2)}{c}\right) - 2\exp\left(\frac{x_1-x_2}{c}\right)\right),$$

$$\Leftrightarrow S_2 = K^2(r_1-r_2)^2 \left(\frac{1-\exp\left(\frac{2(x_1-x_2)}{c}\right)}{1-\exp\left(\frac{-2}{c}\right)} + \frac{\exp\left(\frac{2(x_1-x_2)}{c}\right)-1}{1-\exp\left(\frac{2}{c}\right)} + 2(x_1-x_2)\exp\left(\frac{x_1-x_2}{c}\right)\right),$$

$$\Leftrightarrow S_2 = K^2(r_1-r_2)^2 \left(\frac{1-\exp\left(\frac{2(x_1-x_2)}{c}\right)}{\tanh\left(\frac{1}{c}\right)} + 2(x_1-x_2)\exp\left(\frac{x_1-x_2}{c}\right)\right),$$

and

$$S_3 = \sum_{i=x_2}^{\infty} \left| K\left((r_1-r_2)\exp\left(\frac{x_1-i}{c}\right)+(r_2-r_1)\exp\left(\frac{x_2-i}{c}\right)\right)\right|^2,$$

$$\Leftrightarrow S_3 = K^2(r_1-r_2)^2 \sum_{i=x_2}^{\infty} \left(\exp\left(\frac{2(x_1-i)}{c}\right) + \exp\left(\frac{2(x_2-i)}{c}\right) - 2\exp\left(\frac{(x_1+x_2)-2i}{c}\right)\right),$$

$$\Leftrightarrow S_3 = K^2(r_1-r_2)^2 \left(\frac{\exp\left(\frac{2(x_1-x_2)}{c}\right)}{1-\exp\left(-\frac{2}{c}\right)} + \frac{1}{1-\exp\left(-\frac{2}{c}\right)} - \frac{2\exp\left(\frac{(x_1-x_2)}{c}\right)}{1-\exp\left(-\frac{2}{c}\right)}\right).$$

If the step signal is split into two parts and the diffusion equation is applied on each part, it is equivalent to apply the diffusion equation on two signals corresponding to each part and with a mirror effect. If the signal is split at the jump position then $j_p(s * \varphi_p)$ is equals to 0. Otherwise, if the signal is split at the position a, $j_p(s * \varphi_p)$ is equals to the sum of $\frac{1}{2}j(s_m * \varphi)$ where s_m is a rectangular signal of length $l_c = 2|x - a|$.

$$j_p(s * \varphi_p) = (r_2 - r_1)^2 K^2 \left(\frac{1 - \exp\left(-2\frac{|x-a|}{c}\right)}{\tanh\left(\frac{1}{c}\right)} - 2|x - a| \exp\left(-2\frac{|x-a|}{c}\right) \right). \tag{D.12}$$

The difference of $j_p(s*\varphi_p)$ and $j(s*\varphi)$, given by Equation (D.10) and Equation (D.12) respectively, is

$$j_p(s * \varphi_p) - j(s * \varphi) = -(r_2 - r_1)^2 K^2 \left(\frac{1}{\tanh\left(\frac{1}{c}\right)} + 2|x - a| \right) \exp\left(-2\frac{|x-a|}{c}\right). \tag{D.13}$$

D.4.2. The adjoint equation without noise

The adjoint solution verifies the following equation:

$$-c^2 p_0'' + p_0 = u_0''. \tag{D.14}$$

where u_0 is equal to $s*\varphi$. The adjoint solution is obtained by the convolution of $(s*\varphi)''$ with the function φ.

In the case of the step signal before the jump, using Equation (D.5), we have the following equation:

$$u_0''(a) = (s * \varphi)'(a + 1) - (s * \varphi)'(a) = (r_2 - r_1) K \exp\left(\frac{a - x}{c}\right) \left(1 - \exp\left(-\frac{1}{c}\right)\right). \tag{D.15}$$

In the case of the step signal after the jump, using Equation (D.6), we have the following equation:

$$u_0''(a) = (s * \varphi)'(a + 1) - (s * \varphi)'(a) = (r_2 - r_1) K \exp\left(\frac{x - a}{c}\right) \left(1 - \exp\left(\frac{1}{c}\right)\right). \tag{D.16}$$

D.4.3. Derivative of the adjoint solution with the step signal

If the derivative is taken before the jump, we have the following equation:

$$\begin{aligned} p_0'(a) &= u_0'' * \varphi(a+1) - u_0'' * \varphi(a) = S_1 + S_2 + S_3 \\ &= (r_2 - r_1)K^2 \exp\left(\frac{a-x}{c}\right)\left(2K\cosh\left(\frac{2}{c}\right) + (x-a)\left(\exp\left(\frac{1}{c}\right) - 1\right)^2\right), \end{aligned} \tag{D.17}$$

where

$$S_1 = \sum_{m=-\infty}^{a} u_0''(m)\left(\varphi(a+1-m) - \varphi(a-m)\right),$$

$$\Leftrightarrow S_1 = (r_2 - r_1)K^2\left(2 - \exp\left(\frac{1}{c}\right) - \exp\left(-\frac{1}{c}\right)\right)\exp\left(-\frac{x+a}{c}\right)\sum_{m=-\infty}^{a}\exp\left(\frac{2m}{c}\right),$$

$$\Leftrightarrow S_1 = (r_2 - r_1)K^2\left(2 - \exp\left(\frac{1}{c}\right) - \exp\left(-\frac{1}{c}\right)\right)\exp\left(-\frac{x+a}{c}\right)\frac{-\exp\left(\frac{2a+2}{c}\right)}{1-\exp\left(\frac{2}{c}\right)},$$

$$\Leftrightarrow S_1 = (r_2 - r_1)K^2\left(-\frac{\exp\left(\frac{1}{c}\right)}{\tanh\left(\frac{1}{c}\right)} - \frac{-2\exp\left(\frac{2}{c}\right)}{1-\exp\left(\frac{2}{c}\right)}\right)\exp\left(\frac{a-x}{c}\right),$$

$$\Leftrightarrow S_1 = (r_2 - r_1)K^2\exp\left(\frac{2}{c}\right)\tanh\left(\frac{1}{2c}\right)\exp\left(\frac{a-x}{c}\right),$$

and

$$S_2 = \sum_{m=a+1}^{x} u_0''(m)\left(\varphi(a+1-m) - \varphi(a-m)\right),$$

$$\Leftrightarrow S_2 = (r_2 - r_1)(x-a)K^2\left(\exp\left(\frac{1}{c}\right) - 1\right)^2\exp\left(\frac{a-x}{c}\right),$$

and

$$S_3 = \sum_{m=x+1}^{\infty} u_0''(m)\left(\varphi(a+1-m) - \varphi(a-m)\right),$$

$$\Leftrightarrow S_3 = (r_2 - r_1)K^2\left(2 - \exp\left(\frac{1}{c}\right) - \exp\left(-\frac{1}{c}\right)\right)\exp\left(-\frac{x+a}{c}\right)\sum_{m=x+1}^{\infty}\exp\left(\frac{2m}{c}\right),$$

$$\Leftrightarrow S_3 = (r_2 - r_1)K^2\left(2 - \exp\left(\frac{1}{c}\right) - \exp\left(-\frac{1}{c}\right)\right)\exp\left(-\frac{x+a}{c}\right)\frac{-\exp\left(\frac{2a+2}{c}\right)}{1-\exp\left(\frac{2}{c}\right)},$$

$$\Leftrightarrow S_3 = (r_2 - r_1)K^2\left(-\frac{\exp\left(\frac{1}{c}\right)}{\tanh\left(\frac{1}{c}\right)} - \frac{-2\exp\left(\frac{2}{c}\right)}{1-\exp\left(\frac{2}{c}\right)}\right)\exp\left(\frac{a-x}{c}\right)$$

$$\Leftrightarrow S_3 = (r_2 - r_1)K^2\exp\left(-\frac{2}{c}\right)\tanh\left(\frac{1}{2c}\right)\exp\left(\frac{a-x}{c}\right).$$

If the derivative is taken after the jump, we have the following equation:

$$\begin{aligned} p_0'(a) &= u_0'' * \varphi(a+1) - u_0'' * \varphi(a) \\ &= (r_2 - r_1)K^2 \exp\left(\frac{x-a}{c}\right)\left(-2K\cosh\left(\frac{2}{c}\right) - (x-a)\left(\exp\left(\frac{1}{c}\right) - 1\right)^2\right). \end{aligned} \tag{D.18}$$

D.4.4. First topological derivative without noise

According to the topological derivative theory, if the domain is cut at the position a, the topological asymptotic development of the cost function in the continuous case gives the following result:

$$j_a(s * \varphi_a) - j(s * \varphi) = f(\varepsilon) * g(s,a) + o(\varepsilon). \tag{D.19}$$

In the case of the step signal, the first topological derivative gives the following result:

$$\begin{aligned} g(s,a) &= -c^2 u_0'(a) p_0'(a) - \left(u_0'(a)\right)^2 \\ g(s,a) &= (r_2 - r_1)^2 K^2 \left(-2c^2K^2\cosh\left(\frac{2}{c}\right)\right. \\ &\qquad \left. + |x-a|c^2K\left(\exp\left(\frac{1}{c}\right) - 1\right)^2 - 1\right)\exp\left(-2\frac{|x-a|}{c}\right). \end{aligned} \tag{D.20}$$

Figure D.11 presents the variation of the cost function in blue, $-\left(u_0'(x)\right)^2$ in green and the topological gradient in red. The variation of the cost function in the discrete case cannot be realistically compared with the topological asymptotic developpement in the continuous when an infitesimal cut is done in the domain. Figure D.11 illustrates that the topological gradient is more centered on the jump at the location 3000 than the two others curves.

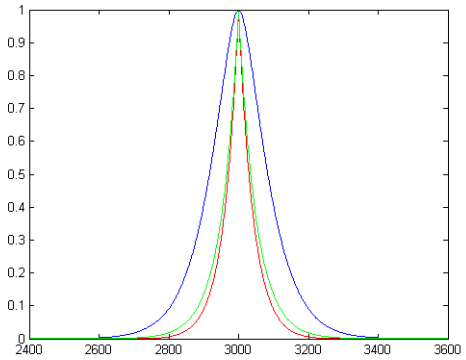


Figure D.11.: Variation of the cost function in blue, $-\left(u_0'(x)\right)^2$ in green and the topological gradient in red. The results have been normalized.

D.5. Tethered Particle Motion

D.5.1. Simulation

The simulation of the data is performed with a dynamical Monte Carlo method. The code in C has been provided by the physicists.

The abscissa data and ordinate data follow the same Gaussian law centred in zero and they have the same variance. The signal $s = \sqrt{x^2 + y^2}$ follows a Rayleigh distribution. The noise has more the appearance of a Gaussian multiplicative noise than the Gaussian additive noise of the theoretical study.

D.5.2. Algorithm

In order to detect transitions, all the information is used and six equations are solved:

$$\begin{aligned}
-c^2 u_x'' + u_x &= |x|, \\
-c^2 u_y'' + u_y &= |y|, \\
-c^2 u_s'' + u_s &= s \\
-c^2 p_x'' + p_x &= u_x'', \\
-c^2 p_y'' + p_y &= u_y'', \\
-c^2 p_s'' + p_s &= u_s''.
\end{aligned}$$

where x is the abscissa data, y is the ordinate data and $s = \sqrt{x^2 + y^2}$.

The topological gradient is computed using the formula:

$$-c^2 u_s'(x) p_s'(x) - \left(u_s'(x)\right)^2 - c^2 u_x'(x) p_x'(x) - \left(u_x'(x)\right)^2 - c^2 u_y'(x) p_y'(x) - \left(u_y'(x)\right)^2 .$$

A hierarchical method is used, the signal is cut in two parts at the minimum of the topological gradient if, at this location, it is under a threshold δ ; otherwise the algorithm stops. If it is cut in two, the same algorithm is applied on each part.

D.5.3. Numerical results

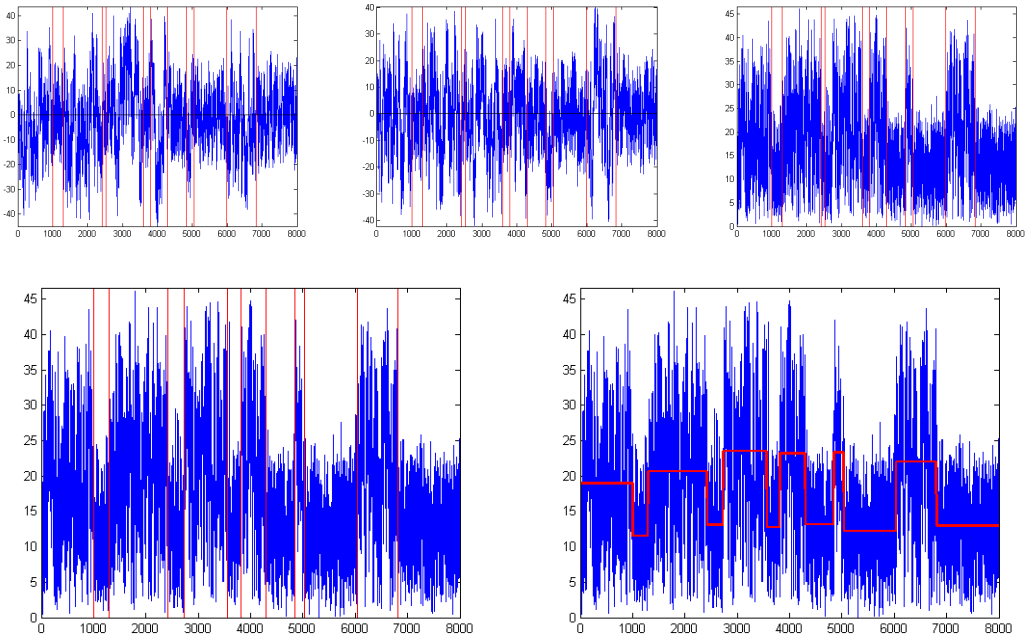


Figure D.12.: First line, from left to right abscissa, ordinate and amplitude motions with the transitions. Second line, the detected transitions.

Algorithm D.5.2 is used to detect the transition of the signal presented on the first line of Figure D.12. The second line of Figure D.12 displays the results. Table D.1 presents the positions of the transitions in the simulation and the results obtained with the topological gradient method. Except for the fourth one, the most difficult one and possibly impossible to detect, the transitions are well detected.

Real transitions	1000	1300	2420	2520	3600	3800	4300	4820	5040	5990
Found transitions	1001	1305	2420	2737	3566	3821	4300	4847	5036	6040
Absolute difference	1	5	0	217	34	21	0	27	4	50

Table D.1.: Transitions determined in the signal presented in Figure D.12.

D.6. Conclusion

The theoretical study demonstrates the interest of using the diffusion equation instead of a sliding window. The diffusion coefficient has to been chosen so as to reduce the noise as long as the differentiation between the different states is possible. The spatial resolution is of relevance to know if we can detect transitions in rectangular signals.

The topological gradient formulation comes from the continuous case, there is a difference with the discrete variation of the cost function but the information is better limited at the transition. The theoretical study proves that the adjoint solution is important. The question of the spatial resolution is also interesting and it is the topic of a work in progress.

The topological gradient has been applied to detect changes in motion amplitude in Tethered Particle Motion and gives promising results. The TPM results could be improved by considering a model which takes into account a multiplicative noise. In addition to that, the simulation has another difference with the theory. The amplitude motions are correlated. This problem could be solved by taking, for example, one sample out of ten. This operation needs to be adapted to real data.

Similar study in two dimensions to provide information on the best choices of diffusion coefficient and threshold is under consideration. Other differential operators in the diffusion equations are also a research perspective.

Articles and talks

Conference article:

S. Larnier and J. Fehrenbach, Edge detection and image restoration with anisotropic topological gradient, ICASSP 2010, Dallas, March 14-19 , 2010.

Submitted articles:

S. Larnier, J. Fehrenbach and M. Masmoudi, Application of topological gradient and anisotropic diffusion in image processing.

S. Larnier and M. Masmoudi, The extended adjoint method.

S. Larnier and R. Mecca, Fractional-order diffusion for image reconstruction. (conference)

S. Larnier, Fractional-order diffusion for color image restoration. (conference)

Talks:

Journées Bordeaux-Pau-Toulouse, Traitement d'images par optimisation topologique, Anglets (France), September 29-30, 2011 (oral presentation)

Workshop Mathématiques-Informatique-Biologie-Santé, Approche mathématiques de la super-résolution, IRIT Toulouse (France), November 23, 2010 (oral presentation)

Journées FREMIT, Application du gradient topologique en traitement d'image, IRIT Toulouse (France), September 20-21, 2010 (oral presentation)

IS10: SIAM Conference on Imaging Science, Topological Gradient for Image Restoration by Anisotropic Diffusion, Chicago (U.S.A.), April 12-14, 2010 (oral presentation)

ICASSP 2010: 35th International Conference on Acoustics, Speech, and Signal Processing, Edge Detection and Image Restoration with Anisotropic Topological Gradient,

Dallas (U.S.A.), March 14-19, 2010 (poster presentation)

Workshop on data and model coupling, Application of Topological Gradient and Anisotropic Diffusion in Image Processing, IMT Toulouse (France), October 12, 2009 (oral presentation)

Spring School: Nouveaux outils mathématiques pour l'analyse d'images et la vision par ordinateur, Application of Topological Gradient and Anisotropic Diffusion in Image Processing, Figeac (France), June 15-19, 2009 (poster presentation)

Bibliography

[1] G. Adde. *Méthodes de traitement d'images appliquées au problème inverse en Magnéto-Electro-Encéphalographie*. PhD thesis, École Nationale des Ponts et Chaussées, 2005. Spécialité informatique.

[2] O. P. Agrawal, J. Sabatier, and J. A. T. Machado. *Advances in fractional calculus: theoretical developments and applications in Physics and Engineering*. Springer, 2007.

[3] G. Allaire. *Shape optimization by the homogenization method*, volume 146. Springer-Verlag, Applied Mathematical Sciences, 2002.

[4] G. Allaire, E. Bonnetier, G. Francfort, and F. Jouve. Shape optimization by the homogenization method. *Numerische Mathematik*, 76:27–68, 1997.

[5] G. Allaire, F. de Gournay, F. Jouve, and A.-M. Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control and Cybernetics*, 34:59–80, 2005.

[6] G. Allaire and F. Jouve. A level-set method for vibration and multiple loads structural optimization. *Computer Methods in Applied Mechanics and Engineering*, 194(30-33):3269 – 3290, 2005. Structural and Design Optimization.

[7] L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via γ-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[8] H. Ammari, E. Bonnetier, Y. Capdeboscq, M. Tanter, and M. Fink. Electrical impedance tomography by elastic deformation. *SIAM Journal on Mathematical Analysis*, 68:1557–1573, 2008.

[9] H. Ammari, E. Iakovleva, D. Lesselier, and G. Perrusson. Music-type electromagnetic imaging of a collection of small three-dimensional inclusions. *SIAM J. Sci. Comput.*, 29:674–709, March 2007.

[10] H. Ammari and H. Kang. High-order terms in the asymptotic expansions of the steady-state voltage potentials in the presence of conductivity inhomogeneities of small diameter. *SIAM Journal on Mathematical Analysis*, 34:1152–1166, 2003.

[11] H. Ammari and H. Kang. *Reconstruction of small inhomogeneities from boundary measurements*, volume 1846. Lecture notes in mathematics, Springer, 2004.

[12] H. Ammari and J. K. Seo. An accurate formula for the reconstruction of conductivity inhomogeneities. *Adv. Appl. Math.*, 30:679–705, May 2003.

[13] S. Amstutz, I. Horchani, and M. Masmoudi. Crack detection by the topological gradient method. *Control and Cybernetics*, 34(1):81–101, 2005.

[14] S. Amstutz, M. Masmoudi, and B. Samet. The topological asymptotic for the Helmoltz equation. *SIAM Journal on Control Optimization*, 42(5):1523–1544, 2003.

[15] G. Aubert, J.-F. Aujol, and L. Blanc-Feraud. Detecting codimension - two objects in an image with Ginzburg-Landau models. *International Journal of Computer Vision*, 65(1-2):29–42, 2005.

[16] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147. Springer-Verlag, Applied Mathematical Sciences, November 2001.

[17] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition - modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, April 2006.

[18] D. Auroux, L. Jaafar Belaid, and B. Rjaibi. Application of the topological gradient method to color image restoration. *SIAM J. Imaging Sci.*, 3(2):153–175, 2010.

[19] D. Auroux and M. Masmoudi. A one-shot inpainting algorithm based on the topological asymptotic analysis. *Computational and Applied Mathematics*, 25(2-3):251–267, 2006.

[20] D. Auroux and M. Masmoudi. Image processing by topological asymptotic analysis. *ESAIM: Proc. Mathematical methods for imaging and inverse problems*, 26:24–44, April 2009.

[21] D. Auroux and M. Masmoudi. Image processing by topological asymptotic expansion. *J. Math. Imaging Vision*, 33(2):122–134, February 2009.

[22] A. E. Badia. Summary of some results on an EEG inverse problem. *J. Neurology and Clinical Neurophysiology*, 1, 2004.

[23] A. E. Badia and M. Farah. Identification of dipole sources in an elliptic equation from boundary measurements : application to the inverse EEG problem. *Journal of Inverse and Ill-posed Problems*, 14(4):331–353, 2006.

[24] J. Bai and X.-C. Feng. Fractional-order anisotropic diffusion for image denoising. *IEEE transactions on image processing*, 16(10):2492–2502, 2007.

[25] M. J. Baillet, S. and R. Leahy. Hybrid MEG/EEG source characterization by cortical remapping and imaging of parametric source models. In *BIOMAG 2000*, August 2000.

[26] S. Baillet, L. Garnero, G. Marin, and J.-P. Hugonin. Combined MEG and EEG source imaging by minimization of mutual information. *Biomedical Engineering, IEEE Transactions on*, 46(5):522–534, may 1999.

[27] L. J. Belaid, M. Jaoua, M. Masmoudi, and L. Siala. Image restoration and edge detection by topological asymptotic expansion. *C. R. Acad. Sci. Paris*, 342(5):313–318, March 2006.

[28] M. Bendsoe and O. Sigmund. *Topology Optimization. Theory, Methods and Applications*. Springer, 2003.

[29] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197 – 224, 1988.

[30] P. Berg. Dipole simulator, http://www.besa.de/updates/tools/.

[31] M. Bonnet. Higher-order topological sensitivity for 2-d potential problems. application to fast identification of inclusions. *International Journal of Solids and Structures*, 46(11-12):2275 – 2292, 2009.

[32] M. Bonnet. Fast identification of cracks using higher-order topological sensitivity for 2-d potential problems. *Engineering Analysis with Boundary Elements*, 35(2):223 – 235, 2011. Special issue on the advances in mesh reduction methods-In honor of Professor Subrata Mukherjee on the occasion of his 65th birthday.

[33] F. Bornemann and T. Marz. Fast image inpainting based on coherence transport. *J. Math. Imaging and Vision*, 28(3):259–278, March 2007.

[34] A. Buades, B. Coll, and J. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation (SIAM interdisciplinary journal)*, 4(2):490–530, 2005.

[35] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[36] Y. Capdeboscq and M. S. Vogelius. A general representation formula for boundary voltage perturbations caused by internal conductivity inhomogeneities of low volume fraction. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:159–173, 2003.

[37] Y. Capdeboscq and M. S. Vogelius. Optimal asymptotic estimates for the volume of internal inhomogeneities in terms of multiple boundary measurements. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:227–240, 2003.

[38] T. F. Chan and J. Shen. Variational image inpainting. *Comunication on Pure and Applied Mathematics*, LVIII:579–619, 2005.

[39] P. Crispel, P. Degond, and M.-H. Vignal. An asymptotic preserving scheme for the two-fluid Euler-Poisson model in the quasineutral limit. *Journal of Computational Physics*, 223:208–234, 2007.

[40] E. Cuesta and J. Codes. Image processing by means of a linear integro-differential equation. In *3rd IASTED Int. Conf. Visualization, Imaging and Image Processing*, volume 1, pages 438–442, 2003.

[41] R. G. de Peralta Menendez, M. M. Murray, C. M. Michel, R. Martuzzi, and S. L. G. Andino. Electrical neuroimaging based on biophysical constraints. *NeuroImage*, 21(2):527 – 539, 2004.

[42] S. Didasa, B. Burgeth, A. Imiya, and J. Weickert. Regularity and scalespace properties of fractional high order linear filtering. *Scale Spaces and PDE Methods in Computer Vision*, pages 13–25, 2005.

[43] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1033, Washington, DC, USA, 1999. IEEE Computer Society.

[44] J. Fehrenbach and M. Masmoudi. Coupling topological gradient and gauss-newton methods. In *IUTAM Symposium on Topological Design Optimization, Bendsoe-Olhoff-Sigmund editors (Springer)*, 2006.

[45] J. Fehrenbach and M. Masmoudi. A fast algorithm for image registration. *Comptes Rendus Mathématique*, 346(9-10):593–598, 2008.

[46] J. Fehrenbach, M. Masmoudi, R. Souchon, and P. Trompette. Detection of small inclusions using elastography. *Inverse Problems*, 22:1055–1069, 2006.

[47] D. Fender. *Source localization of brain electrical activity*. Elsevier, Amsterdam, 1987.

[48] L. Garnero. *Tomography of Electrical Cerebral Activity in Magneto- and Electroencephalography*, pages 393–409. ISTE, 2010.

[49] S. Garreau, P. Guillaume, and M. Masmoudi. The topological asymptotic for pde systems: The elasticity case. *SIAM Journal on Control and Optimization*, 39(6):1756–1778, 2001.

[50] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. http://www.geuz.org/gmsh/.

[51] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[52] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[53] T. Goldstein and S. Osher. The split Bregman method for L1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[54] R. Grech, T. Cassar, J. Muscat, K. Camilleri, S. Fabri, M. Zervakis, P. Xanthopoulos, V. Sakkalis, and B. Vanrumste. Review on solving the inverse problem in EEG source analysis. *Journal of NeuroEngineering and Rehabilitation*, 5(1):25, 2008.

[55] P. Guillaume and M. Hassine. Removing holes in topological shape optimization. *ESAIM Control, Optimisation and Calculus of Variations*, 14(1):160–191, 2008.

[56] P. Guillaume and K. S. Idris. The topological asymptotic expansion for the Dirichlet problem. *SIAM Journal on Control and Optimization*, 41(4):1042–1072, 2002.

[57] P. Guillaume and K. S. Idris. The topological sensitivity and shape optimization for the Stokes equations. *SIAM Journal on Control and Optimization*, 43(1):1–31, 2004.

[58] B. Gunturk and R. Altunbasak, Y.and Mersereau. Color plane interpolation using alternating projections. *Image Processing, IEEE Transactions on*, 11:997 –1013, September 2002.

[59] H. Ammari, S. Moskow, and M. S. Vogelius. Boundary integral formulae for the reconstruction of electric and electromagnetic inhomogeneities of small volume. *ESAIM: COCV*, 9:49–66, 2003.

[60] H. Hallez, B. Vanrumste, R. Grech, J. Muscat, W. D. Clercq, A. Vergult, K. Camilleri, S. Fabri, and S. V. Huffel. Review on solving the forward problem in EEG source analysis. *Journal of NeuroEngineering and Rehabilitation*, 4, 2007.

[61] H. Hallez, B. Vanrumste, P. V. Hese, Y. D'Asseler, I. Lemahieu, and R. V. de Walle. A finite difference method with reciprocity used to incorporate anisotropy in electroencephalogram dipole source localization. *Physics in Medicine and Biology*, 50, 2005.

[62] M. Hassine, S. Jan, and M. Masmoudi. From differential calculus to 0-1 topological optimization. *SIAM, Journal on Control and Optimization*, 45(6):1965–1987, 2007.

[63] K. Hirakawa and T. Parks. Joint demosaicing and denoising. *Image Processing, IEEE Transactions on Image Processing*, 15:2146–2157, August 2006.

[64] T. Y. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134:169–189, 1997.

[65] J. M. Hyman and M. Shashkov. Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids. *Applied Numerical Mathematics*, 25(4):413–442, December 1997.

[66] M. Hämäläinen, R. Hari, R. Iimoniemi, J. Knuutila, and O. Lounasmaa. Magnetoencephalography - theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*, 65(2):413–497, 1993.

[67] B. Kawohl. From Mumford-Shah to Perona-Malik in image processing. *Mathematical Methods in the Applied Sciences*, 27:1099–1476, 2004.

[68] B. Klimpke. A hybrid magnetic field solver using a combined finite element/boundary element field solver http://www.integratedsoft/papers/research/hybrid/.

[69] R. V. Kohn and M. Vogelius. Relaxation of a variational method for impedance computed tomography. *Communications on Pure and Applied Mathematics*, 40(6):745–777, 1987.

[70] H. Krim and M. Viberg. Two decades of array signal processing research. *IEEE Signal Processing Magazine*, 13:67–94, 1996.

[71] S. Larnier and J. Fehrenbach. Edge detection and image restoration with anisotropic topological gradient. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1362 –1365, march 2010.

[72] A. K. Liu, A. M. Dale, and J. W. Belliveau. Monte Carlo simulation studies of EEG and MEG localization accuracy. *Human Brain Mapping*, 16(1):47–62, 2002.

[73] M. Mainberger and J. Weickert. Edge-based image compression with homogeneous diffusion. In *CAIP '09: Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns*, pages 476–483, Berlin, Heidelberg, 2009. Springer-Verlag.

[74] F. Malgouyres and F. Guichard. Edge direction preserving image zooming: a mathematical and numerical analysis. *SIAM Journal on Numerical Analysis*, 39(1):1–37, 2002.

[75] M. Masmoudi. The topological asymptotic. *Glowinski, R., Karawada, H., Periaux, J. (eds.) Computational Methods for Control Applications, GAKUTO international series, Mathematical science and applications*, 16:53–72, 2001.

[76] K. S. Miller and B. Ross. *An introduction to the fractional calculus and fractional differential equations*. Wiley, 1993.

[77] B. Mohammadi and O. Pironneau. Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics*, 36:255–279, 2003.

[78] M. Morini and M. Nigri. Mumford-Shah functional as Γ-limit of discrete Perona-Malik energies. *Mathematical Models and Methods in the Applied Sciences*, 13:785–805, 2003.

[79] J. Mosher and R. Leahy. Recursive MUSIC: A framework for EEG and MEG source localization. *Biomedical Engineering, IEEE Transactions on*, 45(11):1342–1354, November 1998.

[80] J. Mosher and R. Leahy. Source localization using recursively applied and projected (rap) music. *IEEE Transactions on Signal Processing*, 47(2):332 –340, February 1999.

[81] J. C. Mosher, P. S. Lewis, and R. Leahy. Multiple dipole modeling and localization from spatio-temporal MEG data. *IEEE Transactions on Biomedical Engineering*, 39(6):541–553, 1992.

[82] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[83] J. D. Munck. The estimation of time varying dipoles on the basis of evoked potentials. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 77(2):156 – 160, 1990.

[84] M. Ng, P. Weiss, and X.-M. Yuan. Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. *SIAM journal on Scientific Computing*, 2010. Accepted for publication.

[85] K. B. Oldham and J. Spanier. *The Fractional Calculus - Theory and Applications of Differentiation and Integration to Arbitrary Order*. Academic Press, 1974.

[86] J. Ophir, S. Alam, B. Garra, F. Kallel, E. Konofagou, T. Krouskop, C. Merritt, R. Righetti, R. Souchon, S. Srinivan, and T. Varghese. Elastography: Imaging the elastic properties of soft tissues with ultrasound. *J Med Ultrasonics*, 29:155–171, 2002.

[87] J. Ophir, I. Céspedes, H. Ponnekanti, Y. Yazdi, and X. Li. Elastography: A quantitative method for imaging the elasticity of biological tissues. *Ultrasonic Imaging*, 13(2):111 – 134, 1991.

[88] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[89] S. J. Osher and F. Santosa. Level set methods for optimization problems involving geometry and constraints: I. frequencies of a two-density inhomogeneous drum. *Journal of Computational Physics*, 171(1):272 – 288, 2001.

[90] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, 2003.

[91] R. Pascual-Marqui. Standardized low-resolution brain electromagnetic tomography (sLORETA): technical details. *Methods and Findings in Experimental and Clinical Pharmacology*, 24 Suppl D:5–12, 2002.

[92] R. Pascual-Marqui, C. Michel, and D. Lehmann. Low resolution electromagnetic tomography: a new method for localizing electrical activity in the brain. *International Journal of Psychophysiolog*, 18:49–65, 1994.

[93] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Proc. IEEE Comp. Soc. Workshop on Computer Vision (Miami Beach, Nov. 30 - Dec. 2, 1987)*, pages 16–22, 1987.

[94] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990.

[95] G. Peyre. Non-local means toolbox, http://www.mathworks.com/matlabcentral/fileexchange/13619.

[96] J. Pommier and Y. Renard. Getfem++. an open source generic C++ library for Finite Element Methods. http://download.gna.org/getfem/html/homepage/.

[97] Y. Pu, W. Wang, J. Zhou, Y. Wang, and H. Jia. Fractional differential approach to detecting textural features of digital image and its fractional differential filter implementation. *Sci China Ser F-Inf Sci*, 51(9):1319–1339, September 2008.

[98] Y. Pu, J. Zhou, and X. Yuan. Fractional differential mask: A fractional differential-based approach for multiscale texture enhancement. *IEEE transactions on image processing*, 9(2):491–511, 2010.

[99] L. Qiu, Y. Li, and D. Yao. A feasibility study of EEG dipole source localization using particle swarm optimization. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 720 –726 Vol.1, sept. 2005.

[100] A. Rodríguez-Rivera, B. V. Veen, and R. Wakai. Statistical performance analysis of signal variance based dipole models for MEG/EEG source localization and detection. *IEEE Transactions On Biomedical Engineering*, 50:137–149., 2003.

[101] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Communication on Pure and Applied Mathematics*, 60(1):259–268, 1992.

[102] R. Sadleir and A. Argibay. Modeling skull electrical properties. *Annals of Biomedical Engineering*, 35:1699–1712, 2007.

[103] B. Samet. The topological asymptotic with respect to a singular boundary perturbation. *Comptes Rendus Mathematique*, 336, issue 12:1033–1038, June 2003.

[104] C. Schmaltz, J. Weickert, and A. Bruhn. Beating the quality of jpeg 2000 with anisotropic diffusion. In *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, pages 452–461, Berlin, Heidelberg, 2009. Springer-Verlag.

[105] A. Schumacher. *Topologieoptimisierung von Bauteilstrukturen unter Verwendung von Lopchpositionierungkrieterien*. PhD thesis, Universitat-Gesamthochschule Siegen, Germany, 1995.

[106] S. Setzer. Scale space and variational methods in computer vision. *Lecture Notes in Computer Science*, 5567:464–476, 2009.

[107] J. Sokolowski and A. Zochowski. On the topological derivative in shape optimization. *SIAM Journal on Control and Optimization*, 37:1241–1272, 1999.

[108] G. Team. The gnu image manipulation program. *http://www.gimp.org*.

[109] D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *International Journal of Computer Vision*, 68(1):65–82, June 2006.

[110] B. V. Veen and K. Buckley. Beamforming : a versatile approach to spatial filtering. *IEEE ASSP magazine*, 5(2):4–23, 1988.

[111] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1-2):227 – 246, 2003.

[112] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment : From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[113] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[114] J. Weickert. *Anisotropic Diffusion in Image Processing*. Ph.d. thesis, Dept. of Mathematics, University of Kaiserslautern, Germany, 1996.

[115] P. Weiss, L. Blanc-Féraud, and G. Aubert. Efficient schemes for total variation minimization under constraints in image processing. *SIAM journal on Scientific Computing*, 31(3):2047–2080, April 2009.

[116] W. Zeng and X. Lu. Generalized pde framework for image super resolution. *Artificial Intelligence and Computational Intelligence, International Conference on*, 2:230–234, 2010.

Summary

Nowadays, topology optimization has been extensively studied in structural optimization which is a major interest in the design of mechanical systems in the industry and in inverse problems with the detection of defects or inclusions. This work focuses on the topological derivative approach and proposes a more flexible generalization of this method making it possible to address new applications.

In a first part, we study classical image processing problems (restoration, inpainting), and give a common framework to theses problems. We focus on anisotropic diffusion and consider a new problem: super-resolution. Our approach seems to be powerful in comparison with other methods.

Topological derivative method has some drawbacks: it is limited to simple problems, we do not know how to fill holes, ... In a second part, to overcome these difficulties, an extension of the adjoint method is presented. Named the numerical vault, it allows us to consider new fields of applications and to explore new theoretical investigations in the area of topological derivative.

Résumé

De nos jours, l'optimisation topologique a été largement étudiée en optimisation de structure, problème majeur en conception de systèmes mécaniques pour l'industrie et dans les problèmes inverses avec la détection de défauts et d'inclusions. Ce travail se concentre sur les approches de dérivées topologiques et propose une généralisation plus flexible de cette méthode rendant possible l'investigation de nouvelles applications.

Dans une première partie, nous étudions des problèmes classiques en traitement d'images (restauration, inpainting), et exposons une formulation commune à ces problèmes. Nous nous concentrons sur la diffusion anisotrope et considérons un nouveau problème : la super-résolution. Notre approche semble meilleure comparée aux autres méthodes.

L'utilisation des dérivées topologiques souffre d'inconvénients : elle est limitée à des problèmes simples, nous ne savons pas comment remplir des trous ... Dans une seconde partie, une nouvelle méthode visant à surmonter ces difficultés est présentée. Cette approche, nommée voûte numérique, est une extension de la méthode adjointe. Ce nouvel outil nous permet de considérer de nouveaux champs d'application et de réaliser de nouvelles investigations théoriques dans le domaine des dérivées topologiques.