

# Dissertation

presented to obtain the

## HABILITATION À DIRIGER DES RECHERCHES

Université Paul Sabatier Toulouse 3

Mention: Applied Mathematics

by

**Didier AUROUX**

### Fast algorithms for image processing and data assimilation

Defended on 26 November 2008

#### After reviews by:

G. AUBERT, Professor	Université de Nice Sophia-Antipolis
P. ROUCHON, Professor	Mines ParisTech
F. SANTOSA, Professor	University of Minnesota

#### Commitee members:

G. AUBERT, Professor	Université de Nice Sophia-Antipolis	<i>(Reviewer)</i>
J. BLUM, Professor	Université de Nice Sophia-Antipolis	<i>(Examiner)</i>
L. COHEN, Research director	CNRS & Université Paris Dauphine	<i>(Examiner)</i>
P. DEGOND, Research director	CNRS & Université Paul Sabatier	<i>(Examiner)</i>
M. MASMOUDI, Professor	Université Paul Sabatier, Toulouse	<i>(Advisor)</i>
J.-P. PUEL, Professor	Université de Versailles Saint-Quentin	<i>(Examiner)</i>
P. ROUCHON, Professor	Mines ParisTech	<i>(Reviewer)</i>



# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Image processing</b>	<b>11</b>
2.1	Introduction . . . . .	11
2.2	Topological asymptotic analysis . . . . .	12
2.2.1	Presentation of the method . . . . .	12
2.2.2	Main result . . . . .	12
2.3	Inpainting [9, 13] . . . . .	13
2.3.1	Crack localization problem . . . . .	14
2.3.2	Dirichlet and Neumann formulations for the inpainting problem . . . . .	14
2.3.3	Asymptotic expansion . . . . .	15
2.3.4	Algorithm . . . . .	16
2.3.5	Remarks . . . . .	17
2.4	Restoration [16, 22] . . . . .	17
2.4.1	Variational formulation . . . . .	17
2.4.2	Topological gradient . . . . .	18
2.4.3	Algorithm . . . . .	19
2.4.4	Remarks . . . . .	19
2.4.5	Extension to color images . . . . .	19
2.5	Classification [10, 16] . . . . .	21
2.5.1	Introduction to the classification problem . . . . .	21
2.5.2	Restoration and classification coupling . . . . .	21
2.5.3	Extension to unsupervised classification . . . . .	22
2.6	Segmentation [12, 13] . . . . .	23
2.6.1	From restoration to segmentation . . . . .	23
2.6.2	Power series expansion . . . . .	24
2.6.3	Algorithm . . . . .	25
2.7	Complexity and speeding up [13, 16] . . . . .	26
2.7.1	Discrete cosine transform . . . . .	26
2.7.2	Preconditioned conjugate gradient . . . . .	27
2.8	Topological gradient and minimal paths [26] . . . . .	27
2.8.1	Minimal paths . . . . .	28
2.8.2	Fast marching . . . . .	28

2.8.3	Coupled algorithm . . . . .	29
2.9	Conclusions and perspectives . . . . .	30
<b>3</b>	<b>Back and forth nudging</b>	<b>31</b>
3.1	Introduction . . . . .	31
3.2	Back and Forth Nudging (BFN) algorithm [8, 11, 18] . . . . .	33
3.2.1	Forward nudging . . . . .	33
3.2.2	Backward nudging . . . . .	34
3.2.3	BFN algorithm . . . . .	34
3.2.4	Choice of the nudging matrices and interpretation . . . . .	35
3.3	Numerical experiments [11, 14, 21] . . . . .	37
3.3.1	Numerical choice of the nudging matrices . . . . .	37
3.3.2	Experimental approach . . . . .	37
3.3.3	Physical models . . . . .	38
	Lorenz equations . . . . .	38
	Viscous Burgers' equation . . . . .	38
	Shallow water model . . . . .	39
	Layered quasi-geostrophic ocean model . . . . .	39
3.3.4	Conclusions emerging from the numerical experiments . . . . .	40
3.4	Theoretical convergence results [18, 24] . . . . .	41
3.4.1	Linear case . . . . .	41
3.4.2	Transport equations . . . . .	42
	Linear viscous transport . . . . .	42
	Viscous Burgers . . . . .	44
	Inviscid linear transport . . . . .	45
	Inviscid Burgers . . . . .	46
3.5	Nudging and observers [25] . . . . .	47
3.5.1	Observers for a shallow water model . . . . .	48
3.5.2	Invariant correction terms . . . . .	48
3.5.3	Convergence study on a linearized simplified system . . . . .	49
3.5.4	Numerical experiments . . . . .	51
3.6	Conclusion . . . . .	52
<b>4</b>	<b>Image data assimilation</b>	<b>53</b>
4.1	Introduction . . . . .	53
4.2	Description of the algorithm [23] . . . . .	54
4.2.1	Constant brightness assumption . . . . .	54
4.2.2	Cost function . . . . .	55
4.2.3	Regularization . . . . .	55
4.2.4	Muti-grid approach and optimization . . . . .	55
4.2.5	Quality estimate . . . . .	57
4.3	Numerical experiments [23] . . . . .	57
4.3.1	Simulated data . . . . .	57
4.3.2	Experimental data . . . . .	58

4.4 Conclusions . . . . .	59
<b>5 General conclusions and perspectives</b>	<b>61</b>
<b>List of publications</b>	<b>63</b>
Publications linked to the PhD thesis . . . . .	63
Publications after the PhD thesis . . . . .	64
<b>References</b>	<b>67</b>



# Chapter 1

## Introduction

In this work, several problems have been studied with the common goal of providing robust, particularly easy to implement, fast and powerful algorithms. The efficiency of the algorithms is required by the operational context of the methods, and by a need to process more and more data in an increasingly short time. The other constraint that we particularly took into account is the ease of use and implementation of the methods we have developed.

In chapter 2, we will tackle various problems in image processing by an original approach in the field: topological asymptotic analysis, or more simply the topological gradient.

There has recently been a renewed interest in image processing thanks to new applications in telecommunications and medicine: on one hand, new technologies in telecommunications and diffusion of information, which now involve sending and receiving massive flows of numerical data (e.g. images), and on the other hand the medical world, in which huge progress has been made, in particular for the early detection of tumors, thanks to more powerful imaging techniques.

Our study is motivated by several observations. First, the topological gradient is generally used for structural mechanics, design, and shape optimization problems. Also, it has been successfully applied in electromagnetism for the detection of cracks or hidden objects. However, many image processing problems rely on the good identification of a subset of the image, for instance edges or characteristic objects. This common feature seemed interesting to us, and allowed us to adapt the topological gradient method, initially used for crack detection, to several image processing problems (restoration, classification, segmentation, inpainting).

The second interesting aspect is the speed of the method. In various fields, topological asymptotic analysis has made it possible to obtain good results very quickly. However, medical imaging and audiovisual diffusion (e.g. satellite television or internet broadcasting) both require the processing time to be negligible. If the processing time is too large, it will delay the medical diagnosis, or the flow of data. It is thus important to build extremely fast schemes for solving these various problems, in real

time for movies and a negligible time (e.g. smaller than one second) for images.

As we will see hereafter, the topological gradient method actually adapts perfectly to image processing problems, allowing us to obtain very interesting results for a particularly small computation cost.

In chapter 3, we will study data assimilation for environmental and geophysical problems, and more particularly within the framework of atmospheric and oceanic observations. For several years, one of the major concerns has been to appreciably improve our knowledge of these turbulent systems, one of the major goals being the ability to predict their evolution with a high reliability.

Several different challenges appear in data assimilation: short-range (e.g. a few days) weather forecasting, the study of global warming and climate change, detection of extreme climatic phenomena several weeks in advance, . . . For all these problems, the goals are almost similar. They consist of estimating quickly and with a very high degree of accuracy the state of a turbulent system, from the combined knowledge of models and data: on one hand mathematical equations modeling the coupled atmosphere-ocean system, and on the other hand observations of different nature (e.g. in situ, or satellite observations), corresponding to various physical quantities.

Beyond the extreme size of the problem to be solved (several billions of values to be identified from hundreds of millions of observations) and the computational time needed to solve it, another factor appears: the cost of development and use of a data assimilation method. Presently, it is extremely difficult to implement such a method, even on a relatively simple problem. This motivated us to study the possibility of improving one of the simplest methods of data assimilation, nudging (also known as Newtonian relaxation), in order to obtain much better results without complicating the method.

By applying the nudging method to the backward (in time) problem, we noted that it is possible to stabilize the backward system, which is unstable because of the irreversibility of the physical problem. Thus, as detailed in chapter 3, we can go back in time, and obtain a more reliable estimate of the system at a previous time, from which forecasts may be deduced. By applying alternatively and repeatedly the standard nudging method to the forward and backward models, we obtain an iterative algorithm that is very easy to implement and provides definitely better results than the standard nudging. Indeed, the results are of similar quality, and are often obtained much more quickly than by using the standard variational data assimilation method.

Chapter 4 presents a study at the interface of these two fields: the assimilation of images. Presently, a huge quantity of observations coming from satellite images is essentially not used to improve the knowledge of the system state. However, sequences of images obtained by satellites definitely show various characteristic structures (hurricanes, swirls, currents of hot water, pollution, . . .) moving and evolving in time.

Several approaches can be considered to solve this kind of problem, and we made the choice to try to identify and extract velocity fields from the sequences of images.



That appeared to us to be the most adapted choice for rapidly extracting conventional data (i.e. directly related to the model variables), and then being able to use them in a standard assimilation system.

The idea that we develop in chapter 4 is based on the constant brightness assumption, which consists of looking for a displacement field that transports an image to another one. The originality of our approach lies in the nonlinearization of the cost function to be minimized, combined with a fast method to assemble the Jacobian matrix. Finally, a multi-grid approach makes it possible to guarantee the quality of the minimum. Thanks to all these techniques, we are able to extract complete velocity fields in a very short time, and it is also possible to provide a quality estimate of the identified fields, which can be viewed as error statistics of these pseudo-observations within the framework of data assimilation.

Finally some general conclusions and research perspectives are given in chapter 5.



## Chapter 2

# Image processing by topological asymptotic analysis

This chapter summarizes the work presented in [9, 10, 12, 13, 16, 22, 26].

### 2.1 Introduction

The idea of topological asymptotic analysis is to measure the impact of a perturbation of the domain on a cost function. We only consider here the approach that has been introduced for topological optimization purpose, in which the goal is to identify an optimal shape and its complementary in a given domain [133, 98, 104].

Topological shape optimization seems particularly well adapted to solve image processing problems (like classification, segmentation, enhancement, inpainting, . . .), as they mainly consist of identifying a particular subdomain of the image: its edges.

At first sight, the main issue of topological shape analysis is the non-differentiability of the problem. To find the optimal domain is indeed equivalent to identify its characteristic function. Several classical approaches have been developed to make this problem differentiable. We can cite here the relaxation technique, which allows the characteristic function to take all possible values in the interval  $[0; 1]$ , and the level set approach where the characteristic function is replaced by a regular level set function which is positive inside the optimal domain and negative outside [133, 34, 33, 36, 51, 157].

The idea of topological asymptotic analysis is to switch the characteristic function from one to zero (or from zero to one) in a (infinitely) small area. Thus, the variation of the cost function is small when we switch a very small part from the subdomain to its complementary. The topological asymptotic expansion provides this variation, and allows one to derive a *topological* gradient of the cost function [133, 98, 158, 157].

In this chapter, we first present the basic tools of topological asymptotic analysis, and we then study several applications to image processing problems: inpainting (where the goal is to fill a hidden part of an image), restoration and enhancement, classification, and segmentation. Then, we present a very efficient way to speed up

all the algorithms introduced in this chapter, based on discrete cosine transforms and an appropriate preconditioning. Finally, we present a coupled approach combining the topological gradient and the minimal path technique in order to improve the edge detection, and to avoid non-connex contours.

## 2.2 Topological asymptotic analysis

### 2.2.1 Presentation of the method

Let  $\Omega$  be a regular open bounded domain of  $\mathbb{R}^2$  (or  $\mathbb{R}^3$ ). Let us consider a Partial Differential Equation (PDE) problem defined in  $\Omega$ , written in its variational formulation:

$$\text{find } u \in \mathcal{V} \text{ such that } a(u, w) = l(w), \forall w \in \mathcal{V}, \quad (2.1)$$

where  $\mathcal{V}$  is a Hilbert space on  $\Omega$ , usually  $H^1(\Omega)$ ,  $a$  is a bilinear continuous and coercive form defined on  $\mathcal{V}$ , and  $l$  is a linear continuous form on  $\mathcal{V}$ . We finally consider a cost function  $J(\Omega, u)$  to be minimized, where  $u$  is the solution of equation (2.1).

We now consider a small perturbation of the domain, e.g. by the insertion of a crack  $\sigma_\rho = x_0 + \rho\sigma(n)$ , where  $x_0 \in \Omega$  represents the point where the crack is inserted,  $\sigma(n)$  is a straight crack containing the origin of the domain, and  $n$  is a unit vector normal to the crack. Finally,  $\rho > 0$  represents the size of the perturbation, assumed to be small. Let  $\Omega_\rho = \Omega \setminus \sigma_\rho$  be the perturbed domain. We can consider the same PDE problem as before, but on the perturbed domain:

$$\text{find } u_\rho \in \mathcal{V}_\rho \text{ such that } a_\rho(u_\rho, w) = l_\rho(w), \forall w \in \mathcal{V}_\rho, \quad (2.2)$$

where  $\mathcal{V}_\rho$ ,  $a_\rho$  and  $l_\rho$  represent the restriction of the Hilbert space  $\mathcal{V}$  to  $\Omega_\rho$ , and the perturbed bilinear and linear forms respectively.

We can rewrite the cost function  $J$  as a function of  $\rho$  by considering the following map:

$$j : \rho \mapsto \Omega_\rho \mapsto u_\rho \text{ solution of (2.2)} \mapsto j(\rho) := J(\Omega_\rho, u_\rho). \quad (2.3)$$

The topological sensitivity theory provides an asymptotic expansion of  $j$  when  $\rho$  tends to zero. It takes the general form:

$$j(\rho) - j(0) = f(\rho)G(x_0) + o(f(\rho)), \quad (2.4)$$

where  $f(\rho)$  is an explicit positive function going to zero with  $\rho$ , and  $G(x_0)$  is called the topological gradient at point  $x_0$  [133].

Then to minimize the criterion  $j$ , one has to insert small holes (or cracks) at points where the topological gradient  $G$  is the most negative, in order to make the cost function  $j$  decrease quickly (see the asymptotic expansion (2.4)).

### 2.2.2 Main result

In the following, we will consider several times this main result [37]:

**Theorem 2.1** *If there exists a linear form  $L_\rho$  defined on  $\mathcal{V}_\rho$ , a function  $f : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ , and four real numbers  $\delta J_1$ ,  $\delta J_2$ ,  $\delta a$  and  $\delta l$  such that*

- $\lim_{\rho \rightarrow 0} f(\rho) = 0$ ,
- $J(\Omega_\rho, u_\rho) - J(\Omega_\rho, u_0) = L_\rho(u_\rho - u_0) + f(\rho)\delta J_1 + o(f(\rho))$ ,
- $J(\Omega_\rho, u_0) - J(\Omega, u_0) = f(\rho)\delta J_2 + o(f(\rho))$ ,
- $(a_\rho - a_0)(u_0, p_\rho) = f(\rho)\delta a + o(f(\rho))$ ,
- $(l_\rho - l_0)(p_\rho) = f(\rho)\delta l + o(f(\rho))$ ,

where the adjoint state  $p_\rho$  is the solution of the adjoint equation

$$a_\rho(w, p_\rho) = -L_\rho(w), \forall w \in \mathcal{V}_\rho, \quad (2.5)$$

and  $u_\rho$  is the solution of the direct equation (2.2), then the cost function  $j$  has the asymptotic expansion (2.4), where the topological gradient  $G(x)$  is given by

$$G(x) = \delta J_1 + \delta J_2 + \delta a - \delta l. \quad (2.6)$$

## 2.3 Inpainting [9, 13]

In this section, we present an application of the topological asymptotic analysis to the inpainting problem. The goal of inpainting is to fill a hidden part of an image. In other words, if we denote by  $\Omega$  the original image and  $\omega$  the hidden part of the image, the goal is to recover the hidden part  $\omega$  from the known part of the image  $\Omega \setminus \omega$ . There are many applications, for instance removing some spots on a badly preserved movie or image, or deleting encrusted logos and images on television programs, . . .

This problem has been widely studied. Several methods have been considered: learning approaches (neural networks, radial basis functions, support vector machine, . . .), in which the learning data is taken in  $\Omega \setminus \omega$ , and then the approximate function is evaluated in  $\omega$  [177, 178]; minimization of an energy cost function in  $\omega$  based on a total variation norm [67, 68]; morphological analysis for the reconstruction of both cartoon and texture [87]; . . .

In order to study the inpainting problem, we first consider a crack localization method. Crack detection allows us to identify the edges of the hidden part of the image, and the inpainting problem can then be easily solved. We will consider the classical thermal diffusion technique [142, 66, 174, 175, 150] and improve it by modeling the edges by cracks. These cracks are supposed to be highly insulating and to allow the temperature to jump across edges. As both the Dirichlet and Neumann conditions are known on the boundary of the hidden subset, we can define a criterion measuring the discrepancy between the solutions of a Dirichlet and a Neumann problem respectively [118]. This problem is similar to the inverse conductivity problem, also known as the Calderón problem [65], which consists of identifying the coefficients

of a partial differential equation from the knowledge of the Dirichlet to Neumann operator. Only two measurements are needed to recover several simple cracks [30, 31, 48]. From the numerical point of view, several methods [40, 49, 50, 64, 96, 152, 151] have been proposed, but the topological gradient approach seems to be the most efficient method for crack localization. The minimization of the criterion allows us to identify the main edges inside the hidden part of the image. The image is finally filled between the edges thanks to the Laplace operator.

This section summarizes the work introduced in [9, 13]. We also refer to these references for the results of many numerical experiments.

### 2.3.1 Crack localization problem

Let  $\Omega$  be a bounded open set of  $\mathbb{R}^2$ . We assume in this section that  $\Omega$  contains a perfectly insulating crack  $\sigma^*$ . We impose a flux  $\phi \in H^{-1/2}(\Gamma)$  on the boundary  $\Gamma$  of  $\Omega$ , and we want to find  $\sigma \subset \Omega$  such that the solution  $u \in H^1(\Omega \setminus \sigma)$  of

$$\begin{cases} \Delta u = 0 & \text{in } \Omega \setminus \sigma, \\ \partial_n u = \phi & \text{on } \Gamma, \\ \partial_n u = 0 & \text{on } \sigma, \end{cases} \quad (2.7)$$

satisfies  $u|_{\Gamma} = T$ , where  $T \in H^{1/2}(\Gamma)$  is a given function. We also assume some compatibility conditions in order to have a well-posed direct problem.

A topological gradient approach has been introduced in [37], and consists of defining a Dirichlet and a Neumann problem, as we have an over-determination in the boundary conditions:

$$u_D \in H^1(\Omega \setminus \sigma) \text{ such that } \begin{cases} \Delta u_D = 0 & \text{in } \Omega \setminus \sigma, \\ u_D = T & \text{on } \Gamma, \\ \partial_n u_D = 0 & \text{on } \sigma, \end{cases} \quad (2.8)$$

$$u_N \in H^1(\Omega \setminus \sigma) \text{ such that } \begin{cases} \Delta u_N = 0 & \text{in } \Omega \setminus \sigma, \\ \partial_n u_N = \phi & \text{on } \Gamma, \\ \partial_n u_N = 0 & \text{on } \sigma. \end{cases} \quad (2.9)$$

It is clear that for the actual crack  $\sigma^*$ , the two solutions  $u_D$  and  $u_N$  are equal. The idea is then to consider and minimize the following cost function

$$J(\sigma) = \frac{1}{2} \|u_D - u_N\|_{L^2(\Omega)}^2. \quad (2.10)$$

The topological asymptotic expansion of this cost function is detailed in [37].

### 2.3.2 Dirichlet and Neumann formulations for the inpainting problem

In our approach, we now denote by  $\Omega$  the image and  $\Gamma$  its boundary,  $\omega \subset \Omega$  the missing part of the image and  $\gamma$  its boundary. Let  $v$  be the image that we want to restore. We assume that  $v$  is known in  $\Omega \setminus \omega$ , and unknown in  $\omega$ .

The idea is to adapt the crack localization method to inpainting: crack detection first allows us to identify the cracks (or edges)  $\sigma$  of the hidden part  $\omega$  of the image, and then we will impose that the Laplacian of the restored image is equal to zero in  $\omega \setminus \sigma$ . For a given crack  $\sigma \subset \omega$ , as  $v$  (Dirichlet condition) and  $\partial_n v$  (Neumann condition) are known on the boundary  $\gamma$  of  $\omega$ , we can solve two different problems inside  $\omega$ .

For a given crack  $\sigma$ , we denote by  $u_D \in H^1(\Omega \setminus \sigma)$  the solution of the following Dirichlet problem:

$$\begin{cases} \Delta u_D = 0 & \text{in } \omega \setminus \sigma, \\ u_D = v & \text{on } \gamma, \\ \partial_n u_D = 0 & \text{on } \sigma, \\ u_D = v & \text{in } \Omega \setminus \omega. \end{cases} \quad (2.11)$$

Outside  $\omega$ , the solution is equal to the original image, and inside  $\omega$ , we use equation (2.8).

In the same way, if we assume  $v$  to be enough regular, we can consider the solution  $u_N \in H^1(\Omega \setminus \sigma)$  of the following Neumann problem:

$$\begin{cases} \Delta u_N = 0 & \text{in } \omega \setminus \sigma, \\ \partial_n u_N = \partial_n v & \text{on } \gamma, \\ \partial_n u_N = 0 & \text{on } \sigma, \\ u_N = v & \text{in } \Omega \setminus \omega. \end{cases} \quad (2.12)$$

Note that from the numerical point of view, it is much more easy to solve an approximated Neumann problem:

$$\begin{cases} \Delta u_N = 0 & \text{in } \omega \setminus \sigma, \\ \partial_n u_N = \partial_n v & \text{on } \gamma, \\ \partial_n u_N = 0 & \text{on } \sigma, \\ -\alpha \Delta u_N + u_N = v & \text{in } \Omega \setminus \omega, \end{cases} \quad (2.13)$$

where  $\alpha$  is a small positive number.

### 2.3.3 Asymptotic expansion

The cost function remains unchanged, and is still defined by (2.10), as the idea is to find some cracks  $\sigma \subset \omega$  that minimize the difference between the two solutions  $u_N$  and  $u_D$ . We assume that the crack  $\sigma$  is equal to  $x + \rho \bar{\sigma}$ , where  $x$  is the point of insertion of the crack,  $\rho$  is the size of the inserted crack (assumed to be small), and  $\bar{\sigma}$  is a reference crack, of unit normal vector  $n$ . Then, we can rewrite the cost function  $J$  defined by equation (2.10) as a function  $j(\rho)$  of  $\rho$ . The asymptotic expansion is then the following:

$$j(\rho) - j(0) = f(\rho)g(x, n) + o(f(\rho)), \quad (2.14)$$

where the topological gradient  $g$  is defined by

$$g(x, n) = -[(\nabla u_D(x) \cdot n)(\nabla p_D(x) \cdot n) + (\nabla u_N(x) \cdot n)(\nabla p_N(x) \cdot n)], \quad (2.15)$$

where  $u_D$  and  $u_N$  are the solutions of (2.11) and (2.12) respectively, but without any inserted crack ( $\sigma = \emptyset$ ). Also,  $p_D$  and  $p_N$  are the corresponding adjoint states, respectively solutions in  $H^1(\Omega)$  of the following equations:

$$\begin{cases} p_D = 0 & \text{in } \Omega \setminus \omega, \\ p_D = 0 & \text{on } \gamma, \\ -\Delta p_D = -(u_D - u_N) & \text{in } \omega, \end{cases} \quad (2.16)$$

$$\begin{cases} p_N = 0 & \text{in } \Omega \setminus \omega, \\ \partial_n p_N = 0 & \text{on } \gamma, \\ -\Delta p_N = +(u_D - u_N) & \text{in } \omega. \end{cases} \quad (2.17)$$

The topological gradient defined by equation (2.15) can be rewritten in the following way:

$$g(x, n) = n^T M(x) n, \quad (2.18)$$

where  $M(x)$  is the  $2 \times 2$  (resp.  $3 \times 3$  in the case of 3D images, or movies) symmetric matrix defined by

$$M(x) = -\text{sym}(\nabla u_D(x) \otimes \nabla p_D(x) + \nabla u_N(x) \otimes \nabla p_N(x)). \quad (2.19)$$

From this equation, we can deduce that the minimum of  $g(x, n)$  is reached when  $n$  is the eigenvector associated to the lowest eigenvalue  $\lambda_{\min}(M(x))$  of  $M(x)$ .

### 2.3.4 Algorithm

The inpainting algorithm is then the following:

- Calculation of  $u_D$  and  $u_N$ , solutions of the direct problems (2.11) and (2.12) respectively, without any inserted crack (unperturbed problem:  $\sigma = \emptyset$ ).
- Calculation of  $p_D$  and  $p_N$  the two corresponding adjoint states, respectively solutions of equations (2.16) and (2.17).
- Computation of the matrix  $M(x)$  defined by equation (2.19).
- Localization of the cracks: define

$$\sigma = \{x \in \omega; \lambda_{\min}(M(x)) < \delta < 0\}, \quad (2.20)$$

where  $\delta$  is a negative threshold.

- Calculation of the solution of the Neumann problem (2.12) perturbed by the insertion of  $\sigma$ .

This image is then equal to the original image in  $\Omega \setminus \omega$ , and it has been reconstructed in  $\omega$ .



### 2.3.5 Remarks

From the numerical point of view, cracks are modeled by a very small conductivity instead of considering real holes in the domain. The previous algorithm has a complexity of  $\mathcal{O}(n \cdot \log(n))$ , where  $n$  is the size of the image, i.e. the number of pixels, as explained in section 2.7.

The main advantage of this algorithm is that the reconstruction is done in only one iteration of the topological gradient algorithm, which consists of 5 resolutions of a PDE (the two direct and two adjoint unperturbed problems, and then one direct perturbed problem) in the domain  $\Omega$  representing the image. Several numerical results are presented in [9] and show the quality and efficiency of the reconstruction.

The only control parameter of this method is the negative threshold: below a given value, the pixels are considered as being part of the edge set, whereas it is not the case beyond the threshold. The reconstructed image is provided by the resolution of the direct perturbed problem (2.12), and the quality of the image relies on the connectivity of the identified edges. If a given identified edge is not connex, the Laplacian indeed produces a blurred zone. Then, the threshold is usually set such that the main identified edges are connex. Of course, it may lead to the wrong identification of edges. But the various numerical experiments have shown that the threshold can be fixed to an a priori value, as the optimal threshold is almost independent of the images.

Another solution to this problem is presented in section 2.8.

## 2.4 Restoration [16, 22]

In this section, we consider the restoration problem, with the aim of restoring noisy images. The main idea is to use the topological gradient for detecting the edges of the noisy image in order to preserve them during the restoration process.

This method is based on thermal diffusion, like many other variational methods. In order to avoid blurring effects, several nonlinear isotropic and anisotropic methods have been introduced, some of them relying on the minimization of the total variation [142, 66, 124, 174, 175, 43]. We should mention that some non variational approaches also exist, mainly statistical methods [86].

This section summarizes the work presented in [16, 22]. We also refer to these references for the results of numerical experiments.

### 2.4.1 Variational formulation

Let  $\Omega \subset \mathbb{R}^2$  be an open bounded domain, and  $v \in L^2(\Omega)$  be the noisy image. The enhancement of  $v$  is based on the resolution of the following problem:

$$\text{find } u \in H^1(\Omega) \text{ such that } \begin{cases} -\text{div}(c\nabla u) + u = v & \text{in } \Omega, \\ \partial_n u = 0 & \text{on } \partial\Omega, \end{cases} \quad (2.21)$$

where  $n$  is the outward unit normal to  $\partial\Omega$ , and  $c$  is the conductivity, to be defined in the following. Several choices can be made for the conductivity, mainly  $c$

equal to a constant value (linear diffusion method: it is fast, but it blurs important structures), or  $c$  defined by a nonlinear function of  $\nabla u$  (nonlinear diffusion method, edge-preserving [175, 43]). In the topological gradient approach,  $c$  takes only two values: a constant value  $c_0$  (close to 1) in the smooth part of the image, and a very small values  $\varepsilon$  (close to 0) on the edges or cracks in order to preserve them.

Setting  $c = 0$  on a part of the image is equivalent to perturbing the domain by the insertion of cracks. For a given point  $x_0 \in \Omega$  and for a given small parameter  $\rho > 0$ , we consider  $\Omega_\rho = \Omega \setminus \sigma_\rho$  the perturbed domain by the insertion of a crack  $\sigma_\rho = x_0 + \rho\sigma(n)$ , where  $\sigma(n)$  is a straight crack and  $n$  is a unit vector normal to the crack. The variational formulation of the perturbed problem is the following:

$$\text{find } u_\rho \in H^1(\Omega_\rho) \text{ such that } a_\rho(u_\rho, w) = l_\rho(w), \quad \forall w \in H^1(\Omega_\rho), \quad (2.22)$$

where  $a_\rho$  (resp.  $l_\rho$ ) is the following bilinear (resp. linear) form defined on  $H^1(\Omega_\rho)$  (resp.  $L^2(\Omega_\rho)$ ) by

$$a_\rho(u, w) = \int_{\Omega_\rho} (c\nabla u \nabla w + uw) dx, \quad l_\rho(w) = \int_{\Omega_\rho} vw dx. \quad (2.23)$$

Edge detection is equivalent to looking for a subdomain of  $\Omega$  where the energy is small. So our goal is to minimize the energy norm outside edges:

$$j(\rho) = J(\Omega_\rho, u_\rho) = \int_{\Omega_\rho} \|\nabla u_\rho\|^2. \quad (2.24)$$

### 2.4.2 Topological gradient

From theorem 2.1, we can derive the following asymptotic expansion of the cost function (2.24):

$$j(\rho) - j(0) = \rho^2 G(x_0, n) + o(\rho^2), \quad (2.25)$$

where

$$G(x_0, n) = -\pi c (\nabla u_0(x_0) \cdot n) (\nabla p_0(x_0) \cdot n) - \pi |\nabla u_0(x_0) \cdot n|^2, \quad (2.26)$$

where  $p_0$  is the solution of the unperturbed adjoint problem:

$$\begin{cases} -\text{div}(c\nabla p_0) + p_0 = -\partial_u J(\Omega, u_0) & \text{in } \Omega, \\ \partial_n p_0 = 0 & \text{on } \partial\Omega. \end{cases} \quad (2.27)$$

As previously seen, the topological gradient can be rewritten:  $G(x, n) = \langle M(x)n, n \rangle$ , where  $M(x)$  is the following  $2 \times 2$  symmetric matrix:

$$M(x) = -\pi c \frac{\nabla u_0(x) \nabla p_0(x)^T + \nabla p_0(x) \nabla u_0(x)^T}{2} - \pi \nabla u_0(x) \nabla u_0(x)^T. \quad (2.28)$$

### 2.4.3 Algorithm

Our algorithm consists of inserting small heterogeneities (or cracks) in regions where the topological gradient is smaller than a given threshold. These regions are the edges of the image. The algorithm is as follows:

- Initialization:  $c = c_0$  (constant value everywhere).
- Calculation of  $u_0$  and  $p_0$ , respectively solutions of the direct (2.21) and adjoint (2.27) unperturbed problems.
- Computation of the  $2 \times 2$  matrix  $M(x)$  defined by (2.28), and of its lowest eigenvalue  $\lambda_{min}(M(x))$  at each point of the domain.
- Set the new conductivity:

$$c_1 = \begin{cases} \varepsilon & \text{if } x \in \Omega \text{ is such that } \lambda_{min}(M(x)) < \alpha < 0, \\ c_0 & \text{elsewhere,} \end{cases} \quad (2.29)$$

where  $\varepsilon > 0$  is assumed to be small, and  $\alpha$  is a negative threshold.

- Calculation of  $u_1$ , the solution of the perturbed direct problem (2.21) using  $c = c_1$ .

The image  $u_1$  is the restored image.

### 2.4.4 Remarks

From the numerical point of view, it is more convenient to simulate the cracks by a small value of  $c$  instead of considering topological perturbations of  $\Omega$ . The resolution of problem (2.21) with  $c = c_1$  is an approximation of the resolution of the perturbed problem (2.22), becoming more precise as  $\varepsilon$  goes to 0.

As in the previous section (inpainting problems), our algorithm is extremely efficient as it requires only 3 resolutions of a partial differential equation in  $\Omega$ : the direct and adjoint original problems, and then the direct perturbed problem. And the complexity of this algorithm is still  $\mathcal{O}(n \cdot \log(n))$  (see section 2.7).

As shown in [16], the quality of the numerical results is very good. Once again, the algorithm relies on a thresholding of the topological gradient in order to define the edge set. Contrary to inpainting problems, the connexity of the edges is not crucial since it does not change significantly the quality of the restored image. However, section 2.8 presents a way to identify connex edges, with fewer badly identified edges.

### 2.4.5 Extension to color images

In this section, we adapt the topological gradient approach to color images. Color images can be represented or modeled in various ways, for instance the RGB (Red-Green-Blue) space in which images are viewed as functions from  $\Omega$  to  $\mathbb{R}^3$  instead of  $\mathbb{R}$ . A first approach consists of decoupling the three channels, and in solving direct

and adjoint problems for each channel. But it is also possible to consider directly the vectorial minimization problem, involving the resolution of vectorial problems. The topological asymptotic expansion is still given by equations (2.25-2.26) and (2.28), where all functions are vectorial, i.e. the topological gradient is the sum on all channels of the corresponding expressions for each channel [22].

Another approach has also been studied in [22], in which we use a different norm for coupling the different channels. In order to identify the local variations of the color image, Di Zenzo defines a multi-spectral tensor associated to the image vector field [83]:

$$T = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}, \quad t_{ij} = \sum_{k=1}^3 \frac{\partial u^k}{\partial x_i} \frac{\partial u^k}{\partial x_j}, \quad 1 \leq i, j \leq 2, \quad (2.30)$$

in the case of bidimensional images. This tensor describes the first order differential structure of the image, and the Di Zenzo gradient is given by the square root of the largest eigenvalue of the structure tensor:

$$\|\nabla u\|_{DZ} = \frac{1}{\sqrt{2}} \left[ t_{11} + t_{22} + \sqrt{(t_{11} - t_{22})^2 + 4t_{12}^2} \right]^{\frac{1}{2}}. \quad (2.31)$$

It is possible to rewrite this gradient in a different way with the following function:

$$H(\nabla u) = \frac{1 + \sqrt{1 - 4f(\nabla u)}}{2}, \quad (2.32)$$

where

$$f(\nabla u) = \frac{\det^2(\nabla u^1, \nabla u^2) + \det^2(\nabla u^1, \nabla u^3) + \det^2(\nabla u^2, \nabla u^3)}{(|\nabla u^1|^2 + |\nabla u^2|^2 + |\nabla u^3|^2)^2}, \quad (2.33)$$

$$\det^2(\nabla u^s, \nabla u^t) = \left( \frac{\partial u^s}{\partial x_1} \frac{\partial u^t}{\partial x_2} - \frac{\partial u^t}{\partial x_1} \frac{\partial u^s}{\partial x_2} \right)^2. \quad (2.34)$$

Then, we can derive the asymptotic expansion of the cost function defined by equation (2.24) in which the norm is the Di Zenzo norm (2.31):

$$G(x_0, n) = \sum_{k=1}^3 \left[ -\pi c(\nabla u_0^k(x_0) \cdot n)(\nabla v_0^k(x_0) \cdot n) - \pi H(\nabla u_0(x_0)) |\nabla u_0^k(x_0) \cdot n|^2 \right] \quad (2.35)$$

with our standard notations.

In [22], we show that this approach has the same computational cost as the vectorial approach (in which the different channels are decoupled), while it improves the edge detection, and hence it produces a better restored image, more precise on the edges of the image.

## 2.5 Classification [10, 16]

In this section, we now focus on the regularized and unsupervised image classification problem.

Inspired by the work presented in [150, 43], in which the authors propose a classification model coupled with a restoration process, we adapt here our approach based on the topological asymptotic analysis.

This section summarizes the work presented in [16, 10]. We refer to these references for the numerical results.

### 2.5.1 Introduction to the classification problem

Let  $v$  be the original image defined on an open set  $\Omega$  of  $\mathbb{R}^2$ , and let  $C_i, 1 \leq i \leq n$ , be  $n$  classes (i.e. grey or color levels). We first assume that these classes are predefined. The goal of image classification is to find a partition of  $\Omega$  in subsets  $\{\Omega_i\}_{i=1\dots n}$ , such that  $v$  is close to  $C_i$  in  $\Omega_i$ .

A variational approach can be defined: it consists of a cost function measuring the difference between the original image and the classified image:

$$J((\Omega_i)_{i=1\dots n}) = \sum_{i=1}^n \int_{\Omega_i} (v(x) - C_i)^2 dx + \alpha \sum_{i \neq j} |\Gamma_{ij}|, \quad (2.36)$$

where  $\Gamma_{ij}$  represents the interface  $\Omega_i \cap \Omega_j$  between two subsets.

The main difficulty of this approach is that the unknowns are sets, and not variables. This is why the topological asymptotic analysis seems to be appropriate for solving this problem. The topological gradient and the corresponding numerical results are presented in [10].

### 2.5.2 Restoration and classification coupling

Another solution consists of coupling classification with restoration, and to adapt the approach introduced in section 2.4. The idea is to first consider an iteration of the topological asymptotic analysis for the image restoration problem in order to smooth the image, and then to classify this smooth image without any regularization. If we remove the regularization term from equation (2.36), which leads to the unregularized classification problem, then the optimal subset  $\Omega_i$  is the set of pixels that are closer to  $C_i$  than to any other  $C_j$ . In other words, each pixel is assigned to the subset corresponding to its closest class.

In the perturbed problem (2.29), instead of setting  $c = 0$  (or  $c = \varepsilon$  from the numerical point of view) on the edge set and  $c = c_0$  elsewhere, we set

$$c_1 = \begin{cases} \varepsilon & \text{on the edge set,} \\ \frac{c_0}{\varepsilon} & \text{elsewhere.} \end{cases} \quad (2.37)$$

The algorithm is then the following:

- Application of the restoration algorithm defined in section 2.4, with  $c_1$  defined by (2.37) instead of (2.29).
- Unregularized classification of the image  $u_1$ , using for example the closest class algorithm (in which each pixel is assigned to the subset corresponding to its closest class).

As previously seen, the complexity of this algorithm is  $\mathcal{O}(n \cdot \log(n))$ , and the various numerical results presented in [10] show the relative efficiency of these approaches. Moreover, it is possible to regularize more or less the image by choosing different values of  $c_1$ , and it allows us to also obtain good results on noisy images.

### 2.5.3 Extension to unsupervised classification

If the number  $n$  of classes is given, but not their values  $C_i$ , it is possible to determine them in an optimal way. This classification problem can be defined as

$$\min_{(\Omega_i), (C_i)} j((\Omega_i), (C_i)) = \sum_{i=1}^n \int_{\Omega_i} |v(x) - C_i|^2 dx + \alpha \sum_{i \neq j} |\Gamma_{ij}|. \quad (2.38)$$

The idea is to minimize the cost function  $j((\Omega_i), (C_i))$  alternatively with respect to  $\Omega_i$  and with respect to  $C_i$ . The minimization with respect to  $\Omega_i$  consists of classifying the image, while the minimization with respect to  $C_i$  is obtained straightforwardly by the mean of the image in each class:

$$C_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} v(x) dx. \quad (2.39)$$

The unsupervised classification algorithm is then as follows:

- Initialization: define an initial guess  $C_1, \dots, C_n$  (e.g. equi-distributed classes).
- Repeat until convergence:
  - Calculate the classified image using the classes  $C_1, \dots, C_n$  (see previous algorithm).
  - Update the values of the classes using (2.39).

If the number  $n$  of classes is not given, we can add a penalization term “ $+\beta n$ ” in the cost function (2.38), measuring the number of classes. The minimization with respect to  $n$  provides the optimal number of classes. The number of classes is clearly related to the choice of the weighting coefficient  $\beta$ .

## 2.6 Segmentation [12, 13]

This section is concerned with image segmentation, which aim is to find a partition of an image into its constituent parts. The idea is still to apply our topological gradient based algorithm for the detection of edges in the image.

Several approaches have been studied in the literature. One can cite variational methods, for example based on the minimization of the Mumford-Shah functional [136], the active contours and snake methods [55, 156], stochastic approaches [54, 61], wavelets, . . . [43, 45, 42, 140, 149, 150, 176].

This section summarizes the main results presented in [12, 13]. Several numerical experiments are also detailed in these references and show the efficiency of our approach.

### 2.6.1 From restoration to segmentation

We still consider the restoration algorithm, in which the following conductivity is used for the perturbed problem:

$$c(\varepsilon) = \begin{cases} \varepsilon & \text{in } \omega, \\ \frac{1}{\varepsilon} & \text{outside } \omega, \end{cases} \quad (2.40)$$

where  $\omega \subset \Omega$  represents the edge set. We first assume that  $\omega$  is thickened (i.e. of codimension 0 in  $\Omega$ ). From equation (2.40), the algorithm now consists of solving the following problem:

$$(\mathcal{P}_\varepsilon) \quad \begin{cases} -\operatorname{div}(\varepsilon \nabla u_\varepsilon) + u_\varepsilon = v & \text{in } \omega, \\ -\operatorname{div}\left(\frac{1}{\varepsilon} \nabla u_\varepsilon\right) + u_\varepsilon = v & \text{in } \Omega \setminus \omega, \\ \partial_n u_\varepsilon = 0 & \text{on } \partial\Omega, \end{cases} \quad (2.41)$$

where  $u_\varepsilon \in H^1(\Omega)$ , i.e. with the implicit boundary condition that  $c(\varepsilon)\partial_n u_\varepsilon$  has the same value on both sides of  $\partial\omega$ .

Then we have the following asymptotic result [12]:

**Theorem 2.2** *If we denote by  $u_\varepsilon$  the unique solution of problem  $(\mathcal{P}_\varepsilon)$  in  $H^1(\Omega)$ , then*

$$\lim_{\varepsilon \rightarrow 0} (\|\nabla u_\varepsilon - \nabla u_0\|_{L^2(\Omega \setminus \omega)} + \|u_\varepsilon - u_0\|_{L^2(\omega)}) = 0, \quad (2.42)$$

where  $u_0 \in H^1(\Omega \setminus \omega) \cap L^2(\Omega)$  is the solution to the following problem

$$(\mathcal{P}_0) \quad \begin{cases} u_0 = v & \text{in } \omega, \\ -\operatorname{div}(\nabla u_0) = 0 & \text{in } \Omega \setminus \omega, \\ \partial_n u_0 = 0 & \text{on } \partial\omega, \\ \partial_n u_0 = 0 & \text{on } \partial\Omega. \end{cases} \quad (2.43)$$

This result proves that the segmented image  $u_0$  can be approximated by  $u_\varepsilon$  if  $\varepsilon$  is small. We now assume that the edge set  $\omega$  is of codimension 1 in  $\Omega$ . From the point of view of applications, it is completely natural to assume that the edges are flat in the image. In order to have coherent notations, we will further denote by  $\sigma$  the edge set. We assume that  $\sigma$  is known, e.g. provided by the crack detection algorithm previously seen.

We can rewrite the approximated segmentation problem  $(\mathcal{P}_\varepsilon)$  as follows:

$$(\tilde{\mathcal{P}}_\varepsilon) \quad \begin{cases} -\operatorname{div} \left( \frac{1}{\varepsilon} \nabla u_\varepsilon \right) + u_\varepsilon = v & \text{in } \Omega \setminus \sigma, \\ \partial_n u_\varepsilon = 0 & \text{on } \sigma, \\ \partial_n u_\varepsilon = 0 & \text{on } \partial\Omega, \end{cases} \quad (2.44)$$

where  $u_\varepsilon \in H^1(\Omega \setminus \sigma)$ . If  $v \in L^2(\Omega)$ , then problem  $(\tilde{\mathcal{P}}_\varepsilon)$  has a unique solution in  $H^1(\Omega \setminus \sigma)$ . As a corollary of the previous result, we have the following one [12]:

**Theorem 2.3** *If we denote by  $u_\varepsilon$  the unique solution of problem  $(\tilde{\mathcal{P}}_\varepsilon)$  in  $H^1(\Omega \setminus \sigma)$ , then*

$$\|u_\varepsilon\|_{L^2(\Omega)} \leq \|v\|_{L^2(\Omega)}, \quad \|\nabla u_\varepsilon\|_{L^2(\Omega \setminus \sigma)} \leq \sqrt{\varepsilon} \|v\|_{L^2(\Omega)}, \quad (2.45)$$

and

$$\lim_{\varepsilon \rightarrow 0} \|\nabla u_\varepsilon - \nabla u_0\|_{L^2(\Omega \setminus \sigma)} = 0, \quad (2.46)$$

where  $u_0 \in H^1(\Omega \setminus \sigma)$  is the unique solution to the following problem:

$$(\tilde{\mathcal{P}}_0) \quad \begin{cases} -\operatorname{div}(\nabla u_0) = 0 & \text{in } \Omega \setminus \sigma, \\ \int_{\Omega_i} u_0 = \int_{\Omega_i} v & \forall \Omega_i \text{ connex component of } \Omega \setminus \sigma, \\ \partial_n u_0 = 0 & \text{on } \sigma, \\ \partial_n u_0 = 0 & \text{on } \partial\Omega. \end{cases} \quad (2.47)$$

For numerical reasons, it can be very difficult to solve directly problem  $(\tilde{\mathcal{P}}_0)$ , and even problem  $(\tilde{\mathcal{P}}_\varepsilon)$  for too small values of  $\varepsilon > 0$ . Indeed the conditioning of the system to be solved goes to infinity when  $\varepsilon \rightarrow 0$ . In order to overcome this issue, we will expand the solution  $u_\varepsilon$  of problem  $(\tilde{\mathcal{P}}_\varepsilon)$  into a power series of  $\varepsilon$ .

### 2.6.2 Power series expansion

From the knowledge of the power series expansion of  $u_\varepsilon$  and the computation of several solutions  $u_\varepsilon$  for not too small coefficients  $\varepsilon > 0$ , it is possible to approximate the asymptotic solution  $u_0$  [12]:

**Theorem 2.4** *There exist a constant  $\varepsilon_R > 0$  and a family of functions  $(u_n)_{n \in \mathbb{N}}$  of  $H^1(\Omega \setminus \sigma)$  such that for all  $0 \leq \varepsilon \leq \varepsilon_R$ ,*

$$u_\varepsilon = \sum_{n=0}^{\infty} u_n \varepsilon^n. \quad (2.48)$$



Moreover,  $u_0$  is the unique solution in  $H^1(\Omega \setminus \sigma)$  of problem  $(\tilde{\mathcal{P}}_0)$ , and the other functions  $(u_n)$  are the unique solutions in  $H^1(\Omega \setminus \sigma)$  of the following problems:

$$(\tilde{\mathcal{P}}_1) \begin{cases} -\operatorname{div}(\nabla u_1) = -u_0 + v & \text{in } \Omega \setminus \sigma, \\ \int_{\Omega_i} u_1 = 0 & \forall \Omega_i \text{ connex component of } \Omega \setminus \sigma, \\ \partial_n u_1 = 0 & \text{on } \sigma, \\ \partial_n u_1 = 0 & \text{on } \partial\Omega, \end{cases} \quad (2.49)$$

$$n \geq 2, (\tilde{\mathcal{P}}_n) \begin{cases} -\operatorname{div}(\nabla u_n) = -u_{n-1} & \text{in } \Omega \setminus \sigma, \\ \int_{\Omega_i} u_n = 0 & \forall \Omega_i \text{ connex component of } \Omega \setminus \sigma, \\ \partial_n u_n = 0 & \text{on } \sigma, \\ \partial_n u_n = 0 & \text{on } \partial\Omega. \end{cases} \quad (2.50)$$

We can define a function of  $\varepsilon \in \mathbb{R}^+$  as follows

$$f(\varepsilon) := u_\varepsilon \in H^1(\Omega \setminus \sigma). \quad (2.51)$$

From the previous theorem, we know that  $f$  has a power series expansion at the origin given by (2.48). We consider a family of  $N$  points  $(\varepsilon_i)$  in  $[\varepsilon_c, \varepsilon_R]$ , where  $\varepsilon_c$  is the smallest value of  $\varepsilon$  for which it is easy to numerically compute  $f(\varepsilon)$ , and  $\varepsilon_R$  is smaller than the convergence radius of the power series. We can then compute an interpolation polynomial  $g_N$  of degree  $N - 1$  defined by:

$$g_N(\varepsilon) = \sum_{i=1}^N \left( \prod_{j=1, j \neq i}^N \frac{\varepsilon - \varepsilon_j}{\varepsilon_i - \varepsilon_j} \right) u_{\varepsilon_i}, \quad (2.52)$$

where  $N$  is the number of points  $\varepsilon_i$ .

The analyticity of  $f$  allows us to estimate the approximation error:

$$\|u_0 - g_N(0)\|_{H^1(\Omega \setminus \sigma)} = \mathcal{O}(\varepsilon_c^N). \quad (2.53)$$

### 2.6.3 Algorithm

We can then define a segmentation algorithm, based on the restoration algorithm previously defined in section 2.4:

- Solve the direct (2.21) and adjoint (2.27) unperturbed problems with  $c = c_0$  everywhere.
- Compute the  $2 \times 2$  matrix  $M(x)$  defined by equation (2.28) and its lowest eigenvalue  $\lambda_{\min}(M(x))$  at each point of the domain  $\Omega$ .
- Define  $\sigma = \{x \in \Omega; \lambda_{\min} < \alpha < 0\}$  the edge set, where  $\alpha$  is a small negative threshold.

- Set  $\varepsilon_c > 0$  the minimal value of  $\varepsilon$  for which it is easy to compute numerically the solution  $u_\varepsilon$  of problem  $(\tilde{\mathcal{P}}_\varepsilon)$ .
- Choose  $N \in \mathbb{N}^*$  in order to have an approximation error in  $\mathcal{O}(\varepsilon_c^N)$ , and choose  $N$  different values  $(\varepsilon_i)$ .
- Compute the solutions  $(u_{\varepsilon_i})$  in  $H^1(\Omega \setminus \sigma)$  of problems  $(\tilde{\mathcal{P}}_{\varepsilon_i})$ .
- Compute the interpolation polynomial  $g_N$  of degree  $N - 1$ , defined by equation (2.52), for  $\varepsilon = 0$ .

This algorithm has a complexity in  $\mathcal{O}(N.n.\log(n))$ , where  $n$  is the number of pixels in the image, and  $N$  is the degree of the interpolation approximation. In numerical experiments,  $N$  is typically of the order of 2 to 5.

Several numerical tests are detailed in [12].

## 2.7 Complexity and speeding up [13, 16]

In this section, we present the techniques that we have used for solving the PDE problems previously seen, and that lead to a theoretical complexity in  $\mathcal{O}(n.\log(n))$  [13]. Several numerical experiments have confirmed this complexity [16, 13].

### 2.7.1 Discrete cosine transform

In all the algorithms we presented in the previous sections, we only have to solve the following PDE

$$\begin{cases} -\operatorname{div}(c\nabla u) + u = v & \text{in } \Omega, \\ \partial_n u = 0 & \text{on } \partial\Omega, \end{cases} \quad (2.54)$$

for various coefficients  $c$ . The first resolutions are done with a constant value of  $c$ . It is then possible to largely speed up the computation time by using the discrete cosine transform (DCT) method. Problem (2.54) is then equivalent to

$$\sum_{m,n} (1 + c(m\pi)^2 + c(n\pi)^2) u_{m,n} \phi_{m,n} = \sum_{m,n} v_{m,n} \phi_{m,n}, \quad (2.55)$$

where we denote by  $\phi_{m,n} = \delta_{m,n} \cos(m\pi x) \cos(n\pi y)$  a cosine basis of  $\mathbb{R}^2$ , and where  $(v_{m,n})$  represent the DCT coefficients of the original image  $v$ . It is then straightforward to identify  $(u_{m,n})$ , the DCT coefficients of  $u$  in equation (2.55):

$$u_{m,n} = \frac{v_{m,n}}{1 + c(m\pi)^2 + c(n\pi)^2}. \quad (2.56)$$

The complexity of such a resolution is  $\mathcal{O}(n.\log(n))$ , where  $n$  is the number of pixels of the image. The resolution of all unperturbed problems is then done in the following way:

- Computation of  $v_{m,n}$ , the DCT coefficients of the original image  $v$ .

- Computation of  $u_{m,n}$ , the DCT coefficients of  $u$  from equation (2.56).
- Computation of  $u$  using an inverse DCT.

### 2.7.2 Preconditioned conjugate gradient

Then, the solution of all previously detailed problems comes from the resolution of a perturbed problem. For the last resolution of a direct problem with a non constant coefficient  $c$ , we can rewrite the problem in the following way:

$$A(c)u = B, \quad (2.57)$$

where  $u$  is the unknown image. If  $c$  is constant, equation (2.57) is easy to solve. The idea is to precondition equation (2.57) with the DCT solver used in the first resolution. Problem (2.57) is equivalent to

$$[A(c_0)^{-1}A(c)]u = [A(c_0)^{-1}B]. \quad (2.58)$$

As  $c$  is close to  $c_0$  ( $c$  is indeed equal to  $c_0$ , except in a negligible part of the domain), the system matrix  $[A(c_0)^{-1}A(c)]$  is close to the identity operator, and the resolution of (2.58) is then easy: we use a preconditioned conjugate gradient (PCG) method to solve this problem. As the coefficient  $c$  is close to  $c_0$ , we can expect a  $\mathcal{O}(n \cdot \log(n))$  complexity for the resolution of the perturbed problem. The numerical experiments clearly confirm this complexity, both for small and large problems.

The main advantage is that it allows us to process images in a very short time (e.g.  $1600 \times 1200$  images in less than one second) and movies in real time (provided the movie is split into short sequences of a few seconds) with a c++ code.

## 2.8 Coupling between the topological gradient and the minimal path technique [26]

As previously seen, e.g. in section 2.3, it is crucial to identify connected (or continuous) contours. Up to now, we had to threshold the topological gradient with a not too small value, in order to identify connected contours, but this leads to thick identified edges, and also to consider more noisy points as potential edges.

We noticed that the edges correspond to valley lines of the topological gradient. It is of course possible to identify them by adapting the threshold coefficient, but we propose here to use the minimal path and fast marching techniques for identifying the valley lines of the topological gradient [71, 73, 82, 84, 179, 145, 164].

In the following, we consider any of the previous image processing problems. We only assume that the topological gradient  $g$  has been defined and computed everywhere. The goal is to identify the valley lines corresponding to the most negative parts of the topological gradient.

This section summarizes the study presented in [26], in which several numerical experiments are shown in the case of segmentation and inpainting.

### 2.8.1 Minimal paths

Let  $g$  be the topological gradient. The idea of the minimal path technique is to define a potential function, measuring in some sense for any point of  $\Omega$  the cost for a path to contain this point. As we want to identify paths in the most negative part of the topological gradient, and considering that the potential function must be positive, we define the following function:

$$P(x) = g(x) - \min_{y \in \Omega} \{g(y)\}. \quad (2.59)$$

We simply shift the topological gradient from its minimal value, in order to obtain a positive function. We can see that the points where the topological gradient  $g$  reaches its minimal value are costless. This is a way to consider that these points must be on the minimal paths.

We denote by  $C(s)$  a path (or curve) in the image, where  $s$  represents the curvilinear coordinate. We can now define a cost function, measuring the cost of such a path:

$$J(C) = \int_C (P(C(s)) + \alpha) ds, \quad (2.60)$$

where  $\alpha > 0$  is a positive regularization coefficient, measuring the length of this path.

The goal is to minimize  $J$ , in order to find the shortest and least costly path between two points. For this purpose, we define the following distance function:

$$D(x; x_0) = \inf_{C \in \mathcal{A}(x, x_0)} J(C), \quad (2.61)$$

where  $\mathcal{A}(x, x_0)$  is the set of all paths going from  $x_0$  to  $x$  in the image.

### 2.8.2 Fast marching

The fastest way to compute the distance function defined by equation (2.61) is to solve a front propagation equation:

$$\frac{\partial C(s, t)}{\partial t} = \frac{1}{P(C(s, t)) + \alpha} \mathbf{n}_C(s, t), \quad (2.62)$$

where  $\mathbf{n}_C(s, t)$  is the outer normal unit vector to the front  $C$ . We initialize the propagation with  $C(s, 0)$  equal to a infinitely small circle centered at  $x_0$ .

This path evolves with a propagation speed inversely proportional to the potential function. If for example a point in the outer part of the front has a large potential (i.e. a large cost), then the propagation speed will be nearly equal to 0 and the front will not expand at this point. From the theory of Eikonal equations, the distance  $D(x; x_0)$  is simply the instant  $t$  at which the front, initialized at point  $x_0$ , reaches point  $x$  [179, 84].

### 2.8.3 Coupled algorithm

We can certainly consider that the global minimum of the topological gradient is part of the edge set. So we can choose the reference point  $x_0$  as being this minimum. It is then possible to compute the distance between  $x_0$  and any other point  $x$ . A gradient descent algorithm can then be used to minimize the distance function, in order to find the minimal path between  $x_0$  and the initial point of the optimization scheme. If these two points are part of the same edge, as the potential function has been defined such that it is relatively costless to remain on the edge (thanks to the topological gradient), the minimal path will be a very good approximation of this edge. The main advantage is now that we are sure that this path corresponds to a continuous contour.

For a small additional computation cost, it is possible to consider more than one reference point. The distance function corresponds then to the distance to the set of these points. The corresponding Voronoï diagram can be seen as a dual mesh, and the minimum of the distance function on each edge of this mesh is a saddle-point: minimal distance along the edges of the mesh, maximal distance to the reference points.

The hybrid algorithm we propose is the following:

- Compute the topological gradient of the image (see previous sections).
- Choose  $N$  key-points; the main one will be for example the global minimum of the topological gradient.
- Fast marching: computation of the distance function to all these key-points, and of the corresponding Voronoï diagram.
- Saddle-points: on each edge of the Voronoï diagram, determine the point of minimal distance.
- Sort all these saddle-points, from smaller to larger distance.
- For each of these points, from smaller to larger distance, check if it will not be used for connecting two key-points, one of which is already connected to two other key-points.
- If this is not the case, use this point as an initialization for a descent type algorithm in order to connect the two corresponding key-points.

This algorithm clearly converges, and all the key-points are connected to at most two other key-points at convergence. This provides a continuous contour, connecting the key-points. It is then an approximation of one of the main contours of the image as it corresponds to a valley line of the topological gradient.

As seen in [26], it allows us to appreciably improve our inpainting algorithm. It also improves the quality of the segmentation. For all other image processing problems, there were no noticeable improvements.

## 2.9 Conclusions and perspectives

We presented in this chapter many applications in image processing of the crack detection technique, based on the topological gradient. It provides an excellent frame for solving all these image processing problems. It has been successfully applied to image inpainting, restoration, classification and segmentation. In all these cases, we obtain excellent results and the computing time is very small.

We have also seen that this technique can be applied to color images as well as grey-level images, but also three-dimensional images, or movies, without any trouble. The theoretical complexity, confirmed by numerical experiments, allows us to process movies in nearly real time (on a dual-processor laptop, with a `c++` code).

Another interesting point is that all these algorithms rely on the same kernel, as we always solve the same kind of PDE problems. This makes the implementation much more easy.

Several perspectives are currently under study. We can cite here the possibility of taking into account higher order differential operators, with the aim of a better reconstruction of the gradient of the image. For instance, in the case of inpainting problems, the inpainted image is piecewise affine. With the same kind of approach, we should be able to reconstruct more precisely the gradient of the image, and then the image itself.

## Chapter 3

# Data assimilation: the Back and Forth Nudging (BFN) algorithm

This chapter summarizes the work presented in [8, 11, 14, 18, 21, 24, 25].

### 3.1 Introduction

The aim of data assimilation is to combine the observations and models, in order to retrieve a coherent and precise state of the system from a set of discrete spacetime data, and then to provide reliable forecasts of its evolution. Data assimilation covers all the mathematical and numerical techniques in which the observed information is accumulated into the model state by taking advantage of consistency constraints with laws of time evolution and physical properties, and which allow us to blend as optimally as possible all the sources of information coming from theory, models and other types of data [116, 53, 163].

Nudging is a data assimilation method that uses dynamical relaxation to adjust a model towards observations. The standard nudging algorithm consists of adding to the state equations of a dynamical system a feedback term proportional to the difference between the observation and the equivalent quantity computed by integration of the state equations. The model then appears as a weak constraint, and the nudging term forces the state variables to fit as well as possible to the observations. This forcing term in the model dynamics has a tunable coefficient that represents the relaxation time scale. This coefficient is chosen by numerical experimentation so as to keep the nudging terms small in comparison to the state equations, and large enough to force the model to the observations. The nudging term can also be seen as a penalty term, which penalizes the system if the model is too far from the observations. Note that in the linear case, the standard nudging method is nothing else than the Luenberger observer, also called asymptotic observer [129].

The nudging method is a flexible assimilation technique, and computationally much more economical than variational data assimilation methods [123]. First used in meteorology [106], the nudging method has been successfully introduced in oceanog-

raphy in a quasi-geostrophic model [168, 170, 57] and has been applied to a mesoscale model of the atmosphere with synoptic-scale data [160]. The nudging coefficients can be optimized by a variational method [159, 180], where a parameter estimation approach is proposed to obtain optimal nudging coefficients, in the sense that the difference between the model solution and the observations is as small as possible. A comparison between optimal nudging and Kalman filtering can be found in [172]. A drawback of this optimal nudging technique is that it requires the computation of the adjoint state of the model equations, which is not necessary in the standard nudging method.

The backward nudging algorithm consists of solving backwards in time the state equations of the model, starting from the observation of the system state at the final time of the assimilation period. A nudging term, with the opposite sign compared to the standard nudging algorithm, is added to the state equations, and the final state computed in the backward integration is in fact an approximation of the initial state of the system [1].

The Back and Forth Nudging (BFN) algorithm, introduced in [18], consists of solving first the forward nudging equation, and then the model equation backwards in time with a relaxation term (with the opposite sign in comparison with the relaxation term introduced in the forward equation). The initial condition of this backward integration is the final state obtained by the standard nudging method. After integration of this backward equation, one obtains an estimate of the initial state of the system. We then repeat these forward and backward integrations (with the relaxation terms) until convergence of the algorithm.

Such a forward-backward assimilation technique had already been introduced in [162, 161]. In that algorithm, at each observation time, the values predicted by the model for the observed parameters were just replaced by the observed values. This corresponds to the particular case of our BFN algorithm where the nudging coefficients go to infinity.

The BFN algorithm can be compared to the four-dimensional variational algorithm (4D-VAR, see e.g. [123]), which also consists of a sequence of forward and backward integrations. In our algorithm it is useless to linearize the system, even for nonlinear problems, and the backward system is not the adjoint equation but the model equations, with an extra feedback term that stabilizes the numerical integration of this ill-posed backward problem.

Let us finally mention another back and forth data assimilation method, called the quasi-inverse method [116]. In that method, there are no nudging terms, and in the backward integration, the sign of the dissipation terms is changed for stability reasons. The idea of introducing relaxation (or nudging) terms in our algorithm enables us to keep the dissipation terms with the correct sign in the backward integration, as the nudging terms have a stabilizing role.

In this chapter, we first present the standard nudging algorithm in a general case (nonlinear model), then the nudging algorithm applied to the corresponding backward model, and finally we introduce the Back and Forth Nudging algorithm. We then present some theoretical convergence results in simplified cases (full observa-



tions) on various types of models: linear models, transport equations (both linear and nonlinear, with or without viscosity). Then, we present the numerical application of this algorithm to various physical models. Finally, nudging can be seen as a particular type of observer, and we define a specific nudging-based observer for a shallow-water model, allowing us to preserve the natural symmetries of the model, to reduce the sensitivity to the observation noise, and also to correct the non-observed variables with the observed ones. Several conclusions and perspectives are given at the end of this chapter.

## 3.2 “Back and Forth Nudging” (BFN) algorithm [8, 11, 18]

### 3.2.1 Forward nudging

In order to simplify the notations, we assume that the model equations have been discretized in space by a finite difference, finite element, or spectral discretization method. The time continuous model satisfies dynamical equations of the form:

$$\frac{dX}{dt} = F(X), \quad 0 < t < T, \quad (3.1)$$

with an initial condition  $X(0) = x_0$ . In this equation,  $F$  represents all the linear or nonlinear operators of the model equation, including the spatial differential operators.

We will denote by  $C$  the observation operator, allowing us to compare the observations  $X_{obs}(t)$  with the corresponding  $C(X(t))$ , deduced from the state vector  $X(t)$ . The observation operator usually involves interpolation/extrapolation, and some change of variables. The various measurements are not extracted at the same location as the model gridpoints, leading to some necessary interpolation and extrapolation operators. Also, satellites do not observe the physical variables of the model (e.g. temperature, velocity, ...) but some other physical parameters, that can be related to the model state: for instance, many satellites measure radiances, that can be related to the sea surface height or temperature. We do not particularly assume that  $C$  is a linear operator.

If we apply nudging to the model (3.1), we obtain

$$\frac{dX}{dt} = F(X) + K(X_{obs} - C(X)), \quad 0 < t < T, \quad (3.2)$$

with the same initial condition, and where  $K$  is the nudging (or gain) matrix. Note that it may also be a nudging scalar coefficient in some simple cases. The model then appears as a weak constraint, and the nudging term forces the state variables to fit as well as possible to the observations. In the linear case (where  $F$  is a matrix, and  $C$  is a linear operator), the forward nudging method is nothing else than the Luenberger observer [129], also called asymptotic observer, where the matrix  $K$  can be chosen so that the error goes to zero when time goes to infinity. Unfortunately, in most geophysical applications, the assimilation period is not long enough to have the nudging method give good results.

### 3.2.2 Backward nudging

We now assume that we have a final condition in equation (3.1) instead of an initial condition. This leads to the following backward equation:

$$\frac{d\tilde{X}}{dt} = F(\tilde{X}), \quad T > t > 0, \quad (3.3)$$

with a final condition  $\tilde{X}(T) = \tilde{x}_T$ . The backward nudging algorithm consists of solving backwards in time the state equations of the model, starting from the observation of the system state at the final time [1]. If we apply nudging to this backward model with a feedback term of the opposite sign (in order to have a well posed problem), we obtain

$$\frac{d\tilde{X}}{dt} = F(\tilde{X}) - K'(X_{obs} - C(\tilde{X})), \quad T > t > 0, \quad (3.4)$$

where  $K'$  is the backward nudging matrix.

The backward integration of this equation provides a state vector at time  $t = 0$ , which can be seen as an identified initial condition for our data assimilation period.

### 3.2.3 BFN algorithm

The back and forth nudging algorithm, introduced in [18], consists of first solving the forward nudging equation and then the backward nudging equation. The initial condition of the backward integration is the final state obtained after integration of the forward nudging equation. At the end of this process, one obtains an estimate of the initial state of the system. We repeat these forward and backward integrations (with the feedback terms) until convergence of the algorithm:

$$\begin{aligned} k \geq 1 & \quad \begin{cases} \frac{dX_k}{dt} = F(X_k) + K(X_{obs} - C(X_k)), \\ X_k(0) = \tilde{X}_{k-1}(0), \end{cases} \\ k \geq 1 & \quad \begin{cases} \frac{d\tilde{X}_k}{dt} = F(\tilde{X}_k) - K'(X_{obs} - C(\tilde{X}_k)), \\ \tilde{X}_k(T) = X_k(T), \end{cases} \end{aligned} \quad (3.5)$$

with the notation  $\tilde{X}_0(0) = x_0$ .

If  $K = K'$  and if the forward and backward trajectories  $X_k(t)$  and  $\tilde{X}_k(t)$  converge towards the same limit trajectory  $X_\infty(t)$ , then it is clear by adding the two equations of (3.5) that  $X_\infty(t)$  also satisfies the model equation (3.1), and that  $K(X_{obs} - C(X_\infty)) = 0$ .

When the observations are discrete in time, i.e. the observation vector  $X_{obs}$  is only available at some times  $(t_i)_{i=1\dots N}$ , then the nudging term is only added at these time steps:

$$\frac{dX}{dt} = F(X) + \sum_{i=1}^N K(X_{obs} - C(X))\delta(t - t_i), \quad 0 < t < T. \quad (3.6)$$

### 3.2.4 Choice of the nudging matrices and interpretation

#### Variational interpretation of the nudging

The standard nudging method has been widely studied in the past decades [106, 168, 160, 159]. Thus, there are several ways to choose the nudging matrix  $K$  in the forward part of the algorithm. One can for example consider the optimal nudging matrix  $K_{opt}$ , as discussed in [180, 172]. In such an approach, a variational data assimilation scheme is used in a parameter estimation mode to determine the optimal nudging coefficients. This choice theoretically provides the best results for the forward part of the BFN scheme, but the computation of the optimal gain matrix is costly.

When  $K = 0$ , the forward nudging problem (3.2) simply becomes the direct model (3.1). On the other hand, setting  $K = +\infty$  forces the state variables to be equal to the observations at discrete times, as is done in [162, 161]. These two choices have the common drawback of considering only one of the two sources of information (model and data).

Let us assume that we know the statistics of errors on observations, and denote by  $R$  the covariance matrix of observation errors. This matrix is involved in all standard data assimilation, either variational (3D-VAR, 4D-VAR, 4D-PSAS, ...) or sequential (Kalman filters) [78, 90, 144, 99, 100]. Usually, it is impossible to know the exact statistics of errors, and thus only an approximation of  $R$  is available, assumed to be symmetric positive definite.

We assume here the direct model to be linear (or linearized). We consider a temporal discretization of the forward nudging problem (3.2), using for example an implicit scheme. If we denote by  $X^n$  the solution at time  $t_n$  and  $X^{n+1}$  the solution at time  $t_{n+1}$ , and  $\Delta t = t_{n+1} - t_n$ , then equation (3.2) becomes

$$\frac{X^{n+1} - X^n}{\Delta t} = FX^{n+1} + K(X_{obs} - CX^{n+1}). \quad (3.7)$$

We now set the nudging matrix to be

$$K = C^T R^{-1}. \quad (3.8)$$

Then, it is straightforward to see that problem (3.7) is equivalent to the following optimization problem:

$$X^{n+1} = \arg \min_X \left[ \frac{1}{2} \langle X - X^n, X - X^n \rangle - \frac{\Delta t}{2} \langle FX, X \rangle + \frac{\Delta t}{2} \langle R^{-1}(X_{obs} - CX), X_{obs} - CX \rangle \right]. \quad (3.9)$$

The first two terms correspond exactly to the energy of the discretized direct model, and the last term is the observation part of the variational cost function. This variational principle shows that at each time step, the nudging state is a compromise between minimizing the energy of the system and the distance to the observations.

As a consequence, there is no need to consider an additional term ensuring an initial condition close to the background state like in variational algorithms, neither for stabilizing or regularizing the problem, nor from a physical point of view. One can simply initialize the BFN scheme with the background state, without any information on its statistics of errors.

The nudging method naturally provides a correction to the model equations from the observations. The model equations are hence weak constraints in the BFN scheme. In some nonlinear cases, the  $\langle FX, X \rangle$  term in equation (3.9) can be replaced by  $-G(X)$ , where  $G$  is the energy of the system at equilibrium.

### Sequential interpretation

It is also possible to give a sequential interpretation of the standard nudging algorithm by seeing it as a Kalman filter. Indeed, when no observations are available, the nudging method simply consists of solving the model equations, like Kalman filters. On the other hand, when some observations are available, in both nudging and Kalman filters, the model solution is corrected with the innovation vector, i.e. the difference between the observations and the corresponding model state [8].

If at any time, the nudging matrices are set in an optimal way, then the standard nudging method is equivalent to the standard Kalman filter. In the other cases, it can be seen as a suboptimal Kalman filter. However, the iterative and alternative resolutions of forward and backward models appreciably improves the efficiency of the standard nudging method.

### Pole assignment method and backward nudging matrix

The goal of the backward nudging term is both to have a backward data assimilation system and to stabilize the integration of the backward system (3.4), as this system is usually ill posed. The choice of the backward nudging matrix is then imposed by this stability condition.

If we consider a linearized situation, in which the system and observation operators ( $F$  and  $C$ , respectively) are linear, and if we make the change of time variable  $t' = T - t$ , then the backward equation can be rewritten as

$$-\frac{d\tilde{X}}{dt'} = F\tilde{X} - K'(X_{obs} - C\tilde{X}). \quad (3.10)$$

Then, the matrix to be stabilized is  $-F - K'C$ , i.e. the eigenvalues of this matrix should have negative real parts.

We now recall the pole assignment result (see e.g. [80, 41, 60, 167]):

**Theorem 3.1** *If  $(F, C)$  is an observable system, where  $F$  is a  $n \times n$  matrix and  $C$  is a  $m \times n$  matrix (here  $n$  is the size of the control vector  $X$  and  $m$  is the size of the observation vector  $X_{obs}$ ), then there exists at least one matrix  $K'$  such that  $-F - K'C$  is a Hurwitz matrix, i.e. all its eigenvalues are in the negative half-plane.*

We should also recall that  $(F, C)$  is an observable system if and only if the rank of  $[C, CF, \dots, CF^{n-1}]$  is equal to  $n$ . Hence, we can assume that there exists at least one matrix  $K'$  such that the backward nudging system (3.4) is stable. However, such a matrix  $K'$  may be hard to compute, as it usually requires the resolution of a Riccati equation.

### 3.3 Numerical experiments [11, 14, 21]

#### 3.3.1 Numerical choice of the nudging matrices

All numerical experiments have been performed with an easy-to-implement nudging matrix:

$$K = C^T(kI) = kC^T, \quad (3.11)$$

where  $k$  is a positive scalar gain, and  $I$  is the identity matrix of the observation space. This choice is motivated by the following remarks. First, the covariance matrix of observation errors is usually not well known (but if it is available, then one should consider equation (3.8) for the definition of  $K$ ). Secondly, this choice does not require a costly numerical integration of a parameter estimation problem for the determination of the optimal coefficients. Choosing  $K = C^T L$ , where  $L$  is a square matrix in the observation space, has another interesting property: if the observations are not located at a model grid point, or are a function of the model state vector, i.e. if the observation operator  $C$  involves interpolation/extrapolation or some change of variables, then the nudging matrix  $K$  will contain the adjoint operations, i.e. some interpolation/extrapolation back to the model grid points, or the inverse change of variable.

As in the forward part of the algorithm, for simplicity reasons we make the following choice for the backward nudging matrix  $K'$ :

$$K' = C^T(k'I) = k'C^T. \quad (3.12)$$

The only parameters of the BFN algorithm are then the coefficients  $k$  and  $k'$ . In the forward mode,  $k > 0$  is usually chosen such that the nudging term remains small in comparison with the other terms in the model equation. The coefficient  $k'$  is usually chosen to be the smallest coefficient that makes the numerical backward integration stable.

#### 3.3.2 Experimental approach

The same approach has been used for all the numerical experiments. This approach consists of performing twin experiments with simulated data. First, a reference experiment is run and the corresponding data are extracted. From now on this reference trajectory will be called the exact solution. Experimental data are supposed to be obtained every  $n_x$  gridpoints of the model, and every  $n_t$  time steps. The simulated data are then optionally noised with a Gaussian white noise distribution, and provided as observations to the assimilation scheme.

The first guess of the assimilation experiments is chosen to be either a constant field or the reference model state some time before the beginning of the assimilation period. Finally, the results of the assimilation process are compared with the exact solution.

### 3.3.3 Physical models

In this section, we briefly describe the various models on which the BFN algorithm has been implemented and compared with other data assimilation methods. For each experiment, we refer to some references for the details and results of the corresponding numerical experiments.

#### Lorenz equations

The BFN algorithm has been tested on Lorenz' chaotic system [127]:

$$\begin{cases} \frac{dx}{dt} = 10(y - x), \\ \frac{dy}{dt} = 28x - y - xz, \\ \frac{dz}{dt} = -\frac{8}{3}z + xy. \end{cases} \quad (3.13)$$

The Lorenz attractor is a nonlinear three-dimensional structure corresponding to the long-term behaviour of a chaotic flow, noted for its butterfly shape. Several numerical experiments, convergence results, and comparisons with the standard variational data assimilation method are presented in [11].

#### 1-D viscous Burgers' equation

We have then considered a very simple nonlinear geophysical model. The evolution model is the viscous Burgers' equation over a one-dimensional cyclic domain:

$$\frac{\partial X}{\partial t} + \frac{1}{2} \frac{\partial(X^2)}{\partial s} - \nu \frac{\partial^2 X}{\partial s^2} = 0, \quad (3.14)$$

where  $X$  is the state variable,  $s$  represents the distance in meters around the 45°N constant-latitude circle, and  $t$  is the time. The sampling of the observations provide a spatial and temporal density similar to the longitudinal distribution of the mid-latitude radiosonde network. The period of the domain, the diffusion coefficient, and the length of the assimilation period also make the situation as realistic as possible.

Note that this system is nonlinear and the viscosity makes it irreversible. However, it is possible to stabilize the backward resolution with the nudging term. The numerical and convergence results, as well as the comparison with the variational scheme, are detailed in [11]. Some other numerical experiments and comparisons are detailed in [21] in a slightly different situation (i.e. different physical parameters).

### Shallow water model

The shallow water model (or Saint-Venant's equations) is a basic model, representing quite well the temporal evolution of geophysical flows. This model is usually considered for simple numerical experiments in oceanography, meteorology or hydrology. The shallow water equations are a set of three equations, describing the evolution of a two-dimensional horizontal flow. These equations are derived from a vertical integration of the three-dimensional fields, assuming the hydrostatic approximation, i.e. neglecting the vertical acceleration. There are several ways to write the shallow water equations, considering either the geopotential or height or pressure variables. We consider here the following configuration:

$$\begin{cases} \partial_t u - (f + \zeta)v + \partial_x B = \frac{\tau}{\rho_0 h} - ru + \nu \Delta u, \\ \partial_t v + (f + \zeta)u + \partial_y B = \frac{\tau}{\rho_0 h} - rv + \nu \Delta v, \\ \partial_t h + \partial_x(hu) + \partial_y(hv) = 0, \end{cases} \quad (3.15)$$

where  $\zeta = \partial_x v - \partial_y u$  is the relative vorticity,  $B = g^* h + \frac{1}{2}(u^2 + v^2)$  is the Bernoulli potential,  $g^*$  is the reduced gravity,  $f$  is the Coriolis parameter (in the  $\beta$ -plane approximation),  $\rho_0$  is the water density,  $r$  is the friction coefficient, and  $\nu$  is the viscosity (or dissipation) coefficient. The unknowns are  $u$  and  $v$  the horizontal components of the velocity, and  $h$  the geopotential height. Finally,  $\tau$  is the forcing term of the model (e.g. the wind stress) [56].

Many numerical experiments and comparisons with the variational scheme are presented in [14]. This article also reports the results of an hybridization between the BFN and variational schemes.

### Layered quasi-geostrophic ocean model

We have finally considered a layered quasi-geostrophic ocean model [107, 169, 57]. This model arises from the primitive equations (conservation laws of mass, momentum, temperature and salinity), assuming first that the rotational effect (Coriolis force) is much stronger than the inertial effects. The Rossby number, ratio between the characteristic time of the Earth's rotation and the inertial time, must then be small compared to 1. Second, the thermodynamic effects are completely neglected in this model. Quasigeostrophy assumes that the horizontal dimension of the ocean is small compared to the size of the Earth, with a ratio of the order of the Rossby number. We finally assume that the depth of the basin is small compared to its width. In the case of the Atlantic Ocean, not all these assumptions are valid, notably the horizontal extension of the ocean. But it has been shown that the quasi-geostrophic approximation is fairly robust in practice, and that this approximate model reproduces quite well the ocean circulations at mid-latitudes, such as the jet stream (e.g. Gulf Stream in the case of the North Atlantic Ocean) and ocean boundary currents.

The model system is then composed of  $n$  coupled equations resulting from the conservation law of the potential vorticity. The equations can be written as:

$$\frac{D_1(\theta_1(\Psi) + f)}{Dt} + A_4 \nabla^6 \Psi_1 = F_1 \quad \text{in } \Omega \times ]0, T[, \quad (3.16)$$

at the surface layer ( $k = 1$ );

$$\frac{D_k(\theta_k(\Psi) + f)}{Dt} + A_4 \nabla^6 \Psi_k = 0 \quad \text{in } \Omega \times ]0, T[, \quad (3.17)$$

at the intermediate layers ( $k = 2, \dots, n - 1$ );

$$\frac{D_n(\theta_n(\Psi) + f)}{Dt} + A_1 \Delta \Psi_n + A_4 \nabla^6 \Psi_n = 0, \quad (3.18)$$

in  $\Omega \times ]0, T[$ , at the bottom layer ( $k = n$ ).

$\Omega$  is the circulation basin,  $\psi_k$  is the stream function at layer  $k$ ,  $\theta_k$  is the sum of the dynamical and thermal vorticities at layer  $k$ ,  $f$  is the Coriolis force, and the dissipative terms correspond to the lateral friction and the bottom friction dissipation. Finally,  $F_1$  is the forcing term of the model, the wind stress applied to the ocean surface. We refer to [107, 169, 57] for more details about this model and its equations.

We refer to [11] for the reports of numerical simulations on this model: convergence, comparison with the 4D-VAR algorithm, sensitivity studies, ...

### 3.3.4 Conclusions emerging from the numerical experiments

The BFN algorithm appears to be a very promising data assimilation method. It is extremely easy to implement: no linearization of the model equations, no computation of the adjoint state, no optimization algorithm. The only necessary work is to add a relaxation term to the model equations. The key point in the backward integration is that the nudging term (with the opposite sign to the forward integration one) makes it numerically stable. Hence the nudging (or relaxation) term has a double role: it forces the model to the observations and it stabilizes the numerical integration. It is simultaneously a penalization and regularization term.

The BFN algorithm has been compared with the variational method on several types of non-linear systems: Lorenz (chaotic 1D ODE), Burgers (1D PDE), shallow water model (2D) and quasi-geostrophic model (3D). The conclusion of these various experiments is that the BFN algorithm is better than the variational method for the same number of iterations (and hence for the same computing time). It converges in a small number of iterations. Of course the initial condition is usually poorly identified by the BFN scheme, but on the other hand, the final state of the assimilation period is much better identified by the BFN algorithm than by the variational assimilation algorithm, which is a key point for the prediction phase that starts at the end of the assimilation period. Hence the prediction phase is usually better when it comes after



an assimilation period treated by the BFN algorithm, rather than by a variational assimilation method.

The two algorithms can be combined: we have introduced a new hybrid scheme, in which a very small number of BFN iterations are performed (2 or 3 for instance), before providing the identified initial condition to the standard 4D-VAR algorithm. By doing this, the convergence of the 4D-VAR is reached more quickly, as it sometimes divides by two the number of iterations required. Also, for a fixed given number of iterations (or for a given computation time), the quality of the identified solution is significantly improved by this preprocessing (note that the number of 4D-VAR iterations is decreased by the number of BFN iterations in this scheme, in order to consider the same number of iterations in the standard 4D-VAR and the hybrid scheme).

Finally the BFN algorithm enables one to consider the problem of imperfect models at no additional cost, as the model equations are not strong constraints in this nudging method (while they are usually strong constraints in a variational method) and the relaxation term can be seen as a model error term.

## 3.4 Theoretical convergence results [18, 24]

### 3.4.1 Linear case

We consider here a linear situation, although simple, that describes quite well how the BFN algorithm works. We assume that the observation operator  $C$  is equal to the identity, and that the model  $F$  is linear. We also assume that  $F$  and  $K$  commute. Note that this assumption is valid in our experiments as  $K$  is set proportional to the identity matrix. In this pretty simple situation, we can explicit the BFN trajectories. For the sake of concision and clarity, we assume that  $K' = K$ , but the following results remain valid if  $K' \neq K$ . We finally assume that the length of the assimilation period is  $T > 0$ .

Then, for all  $n > 1$ ,

$$X_n(0) = (I - e^{-2KT})^{-1} (I - e^{-2nKT}) \int_0^T \left( e^{-(K+F)s} + e^{-2KT} e^{(K-F)s} \right) K X_{obs}(s) ds + e^{-2nKT} x_0 \quad (3.19)$$

and for all  $t \in [0, T]$ ,

$$X_n(t) = e^{-(K-F)t} \int_0^t e^{(K-F)s} K X_{obs}(s) ds + e^{-(K-F)t} X_n(0). \quad (3.20)$$

The following result proves the existence of a limit trajectory [18]:

**Theorem 3.2** *If  $n \rightarrow +\infty$ , then  $X_n(0)$  converges and*

$$\lim_{n \rightarrow +\infty} X_n(0) = X_\infty(0) = (I - e^{-2KT})^{-1} \int_0^T \left( e^{-(K+F)s} + e^{-2KT} e^{(K-F)s} \right) K X_{obs}(s) ds. \quad (3.21)$$

Moreover, if  $T > 0$ , for any  $t \in [0, T]$ ,

$$\lim_{n \rightarrow +\infty} X_n(t) = X_\infty(t) = e^{-(K-F)t} \int_0^t e^{(K-F)s} K X_{obs}(s) ds + e^{-(K-F)t} X_\infty(0). \quad (3.22)$$

Under the same hypothesis, we have a similar result for backward trajectories, i.e. there exists a function  $\tilde{X}_\infty(t)$  such that  $\lim_{n \rightarrow +\infty} \tilde{X}_n(t) = \tilde{X}_\infty(t)$ , for all  $t \in [0, T]$ . This proves the convergence of the BFN algorithm in such a situation.

Note that the limit function  $X_\infty$  (resp.  $\tilde{X}_\infty$ ) is totally independent of the initial condition  $x_0$  of the algorithm.

Moreover, if the observations are perfect, i.e.  $X_{obs}$  satisfies the direct model equation (3.1), then for all  $t \in [0, T]$ ,

$$X_{obs}(t) = e^{Ft} X_{obs}(0). \quad (3.23)$$

It is then straightforward to see in equations (3.21) and (3.22) that

$$\lim_{n \rightarrow \infty} X_n(t) = X_{obs}(t), \quad \forall t \in [0, T]. \quad (3.24)$$

The BFN algorithm also has a similar behaviour on linear parabolic operators in infinite dimension (e.g. the heat operator). A Fourier decomposition of the trajectories allows us to study only first order ordinary differential equations, and gives then the convergence of the algorithm.

### 3.4.2 Transport equations

In this section, we only consider one iteration of the back and forth nudging algorithm, i.e. one forward and one backward resolution with nudging terms. The idea consists of studying the decrease of the error (between the BFN trajectory and the true solution) during one iteration. All the results can then be extended very easily to an arbitrary number of BFN iterations. For instance, if the error decreases by a constant factor of less than 1 during one iteration, then the algorithm is contractive and the error decreases exponentially to 0 with the iterations.

We refer to [24] for the proofs of all the following results.

#### Linear viscous transport

We first consider a linear viscous transport equation.

$$\begin{aligned} (F) \quad & \begin{cases} \partial_t u - \nu \partial_{xx} u + a(x) \partial_x u &= -K(u - u_{obs}), \\ u|_{x=0} = u|_{x=1} &= 0, \\ u|_{t=0} &= u_0, \end{cases} \\ (B) \quad & \begin{cases} \partial_t \tilde{u} - \nu \partial_{xx} \tilde{u} + a(x) \partial_x \tilde{u} &= K'(\tilde{u} - u_{obs}), \\ \tilde{u}|_{x=0} = \tilde{u}|_{x=1} &= 0, \\ \tilde{u}|_{t=T} &= u(T), \end{cases} \end{aligned} \quad (3.25)$$

where the following notations hold for all further cases:

- the time period considered here is  $t \in [0, T]$ ;
- the first equation ( $F$ ) is called the forward equation, the second one ( $B$ ) is called the backward one;
- $K$  and  $K'$  are positive functions, and may depend on  $t$  and  $x$ , but for simplicity reasons, we will always assume that there exists a constant  $\kappa \in \mathbb{R}_+^*$  such that  $K'(t, x) = \kappa K(t, x)$ ;
- $a(x) \in W^{1,\infty}(\Omega)$ ,  $\Omega$  being the considered space domain, either the interval  $[0, 1]$  or the torus  $[0, 1]$ ;
- the viscosity coefficient  $\nu > 0$  is constant;
- the observation function  $u_{obs}$  is a solution of the forward equation (without any nudging term) with initial condition  $u_{obs}^0$ :

$$\begin{cases} \partial_t u_{obs} - \nu \partial_{xx} u_{obs} + a(x) \partial_x u_{obs} = 0, \\ u|_{x=0} = u|_{x=1} = 0, \\ u|_{t=0} = u_{obs}^0. \end{cases} \quad (3.26)$$

Then, the following result holds true for linear viscous transport equations [24]:

**Theorem 3.3** *We consider one step of the BFN algorithm (3.25) with observations  $u_{obs}$  satisfying equation (3.26). We denote*

$$\begin{aligned} w(t) &= u(t) - u_{obs}(t), \\ \tilde{w}(t) &= \tilde{u}(t) - u_{obs}(t), \end{aligned} \quad (3.27)$$

the forward and backward errors.

1. If  $K(t, x) = K$ , then for all  $t \in [0, T]$ :

$$\tilde{w}(t) = e^{(-K-K')(T-t)} w(t). \quad (3.28)$$

2. If  $K(t, x) = K(x)$ , with Support  $(K) \subset [a, b]$  where  $a < b$  and  $a \neq 0$  or  $b \neq 1$ , then equation (3.25) is ill-posed: there does not exist a solution  $(u, \tilde{u})$  in general.

3. If  $K(t, x) = K \mathbb{1}_{[t_1, t_2]}(t)$  with  $0 \leq t_1 < t_2 \leq T$ , then we have

$$\tilde{w}(0) = e^{(-K-K')(t_2-t_1)} w(0). \quad (3.29)$$

This result shows that, when applied to linear viscous transport equations, the BFN algorithm converges if the feedback acts on the entire domain. For instance, in the first point of theorem 3.3, equation (3.28) shows that the error has been decreased by a factor of  $e^{(-K-K')T}$  during one iteration. Thus, the error decreases by a factor of  $e^{-N(K+K')T}$  during  $N$  iterations. As  $K > 0$  (or  $K' > 0$ ) and  $T > 0$ , this clearly proves the convergence of the BFN algorithm in this case. On the contrary, if a part of the space domain is not observed (i.e. the support of  $K$  does not cover the entire domain), then the algorithm does not converge as the diffusion term cannot be controlled and the backward resolution is ill-posed.

### Viscous Burgers

We now consider the viscous Burgers' equation, a standard nonlinear transport equation. We also consider only one iteration of the BFN algorithm:

$$\begin{aligned}
 (F) \quad & \begin{cases} \partial_t u - \nu \partial_{xx} u + u \partial_x u &= -K(u - u_{obs}), \\ u|_{x=0} = u|_{x=1} &= 0, \\ u|_{t=0} &= u_0, \end{cases} \\
 (B) \quad & \begin{cases} \partial_t \tilde{u} - \nu \partial_{xx} \tilde{u} + \tilde{u} \partial_x \tilde{u} &= K'(\tilde{u} - u_{obs}), \\ \tilde{u}|_{x=0} = \tilde{u}|_{x=1} &= 0, \\ \tilde{u}|_{t=T} &= u(T), \end{cases}
 \end{aligned} \tag{3.30}$$

with the same notations as before. The observations  $u_{obs}$  also satisfy the forward Burgers' equation:

$$\begin{cases} \partial_t u_{obs} - \nu \partial_{xx} u_{obs} + u_{obs} \partial_x u_{obs} &= 0, \\ u|_{x=0} = u|_{x=1} &= 0, \\ u|_{t=0} &= u_{obs}^0. \end{cases} \tag{3.31}$$

Then, the following result holds true [24]:

**Theorem 3.4** *If  $K(t, x) \not\equiv 0$ , then the BFN iteration (3.30) for viscous Burgers' equation, with observations satisfying (3.31), is ill-posed, even when  $K(t, x)$  is constant: there does not exist, in general, a solution  $(u, \tilde{u})$ .*

In the particular case where  $K \equiv K' \equiv 0$ , the backward problem is ill-posed in the sense of Hadamard (the solution does not depend continuously on the data), but it has a unique solution if the final condition  $\tilde{u}|_{t=T}$  is set to a final solution of the direct equation. Moreover, in this particular case, the backward solution is exactly equal to the forward one:  $\tilde{u}(t) = u(t)$  for all  $t \in [0, T]$ . The main result is the following [24]:

**Proposition 3.1** *If  $K \equiv K' \equiv 0$ , then problem (3.30) is well-posed in the sense of Hadamard, and there exists a unique solution  $(u, \tilde{u})$ . Moreover,  $u = \tilde{u}$ .*

The BFN algorithm is then ill-posed (except if  $K \equiv K' \equiv 0$ ) when applied to a viscous Burgers' equation, as there does not exist a solution to the backward problem. However, from the numerical point of view, the BFN algorithm has been successfully applied to this model [11]. This phenomenon is probably due to the fact that we numerically solve a discrete problem and not the exact continuous one.

### Inviscid linear transport

We now consider the inviscid case for a linear transport equation. The BFN equations are:

$$\begin{aligned}
 (F) \quad & \begin{cases} \partial_t u + a(x)\partial_x u &= -K(u - u_{obs}), \\ u|_{x=0} &= u|_{x=1}, \\ \partial_x u|_{x=0} &= \partial_x u|_{x=1}, \\ u|_{t=0} &= u_0, \end{cases} \\
 (B) \quad & \begin{cases} \partial_t \tilde{u} + a(x)\partial_x \tilde{u} &= K'(\tilde{u} - u_{obs}), \\ \tilde{u}|_{x=0} &= \tilde{u}|_{x=1}, \\ \partial_x \tilde{u}|_{x=0} &= \partial_x \tilde{u}|_{x=1}, \\ \tilde{u}|_{t=T} &= u(T), \end{cases}
 \end{aligned} \tag{3.32}$$

where  $a(x)$  can be constant or not. The following result holds true [24]:

**Theorem 3.5** *We consider the non viscous one-step BFN (3.32), with observations  $u_{obs}$  satisfying (3.32-F) with  $K = 0$ . We denote*

$$\begin{aligned}
 w(t) &= u(t) - u_{obs}(t), \\
 \tilde{w}(t) &= \tilde{u}(t) - u_{obs}(t).
 \end{aligned} \tag{3.33}$$

We denote by

$$(s, \psi(s, x)) \tag{3.34}$$

the characteristic curve of equation (3.32-F) with  $K = 0$ , with foot  $x$  in time  $s = 0$ , i.e. such that

$$(s, \psi(s, x))|_{s=0} = (0, x). \tag{3.35}$$

We assume that the final time  $T$  is such that the characteristics are well defined and do not intersect over  $[0, T]$ . Then:

1. If  $K(t, x) = K$ , then we have, for all  $t \in [0, T]$ ,

$$\tilde{w}(t) = w(t)e^{(-K-K')(T-t)}. \tag{3.36}$$

2. If  $K(t, x) = K\mathbb{1}_{[t_1, t_2]}(t)$  with  $0 \leq t_1 < t_2 \leq T$ , then we have

$$\tilde{w}(0) = w(0)e^{(-K-K')(t_2-t_1)}. \tag{3.37}$$

3. If  $K(t, x) = K(x)$ , then we have, for all  $t \in [0, T]$ ,

$$\tilde{w}(t, \psi(t, x)) = w(t, \psi(t, x)) \exp\left(-\int_t^T K(\psi(s, x)) + K'(\psi(s, x)) ds\right). \tag{3.38}$$

From this result, we deduce that the BFN algorithm applied to inviscid linear transport equation does converge if all the domain is observed (first two cases of theorem 3.5). Moreover, if the support of  $K$  does not cover all the domain (third case of theorem 3.5, e.g. when the system is not fully observed), the algorithm converges as soon as all the characteristics intersect the support of  $K$ . This constraint is satisfied as soon as the system is observable (see remarks below proposition 3.2).

### Inviscid Burgers

We finally consider non viscous Burgers' equation, with periodic boundary conditions, and for a time  $T$  such that there is no shock in the interval  $[0, T]$ :

$$\begin{aligned}
 (F) \quad & \begin{cases} \partial_t u + u \partial_x u &= -K(u - u_{obs}), \\ u|_{x=0} &= u|_{x=1}, \\ \partial_x u|_{x=0} &= \partial_x u|_{x=1}, \\ u|_{t=0} &= u_0, \end{cases} \\
 (B) \quad & \begin{cases} \partial_t \tilde{u} + \tilde{u} \partial_x \tilde{u} &= K'(\tilde{u} - u_{obs}), \\ \tilde{u}|_{x=0} &= \tilde{u}|_{x=1}, \\ \partial_x \tilde{u}|_{x=0} &= \partial_x \tilde{u}|_{x=1}, \\ \tilde{u}|_{t=T} &= u(T). \end{cases}
 \end{aligned} \tag{3.39}$$

Then, the following result holds true [24]:

**Theorem 3.6** *We consider one step of the BFN algorithm applied to the non viscous Burgers' equation (3.39), with observations  $u_{obs}$  satisfying (3.39-F) with  $K = 0$ . We denote*

$$\begin{aligned}
 w(t) &= u(t) - u_{obs}(t), \\
 \tilde{w}(t) &= \tilde{u}(t) - u_{obs}(t).
 \end{aligned} \tag{3.40}$$

We assume that  $u_{obs} \in W^{1,\infty}([0, T] \times \Omega)$ , i.e. there exists  $M > 0$  such that

$$|\partial_x u_{obs}(t, x)| \leq M, \quad \forall t \in [0, T], \forall x \in \Omega. \tag{3.41}$$

Then:

1. If  $K(t, x) = K$ , then we have, for all  $t \in [0, T]$ ,

$$\|\tilde{w}(t)\| \leq e^{(-K-K'+M)(T-t)} \|w(t)\|. \tag{3.42}$$

2. If  $K(t, x) = K \mathbb{1}_{[t_1, t_2]}(t)$  with  $0 \leq t_1 < t_2 \leq T$ , then we have

$$\|\tilde{w}(0)\| \leq e^{(-K-K')(t_2-t_1)+MT} \|w(0)\|. \tag{3.43}$$

**Proposition 3.2** *We consider one forward (resp. backward) BFN step of the non viscous Burgers' equation (3.39-F) (resp. (3.39-B)). With the notations of theorem 3.6, if  $K(t, x) = K(x)$ , then we have*

$$w(T, \psi(T, x)) = w(0, x) \exp \left( - \int_0^T K(\psi(\sigma, x)) d\sigma - \int_0^T \partial_x u_{obs}(\sigma, \psi(\sigma, x)) d\sigma \right). \tag{3.44}$$

**Remark:** For the special case  $K(t, x) = K(x) = K \mathbb{1}_{[a,b]}(x)$ , where  $K$  is a constant and  $[a, b]$  is a sub-interval of  $[0, 1]$ , we have

$$w(T, \psi(T, x)) = w(0, x) \exp \left( -K\chi(x) - \int_0^T \partial_x u_{obs}(\sigma, \psi(\sigma, x)) d\sigma \right), \quad (3.45)$$

where

$$\chi(x) = \int_0^T \mathbb{1}_{Supp(K)}(\psi(\sigma, x)) d\sigma \quad (3.46)$$

is the time during which the characteristic curve  $\psi(\sigma, x)$  with foot  $x$  of equation (3.39-F) with  $K = 0$  lies in the support of  $K$ .

The system is then observable if and only if the function  $\chi$  has a non-zero lower bound, i.e.  $m := \min_x \chi(x) > 0$ , the observability being defined by (see e.g. [147]):

$$\exists C, \forall u \text{ solution of (3.39-F) with } K = 0, \quad \|u(T, \cdot)\|^2 \leq C \int_0^T \|K(\cdot)u(s, \cdot)\|^2 ds. \quad (3.47)$$

In this case, proposition 3.2 proves the global exponential decrease of the error, provided  $K$  is larger than  $\frac{MT}{m}$ , where  $M$  is defined by equation (3.41).

From this remark, we can easily deduce that if for each iteration, both in the forward and backward integrations, the observability condition is satisfied, then the algorithm converges and the error decreases exponentially to 0. Note that this is not a necessary condition, as even if  $\chi(x) = 0$ , the last exponential of equation (3.45) is bounded.

### Remarks on the theoretical results

In real geophysical applications (either meteorology or oceanography), there is usually no viscosity. In this case, assuming the observability condition, the BFN algorithm is well posed, and theorem 3.6 and proposition 3.2 show that the solution tends to the observation trajectory everywhere, and not only on the support of  $K$  [24].

From a numerical point of view, we can observe that even with discrete and sparse observations in space, the numerical solution is corrected everywhere. We also observed that, with a not too large viscosity coefficient, the behaviour of the algorithm remains unchanged [11].

## 3.5 Nudging and observers [25]

In this section, we consider nudging as a particular type of observers, e.g. Luenberger observer, or Kalman filters [129, 114]. In most Kalman-type filters and observers, the gain matrices do not take into account the symmetries of the model. They are mainly designed to provide for each time step the optimal estimate of the system state. However, it seems interesting to preserve the model symmetries while

adding a nudging term in the equations. Inspired by the recent works in observer design, we define symmetry-preserving nudging (or feedback) terms for a shallow-water model. This section summarizes the work presented in [25].

### 3.5.1 Observers for a shallow water model

We consider here a shallow water model, similar to the model introduced in section 3.3.3. However, the equations are rewritten in order to clearly see the symmetries. We refer to [113] for more details about these equations. In the following,  $h$  is the fluid height, and  $v$  is the bi-dimensional velocity field. The equations write:

$$\frac{\partial(hv)}{\partial t} + (\nabla \cdot (hv) + (hv) \cdot \nabla) v = -g' h \nabla h - k \times f(hv) + (A \nabla^2 - R)(hv) + \frac{\tilde{\tau}}{\rho} i \quad (3.48)$$

for the vectorial velocity, and

$$\frac{\partial h}{\partial t} = -\nabla \cdot (hv) \quad (3.49)$$

for the scalar height. In these equations,  $g'$  represents the reduced gravity,  $\rho$  is the fluid density,  $f$  is the Coriolis parameter,  $i$  is the longitudinal unit vector (pointing towards East) and  $k$  is the upward unit vector. Finally,  $R$ ,  $A$  and  $\tilde{\tau}$  represent friction, lateral viscosity, and the forcing term (zonal wind stress) respectively.

We assume that the physical system is observed by several satellites that provide measurements of the sea surface height (SSH)  $h$  only.

An observer  $(\hat{h}, \hat{v})$  for the system (3.48-3.49) writes:

$$\frac{\partial(\hat{h}\hat{v})}{\partial t} + \left( \nabla \cdot (\hat{h}\hat{v}) + (\hat{h}\hat{v}) \cdot \nabla \right) \hat{v} = -g' \hat{h} \nabla \hat{h} - k \times f(\hat{h}\hat{v}) + (A \nabla^2 - R)(\hat{h}\hat{v}) + \frac{\tilde{\tau}}{\rho} i + F_v(h, \hat{v}, \hat{h}) \quad (3.50)$$

and

$$\frac{\partial \hat{h}}{\partial t} = -\nabla \cdot (\hat{h}\hat{v}) + F_h(h, \hat{v}, \hat{h}). \quad (3.51)$$

The only difference between the observer and model equations comes from the innovation terms  $F_v(h, \hat{v}, \hat{h})$  and  $F_h(h, \hat{v}, \hat{h})$ . The correction terms must vanish when the estimated height  $\hat{h}$  is equal to the observed height  $h$ . The goal is to define functions  $F_h$  and  $F_v$  such that the observer tends to the true solution. Moreover, these feedback terms also have to preserve the symmetries of the model.

### 3.5.2 Invariant correction terms

The shallow-water equations do not depend neither on the orientation nor on the origin of the frame in which the coordinates are expressed: they are invariant under the action of  $SE(2)$ , the Special Euclidean group of isometries of the plane  $\mathbb{R}^2$ . Consequently, functions  $F_h$  and  $F_v$  must be invariant under the action of  $SE(2)$ . Symmetries have been very recently introduced for observer design in [28, 59] for engineering problems. The aim of this work is to consider correction terms that respect the underlying physics of the system.



To find the scalar term  $F_h$ , we use the standard result (see e.g. [155]), which states that any scalar differential operator invariant by rotation and translation writes  $Q(\Delta)$ , where  $Q$  is a polynomial and  $\Delta$  is the Laplacian. By considering the invariance by rotation for the vectorial velocity [139], we get the following family of scalar terms:

$$F_h = Q_1(\Delta, h, |\hat{v}|^2, \hat{h} - h) + \nabla \left( Q_2(\Delta, h, |\hat{v}|^2, \hat{h} - h) \right) \cdot \hat{v} + f_h, \quad (3.52)$$

where  $Q_1$  and  $Q_2$  are scalar polynomials in  $\Delta$ , and  $f_h$  is an integral term defined below. More precisely,

$$Q_i(\Delta, h, |\hat{v}|^2, \hat{h} - h) = \sum_{k=0}^N a_k^i(h, |\hat{v}|^2, \hat{h} - h) \Delta^k \left( b_k^i(h, |\hat{v}|^2, \hat{h} - h) \right), \quad (3.53)$$

where  $a_k^i$  and  $b_k^i$  are smooth scalar functions such that

$$a_k^i(h, |\hat{v}|^2, 0) = b_k^i(h, |\hat{v}|^2, 0) = 0. \quad (3.54)$$

For the vectorial correction term  $F_v$ , we use the vectorial counterpart:

$$F_v = P_1(\Delta, h, |\hat{v}|^2, \hat{h} - h) \hat{v} + \nabla \left( P_2(\Delta, h, |\hat{v}|^2, \hat{h} - h) \right) + f_v, \quad (3.55)$$

where  $P_1$  and  $P_2$  are polynomials in  $\Delta$ , like  $Q_1$  and  $Q_2$ .

Let us now find the integral terms  $f_v$  and  $f_h$  that are invariant by rotation and translation. They can be expressed as a convolution between the previous invariant differential terms and a two-dimensional kernel  $\psi(\xi, \zeta)$ . The previous terms being invariant by rotation, the value of the kernel should not depend on a particular direction, and so  $\psi$  must be a function of the invariant  $\xi^2 + \zeta^2$ . The integral correction terms write:

$$f_v(x, y, t) = \iint \left[ R_1(\Delta, h, |\hat{v}|^2, \hat{h} - h) \hat{v} + \nabla \left( R_2(\Delta, h, |\hat{v}|^2, \hat{h} - h) \right) \right]_{(x-\xi, y-\zeta, t)} \phi_v(\xi^2 + \zeta^2) d\xi d\zeta, \quad (3.56)$$

$$f_h(x, y, t) = \iint \left[ S_1(\Delta, h, |\hat{v}|^2, \hat{h} - h) + \nabla \left( S_2(\Delta, h, |\hat{v}|^2, \hat{h} - h) \right) \cdot \hat{v} \right]_{(x-\xi, y-\zeta, t)} \phi_h(\xi^2 + \zeta^2) d\xi d\zeta, \quad (3.57)$$

where  $R_i$  and  $S_i$  are defined like  $Q_i$  and  $P_i$ .

The support of  $\phi_v$  (resp.  $\phi_h$ ) is a subset of  $\mathbb{R}$ . Its characteristic size defines a zone in which it is significant to correct the estimation with the measurements. The integral formulation is actually quite general: if  $\phi_v$  and  $\phi_h$  are set equal to Dirac functions, one obtains the differential terms.

### 3.5.3 Convergence study on a linearized simplified system

In order to avoid the amplification of the measurement noise by a differentiation process, only the integral correction terms are kept: one sets  $Q_1 = Q_2 = P_1 = P_2 = 0$ ,  $R_1 = S_2 = 0$  and  $S_1 = R_2 = h - \hat{h}$ .

For the sake of clarity, we now simplify the model equations, by assuming that there is no Coriolis force, no friction, no dissipation, and no wind stress. An observer for this simplified system satisfies then:

$$\frac{\partial \hat{h}}{\partial t} = -\nabla(\hat{h}\hat{v}) + \phi_h * (h - \hat{h}), \quad (3.58)$$

$$\frac{\partial \hat{v}}{\partial t} = -\hat{v}\nabla\hat{v} - g\nabla\hat{h} + \phi_v * \nabla(h - \hat{h}). \quad (3.59)$$

Note that in the degenerate case where  $\phi_h = K_h\delta_0$  and  $\phi_v = K_v\delta_0$ ,  $K_h$  and  $K_v$  being positive scalars, we find the standard nudging terms, or Luenberger observer.

As it seems difficult to first study the convergence on the nonlinear system, we now linearize the equations around an equilibrium position  $h = \bar{h}$  and  $v = \bar{v}$ . We only consider small velocities  $\delta v = v - \bar{v} \ll \sqrt{g\bar{h}}$  and heights  $\delta h = h - \bar{h} \ll \bar{h}$ , where  $\bar{h}$  and  $\bar{v} = 0$  represent the equilibrium height and speed respectively. We denote by  $\tilde{h}$  (resp.  $\tilde{v}$ ) the estimation errors, differences between the observer and true solution, for the height (resp. velocity). These errors are solution of the following linear equations:

$$\frac{\partial \tilde{h}}{\partial t} = -\bar{h}\nabla\tilde{v} - \phi_h * \tilde{h}, \quad (3.60)$$

$$\frac{\partial \tilde{v}}{\partial t} = -g\nabla\tilde{h} - \phi_v * \nabla\tilde{h}. \quad (3.61)$$

A reasonable choice for the kernels  $\phi_h$  and  $\phi_v$  is the following:

$$\phi_h(x, y) = \beta_h \exp(-\alpha_h(x^2 + y^2)), \quad (3.62)$$

$$\phi_v(x, y) = \beta_v \exp(-\alpha_v(x^2 + y^2)), \quad (3.63)$$

as one usually assumes that the observation error is a white Gaussian noise. However, the following convergence results can be extended to more general kernel functions defined by

$$\phi_h(x, y) = (f(x) * f(x))(f(y) * f(y)), \quad (3.64)$$

$$\phi_v(x, y) = (g(x) * g(x))(g(y) * g(y)), \quad (3.65)$$

$$(3.66)$$

where  $f$  and  $g$  are smooth even functions, all their Fourier coefficients being strictly positive.

Eliminating the velocity  $\tilde{v}$  in equations (3.60-3.61) leads to a modified damped wave equation with external viscous damping:

$$\frac{\partial^2 \tilde{h}}{\partial t^2} = g\bar{h}\Delta\tilde{h} + \bar{h}\phi_v * \Delta\tilde{h} - \phi_h * \frac{\partial \tilde{h}}{\partial t}. \quad (3.67)$$

Equation (3.67) can be rewritten in the following way:

$$\frac{\partial^2 \tilde{h}}{\partial t^2} = \phi_v * \Delta\tilde{h} - \phi_h * \frac{\partial \tilde{h}}{\partial t}, \quad (3.68)$$

if  $\phi_v$  is now the following function

$$\phi_v(x, y) = g\bar{h}\delta_0 + \bar{h}\beta_v \exp(-\alpha_v(x^2 + y^2)), \quad (3.69)$$

where  $\delta_0$  is the Dirac measure at the origin.

Then, we have the following result [25]:

**Theorem 3.7**

$$\lim_{t \rightarrow +\infty} \int_{\Omega} \left( \|\nabla \tilde{h}\|^2 + \left| \frac{\partial \tilde{h}}{\partial t} \right|^2 \right) = 0. \quad (3.70)$$

This result proves the strong and asymptotic convergence of the error  $\tilde{h}$  towards 0, and then it also gives the same convergence for  $\tilde{v}$ . We deduce that the observer tends to the true state when time goes to infinity. Note that even if only the height is observed, all variables are corrected.

A dimensional analysis also provides the following gain tuning (see equations (3.62) and (3.69)):

$$\beta_h = 2\xi_0\omega_0, \quad \bar{h}\beta_v = L_0^2\omega_0^2 - g\bar{h}, \quad (3.71)$$

where  $\omega_0$  and  $L_0$  are characteristic pulsation and length of the flow respectively, and  $\xi_0$  is the damping coefficient of the system equation. Moreover,  $\alpha_h^{-2} = \alpha_v^{-2}$  is the size of the region of influence. This region can be related to the level of observation noise, and to the spatial density of the observations.

### 3.5.4 Numerical experiments

The results of many numerical simulations on both the linearized and nonlinear shallow water models are reported in [25]. The following feedback terms have been considered:  $\phi_h * (h - \hat{h})$  for the fluid height, and  $\phi_v * \nabla(h - \hat{h})$  for the velocity, where  $\phi_h$  and  $\phi_v$  are defined by equations (3.62) and (3.63). Several values of the parameters  $\alpha_h$ ,  $\alpha_v$ ,  $\beta_h$  and  $\beta_v$  are considered, as well as several levels of observation noise. A comparison between the standard nudging (or Luenberger observer) and this observer is also given in [25].

All these simulations show the interest of such a choice of invariant gains. They provide better results than the standard nudging, even on the nonlinear system, because the error converges faster, the residual error is smaller, and noise is better filtered. Indeed the observer is nearly insensible to gaussian white noise. The numerical experiments also confirm that, as predicted by the theory, it is possible to correct the non-observed variables with the observed ones, thanks to model coupling.

Note that the computational cost of such an observer is not much larger than for the standard nudging, as we have considered a truncated convolution integral instead of the complete convolution over the whole domain. The truncation radius can be set equal to at most 10 pixels in similar experiments.

Several other gain functions should now be studied to see if it is possible to filter other types of observation noise. Some experiments will also be carried out in the case of sparse observations, both in time and space.

### 3.6 Conclusion

The BFN algorithm appears to be a very promising data assimilation method. It is extremely easy to implement: no linearization of the model equations, no computation of the adjoint state, no optimization algorithm. The only necessary work is to add a relaxation term to the model equations. The key point in the backward integration is that the nudging term (with the opposite sign to the forward integration one) makes it numerically stable. Hence the nudging (or relaxation) term has a double role: it forces the model to the observations and it stabilizes the numerical integration. It is simultaneously a penalization and regularization term.

The BFN algorithm has been compared with the variational method on several types of non-linear and turbulent systems. The conclusion of the various experiments is that the BFN algorithm is better than the variational method for the same number of iterations (and hence for the same computing time). It converges in a small number of iterations. Of course the initial condition is usually poorly identified by the BFN scheme, but on the other hand, the final state of the assimilation period is much better identified by the BFN algorithm than by the variational assimilation algorithm, which is a key point for the prediction phase that starts at the end of the assimilation period. Hence the prediction phase is usually better when it comes after an assimilation period treated by the BFN algorithm, rather than by a variational assimilation method.

The two algorithms can be combined, in the sense that one can perform several BFN iterations before switching to the variational method and this will considerably accelerate the convergence of the variational method. Finally the BFN algorithm enables one to consider the problem of imperfect models at no additional cost, as the model equations are not strong constraints in this nudging method (while they usually are in a variational method) and the relaxation term can be seen as a model error term.

Finally, several theoretical results explain and justify the efficiency of this algorithm in simple situations.

The main perspective is the following: the determination of the nudging coefficients (or matrices) should be improved, particularly by a numerical stability study of the backward integration. This will give the optimal nudging coefficients that make the backward integration stable, while preserving the extreme simplicity of the algorithm.

## Chapter 4

# Image data assimilation

This chapter summarizes the work presented in [23].

### 4.1 Introduction

This chapter presents a study at the interface of image processing and data assimilation: the assimilation of images. The numerical forecast of geophysical fluids is extremely difficult, mainly because they are governed by the general nonlinear equations of fluid dynamics. Over the past 20 years, observations of ocean and atmosphere circulation have become much more readily available, as a result of new satellite techniques. However, the huge amount of information provided by satellite images must therefore be exploited, as more and more space-borne observations of increasing quality are available.

Several ideas have been very recently developed to assimilate image data. A first idea consists of identifying some characteristic structures of the image and then in tracking them in time. This is currently developed in meteorology, using an adaptive thresholding technique for radiance temperatures in order to identify and track several cells [135]. Another idea is to consider a dual problem and to create some model images, coming from the numerical model itself, and to compare the satellite images with these model images, using for example a curvlet approach [132].

We propose here to define a fast and efficient way to identify, or extract, velocity fields from several images (or a complete sequence of images). Assuming this point, we would then be able to obtain billions of pseudo-observations, corresponding to the extracted velocity fields, that could be considered in the usual data assimilation processes. The main advantage of such an approach is to provide a lot of information on the velocity, which is a state variable of all geophysical models, as it is much more easy to assimilate data that are directly related to the state variables. We should mention that a satellite image can have a resolution of  $5000 \times 5000$  pixels, and that some satellites transmit such images every 15 to 60 minutes [103]. We propose in this paper a way to identify one velocity vector for each pixel of the image. Of course, we will see that all the identified velocity fields are not reliable, mainly when there is

no visible characteristic phenomenon, but we should be able to provide an amount of information that is comparable to the currently assimilated observations.

The hypothesis that is underlying this work is that the grey level of the points are preserved during the motion, this is known as the constant brightness hypothesis. The constant brightness hypothesis was introduced in [109], and the linearized equation derived from this hypothesis is the cornerstone of optical flow methods [128, 39, 47]. This hypothesis is sometimes replaced by an integrated continuity equation in order to take into account the spreading of intensity sources [93, 94, 76, 122].

This hypothesis is justified here in the framework of oceanography, as the object of interest, allowing us to track the fluid and identify its velocity, is usually a passive tracer, at least on relative short time periods: chlorophyll, sea surface temperature, chemical pollutants (e.g. hydrocarbons), . . . All these tracers do not interact with the water on a short time period, and they are passively transported by the fluid.

We propose here to use an integrated version of the constant brightness hypothesis. Instead of linearizing the constant brightness hypothesis like in standard optical flow techniques, we define a nonlinear cost function that takes into account the fact that time sampling occurs at a finite rate. The cost function obtained from the integrated constant brightness assumption is minimized in nested subspaces of admissible displacement vector fields. Several regularization norms are considered.

We refer to [23] for the results of many numerical experiments, on both simulated and real data. These results show that our method provides very quickly full velocity fields, with an estimator of the quality of the results, while the PIV (Particle Imaging Velocimetry) method, currently considered as a reference method in fluid mechanics and oceanography, is unable to provide more than one pertinent velocity vector every  $10 \times 10$  pixels.

## 4.2 Description of the algorithm [23]

This section is devoted to the description of the algorithm that we propose.

### 4.2.1 Constant brightness assumption

Let  $\Omega \subset \mathbb{R}^2$  be the rectangular domain where the images are defined. The motion between the instants  $t_0$  and  $t_1$  where the images are  $I_0$  and  $I_1$  is then the vector field  $(u, v)$  such that for every point  $(x, y) \in \Omega$ ,

$$I_1(x + u(x, y), y + v(x, y)) = I_0(x, y). \quad (4.1)$$

A vector field satisfying equation (4.1) is not unique, this is known as the aperture problem in optical flow. Moreover, measurement errors make the equality (4.1) unlikely to be strictly satisfied.

### 4.2.2 Cost function

We propose then a least square optimization to replace the exact equality (4.1):

$$J(u, v) = \frac{1}{2} \int_{\Omega} [F(I_0, I_1; u, v)(x, y)]^2 dx dy + \frac{1}{2} \alpha R(u, v), \quad (4.2)$$

where  $R(u, v)$  is a spatial regularization term, and  $\alpha > 0$  is the regularization factor. Finally,  $F$  is the following function:

$$F(I_0, I_1; u, v)(x, y) = I_1(x + u(x, y), y + v(x, y)) - I_0(x, y). \quad (4.3)$$

### 4.2.3 Regularization

The following regularization terms were used in our numerical experiments:

$$R_0(u, v) = \|u\|^2 + \|v\|^2, \quad (4.4)$$

$$R_1(u, v) = \|\nabla u\|^2 + \|\nabla v\|^2 = \|\partial_x u\|^2 + \|\partial_y u\|^2 + \|\partial_x v\|^2 + \|\partial_y v\|^2, \quad (4.5)$$

$$R_{div}(u, v) = \|\operatorname{div}(u, v)\|^2 = \|\partial_x u + \partial_y v\|^2, \quad (4.6)$$

$$R_{curl}(u, v) = \|\operatorname{curl}(u, v)\|^2 = \|\partial_y u - \partial_x v\|^2, \quad (4.7)$$

$$R_{div/curl}(u, v) = \|\operatorname{div}(u, v)\|^2 + \|\operatorname{curl}(u, v)\|^2 = \|\partial_x u + \partial_y v\|^2 + \|\partial_y u - \partial_x v\|^2, \quad (4.8)$$

$$R_{\nabla div}(u, v) = \|\nabla \operatorname{div}(u, v)\|^2 = \|\partial_{xx}^2 u + \partial_{xy}^2 v\|^2 + \|\partial_{xy}^2 u + \partial_{yy}^2 v\|^2, \quad (4.9)$$

$$\begin{aligned} R_{\nabla div/\nabla curl}(u, v) &= \|\nabla \operatorname{div}(u, v)\|^2 + \|\nabla \operatorname{curl}(u, v)\|^2 \\ &= \|\partial_{xx}^2 u + \partial_{xy}^2 v\|^2 + \|\partial_{xy}^2 u + \partial_{yy}^2 v\|^2 + \|\partial_{xy}^2 u - \partial_{xx}^2 v\|^2 + \|\partial_{yy}^2 u - \partial_{xy}^2 v\|^2. \end{aligned} \quad (4.10)$$

In all the cases, we can write  $R(u, v) = \|S(u, v)\|^2$ , where  $S$  is a linear operator. Some scalar coefficients have also been considered in order to weight the different terms of a given regularization.

### 4.2.4 Multi-grid approach and optimization

The minimization of the cost function  $J$  is performed in nested subspaces:

$$\mathcal{C}_{16} \subset \mathcal{C}_8 \subset \mathcal{C}_4 \subset \mathcal{C}_2 \subset \mathcal{C}_1, \quad (4.11)$$

where  $\mathcal{C}_q$  is the set of admissible displacement fields at the scale  $q$ , containing piecewise affine vector fields with respect to each space variable, on squares of size  $q \times q$  pixels.

The difference with hierarchical techniques issued from the optical flow family (see e.g. [134, 146]) is that we do not linearize the cost function. This should help to find large displacements, where the domain of linearity of the luminance function is not valid.

The space  $\mathcal{C}_{16}$  is typically of small dimension, hence the minimization of  $J$  on  $\mathcal{C}_{16}$  is fast and robust when a zero vector field is used as initial guess. The optimal vector field obtained at a given scale in the space  $\mathcal{C}_q$  is used as initial guess to find

the minimum at the finer scale in the space  $\mathcal{C}_{q/2}$ . This process is iteratively repeated, until an optimal solution is identified on the finest grid.

All the optimizations of the nonlinear cost function are performed by a Gauss-Newton method. When an initial guess  $(u^0, v^0)$  is given (a constant null field, in our experiments), the  $k$ -th iteration read

$$(u^k, v^k) := (u^{k-1}, v^{k-1}) + (du^k, dv^k), \quad (4.12)$$

where  $(du^k, dv^k)$  solves

$$(DF^T DF + \alpha S^T S)(du, dv) = -DF^T F - \alpha S^T S(u, v), \quad (4.13)$$

where  $F = F(I_0, I_1; u^{k-1}, v^{k-1})$  is the error,  $DF = DF(I_0, I_1; u^{k-1}, v^{k-1})$  is the Jacobian matrix of the error, and  $S$  is the linear operator associated to the regularization term.

Another innovation of the present work is the efficient computation of the product  $DF^T DF$  of the Jacobian of the first term of the cost function (4.2) by its transpose. This efficient computation comes from the observation that this matrix is sparse and can be assembled like a finite-element matrix using one loop over the data.

Let  $V \in \mathcal{C}_q$  be a vector field. Let  $(\mathbf{e}_i)$  be the canonical orthonormal basis of  $\mathcal{C}_q$ , containing vector fields that are equal to 0 at every but one control point, where the vector field is directed along the horizontal or vertical axis. The  $(k, l)^{\text{th}}$  coefficient of the matrix  $DF^T DF$  is

$$(DF^T DF)_{k,l} = (DF^T DF \mathbf{e}_k | \mathbf{e}_l) = (DF \mathbf{e}_k | DF \mathbf{e}_l)_{L^2(\Omega)}. \quad (4.14)$$

Since the elementary displacements  $\mathbf{e}_k$  are non zero at only one control point, the matrix  $DF^T DF$  has a sparse structure. If we consider the following formulation of the jacobian matrix:

$$DF(u, v).d(x, y) = \nabla I_1(x + u(x, y), y + v(x, y)).d(x, y), \quad (4.15)$$

then the matrix  $DF^T DF$  can be assembled like a finite-element matrix:

$$\begin{aligned} DF^T DF &= \sum_{k,l} (DF^T DF)_{k,l} \mathbf{e}_k \otimes \mathbf{e}_l \\ &= \sum_{k,l} \int_{\Omega} (\nabla I_1(x') | \mathbf{e}_k(x)) (\nabla I_1(x') | \mathbf{e}_l(x)) \mathbf{e}_k \otimes \mathbf{e}_l dx \\ &= \sum_{k,l} \sum_{R \in \mathcal{R}_q} \int_R (\nabla I_1(x') | \mathbf{e}_k(x)) (\nabla I_1(x') | \mathbf{e}_l(x)) \mathbf{e}_k \otimes \mathbf{e}_l dx \\ &= \sum_{R \in \mathcal{R}_q} \sum_{k,l} \int_R (\nabla I_1(x') | \mathbf{e}_k(x)) (\nabla I_1(x') | \mathbf{e}_l(x)) \mathbf{e}_k \otimes \mathbf{e}_l dx, \end{aligned} \quad (4.16)$$

where we write  $x' = x + V(x)$ , and where  $\mathcal{R}_q$  represents the set of all squares of the  $q \times q$  grid. There are 8 quantities of the form  $(\nabla I_1(x + V(x)) | \mathbf{e}_k(x))$  to be computed



for each element of  $\mathcal{R}_q$ , and the matrix  $DF^T DF$  can be assembled by reading once the data. The vector field  $DF^T F$  can be assembled rapidly in a similar way, and the term  $S^T S$  is easy to compute.

Finally, equation (4.13) is solved using a conjugate gradient method without preconditioning.

#### 4.2.5 Quality estimate

An estimation of the quality of our results is highly motivated by the application that we presented in the introduction, namely data assimilation. A well known issue and a crucial point in data assimilation is the knowledge of the statistics of observation errors. Hence, we propose here an estimation of the quality of the pseudo-observations identified by our algorithm.

We propose a normalized quality estimate, where the quality of the motion depends on the ratio between the grey-level differences before and after registration:

$$e(I_0, I_1; u, v)(x, y) = 1 - \frac{|I_1(x + u(x, y); y + v(x, y)) - I_0(x, y)|}{|I_1(x, y) - I_0(x, y)|} \quad (4.17)$$

if the denominator is non-zero, otherwise we define  $e(I_0, I_1; u, v) = 0$ .

We can clearly see that if the two images were quite different on a pixel  $(x, y)$  before the process, and much less different after, then the estimate  $e$  is nearly equal to 1. We will further see that in some regions of the images, there is almost no signal, and then the two images are equal, both before and after the identification process. This leads to an estimate  $e$  equal to 0, not because the identified velocity is wrong, but because we cannot quantify whether it is good or not. This estimator is provided by our algorithm, so that it can be used along with the identified velocity fields in data assimilation experiments.

### 4.3 Numerical experiments [23]

In this section, we briefly present the numerical experiments that have been carried out on both simulated and real data. We refer to [23] for the results of these experiments.

#### 4.3.1 Simulated data

We first try our algorithm on simulated data. We consider a basic model, the shallow water model (or Saint-Venant's equations), representing quite well the temporal evolution of geophysical flows. This model is detailed in section 3.3.3 (with different parameters), or in [23].

This model is then coupled with an advection-diffusion equation, modeling the fact that the concentration of a passive tracer is transported by the fluid velocity:

$$\partial_t c + u \partial_x c + v \partial_y c = 0, \quad (4.18)$$

where  $c$  is the concentration of the passive tracer (e.g. chlorophyll in oceans). We also add to this equation an initial condition  $c(t = 0)$ . We consider then a trajectory of this shallow water model coupled with a concentration equation, from which a concentration image is extracted every 100 time steps (in order to reproduce the time sampling of the satellite images).

Two consecutive images are extracted from these simulated data, and we apply our algorithm to these two images, with the aim of identifying the entire velocity field. As shown in [23], our algorithm quickly extracts very accurate velocity fields. This is mainly due to the combination of a multi-grid approach and an efficient optimization scheme (no a priori information and no linearization). The registration between two images is almost perfect after a few iterations, and the identified velocity field reproduces very well the global structure of the true velocity (a rotating vortex in a translation field in our experiments).

Concerning the regularization, we can note that the best results correspond to the  $R_1$  norm (see equation (4.5)). The most physical regularization is probably  $R_{div}$  (see equation (4.6)), as we expect a null divergence velocity field in geophysical flows. But the decrease of the cost function is not as good as for some other regularizations. Considering that the images are acquired every 100 time steps only, the velocity we want to identify between these two images is a time Lagrangian integration of many instantaneous velocities, and it cannot have a divergence equal to zero.

We also present an interesting application of the identification process. Assume that we have a particular object in the first image, e.g. a characteristic structure, that has been manually selected. In our case, we can identify one specific vortex. We can then limit the identification process to a region around this object. This region is propagated from one pair of images to the next one by the mean of the identified velocity. This allows us to track this object in time, in a fully automatic way.

### 4.3.2 Experimental data

We have then considered data extracted from several experiments on the Coriolis rotating platform [75]. A large rotating turntable (diameter: 13 meters) allows us to reproduce the oceanic or atmospheric flows. Depending on the experiments, either some colorant or particles are inserted in the water as the platform rotates, and among the various measurement devices, a camera takes pictures of the experiment [95].

Several test cases have been studied, corresponding to either small or large acquisition times between two consecutive images. In all these different situations, the global structure of the displacement field matches perfectly with the real displacement of the fluid. The multi-grid approach has been compared with the standard approach, in which the minimization is directly performed on the fine grid. Both the computation time and the quality of the results are degraded.

These results have been compared with those produced by the PIV (Particle Imaging Velocimetry) method. PIV is the reference method for the extraction of velocity fields in geophysics and fluid mechanics. The results are qualitatively equivalent, in the sense that the identified fields look alike. However, our algorithm represents

two main improvements: the computation time, allowing us to extract velocity fields from several hundreds of images in a relatively short time; and the preciseness of the results, as we extract one velocity vector for each pixel of the image, while the PIV method usually gives only one vector every nearly  $10 \times 10$  pixels. This allows us to track the evolution of very small structures.

## 4.4 Conclusions

We presented in this section an algorithm to estimate the motion between two images. This algorithm is based on the constant brightness assumption. A multiscale approach allows us to perform a minimization of the cost function in nested subspaces, the Jacobian matrix of the cost function being rapidly assembled at each scale using a finite element method. The coarse estimation allows one to avoid local minima, while the fine scales give more precise details. Several regularization terms are discussed, and it appears that the  $L^2$  norm of the gradient gives reliable results.

The results of this algorithm on both simulated data and real fluid flows are presented, and they are encouraging, both from their computational efficiency and from the quality of the estimated motion. Our algorithm has also been tested on full high-resolution movies provided by the Coriolis platform, confirming the efficiency of the proposed method.

As previously explained, the extracted velocity fields can be viewed as pseudo-observations of the fluid velocity, and the next step will be to consider the assimilation of these data. However, because of the time sampling of the images, these fields correspond to Lagrangian velocities, and a Lagrangian data assimilation method is then required. Note that if the time between the acquisition of two images is small, then the identified (or apparent) velocity can be directly assimilated as a standard Eulerian velocity.



## Chapter 5

# General conclusions and perspectives

We presented in this work several algorithms for solving image processing and data assimilation problems. All these algorithms are robust, easy to implement, fast and powerful. This work has been essentially motivated by the applications of such problems. In the case of image processing, one of these constraints could be to be able to process movies in real time or large images in a negligible time. For data assimilation problems, the goal is to assimilate a huge amount of data in a given time, bounded by some operational constraints (e.g. of providing some short or medium-range weather forecasts in a given time).

It seemed crucial to us to develop some algorithms that are quite far from the state of the art in both image processing and data assimilation. For instance, the topological gradient has been introduced in the image processing field, providing a more global information than the standard gradient of the image. Also, the data assimilation community is currently split into two parts: variational and sequential methods. The first ones (e.g. the 4D-VAR algorithm) require a huge human cost for the implementation of the adjoint code, and the second ones (e.g. Kalman filters) rely on the very precise knowledge of the error statistics. Thus, we made the choice of introducing an algorithm at the interface of these two categories, in order to combine the advantages without the main drawbacks.

There are still many perspectives in these research fields, because some problems have not been studied yet, and also because our algorithms can still be improved. For instance, all the algorithms introduced for image processing problems are based on the edge detection by topological gradient. It seems interesting to define more than two conductivity values, in order to identify more than one edge set, as the edges do not correspond to the same level of discontinuities. In data assimilation, the back and forth nudging algorithm can also be improved, for instance by automatically decreasing or increasing the gain coefficients with the iterations, in order to keep a relative equilibrium between the physical model and the feedback to the observations.

As long term perspectives in image processing, we can cite for instance the compression and deblurring problems, for which it should also be possible to define an approach by topological asymptotic analysis. Also, an interesting challenge in data assimilation is to test the back and forth nudging algorithm on a primitive equation model with real data, in order to study the behaviour of this algorithm in real conditions.

# List of publications

## 1. Publications linked to the PhD thesis

### PhD thesis

- [1] D. Auroux. *Étude de quelques méthodes d'assimilation de données pour l'environnement*. PhD thesis, Université de Nice Sophia-Antipolis, Décembre 2003.

### Peer-reviewed journal articles

- [2] F. Veersé, D. Auroux, and M. Fisher. Limited-memory BFGS diagonal preconditioners for a data assimilation problem in meteorology. *Optim. Engineer.*, 1.3:323–339, 2000.
- [3] D. Auroux. Generalization of the dual variational data assimilation algorithm to a nonlinear layered quasi-geostrophic ocean model. *Inverse Problems*, 23:2485–2503, 2007.

### Book chapters

- [4] D. Auroux and J. Blum. *Data assimilation methods for an oceanographic problem*, volume XVI of *Lecture Notes - Mathematics in Industry*, pages 179–194. Springer-Verlag, 2004.

### Notes

- [5] D. Auroux and J. Blum. A dual data assimilation method for a layered quasi-geostrophic ocean model. *Rev. R. Acad. Cien. Serie A Mat.*, 96(3):316–320, 2002.

### Research reports

- [6] F. Veersé and D. Auroux. Some numerical experiments on scaling and updating L-BFGS diagonal preconditioners. Technical Report 3858, INRIA, 2000.

### Peer-reviewed national proceedings

- [7] D. Auroux. Assimilation variationnelle de données océanographiques - approches primale et duale. 9(2), pages 147–159. Annales Mathématiques Blaise Pascal, 2002.

## 2. Publications after the PhD thesis

### Peer-reviewed journal articles

- [8] D. Auroux. Several data assimilation methods for geophysical problems. *Ind. J. Pure Appl. Math.*, 37(1):41–58, 2006.
- [9] D. Auroux and M. Masmoudi. A one-shot inpainting algorithm based on the topological asymptotic analysis. *Comp. Appl. Math.*, 25(2-3):1–17, 2006.
- [10] D. Auroux, L. Jaafar Belaid, and M. Masmoudi. A topological asymptotic analysis for the regularized grey-level image classification problem. *Math. Model. Numer. Anal.*, 41(3):607–625, 2007.
- [11] D. Auroux and J. Blum. A nudging-based data assimilation method for oceanographic problems: the Back and Forth Nudging (BFN) algorithm. *Nonlin. Proc. Geophys.*, 15:305–319, 2008.
- [12] D. Auroux. From restoration by topological gradient to medical image segmentation via an asymptotic expansion. *Math. Comput. Model.*, 2008. In press.
- [13] D. Auroux and M. Masmoudi. Image processing by topological asymptotic expansion. *J. Math. Imaging Vision*, 2008. In press.
- [14] D. Auroux. The back and forth nudging algorithm applied to a shallow water model, comparison and hybridization with the 4D-VAR. *Int. J. Numer. Methods Fluids*, 2008. In press.
- [15] M. Masmoudi, D. Auroux, and Y. Parte. The state of the art in collaborative design. *Comput. Fluid Dyn. J.*, 2008. Accepted for publication.

### Book chapters

- [16] D. Auroux, M. Masmoudi, and L. Jaafar Belaid. *Image restoration and classification by topological asymptotic expansion*. Variational Formulations in Mechanics: Theory and Applications, E. Taroco, E.A. de Souza Neto and A.A. Novotny (Eds). CIMNE, Barcelona, Spain, 2006.
- [17] D. Auroux, J. Clément, J. Hermetz, M. Masmoudi, and Y. Parte. *État de l'art et nouvelles tendances en conception collaborative*. Optimisation multidisciplinaire en mécanique. Hermes Science Publishing, London, 2008.



## Notes

- [18] D. Auroux and J. Blum. Back and forth nudging algorithm for data assimilation problems. *C. R. Acad. Sci. Paris, Ser. I*, 340:873–878, 2005.

## Peer-reviewed international proceedings

- [19] G. Caille, Y. Cailloce, C. Guiraud, D. Auroux, T. Touya, and M. Masmoudi. Large multibeam array antennas with reduced number of active chains. In *Proc. EuCAP 2007 - Antennas and Propagation*, pages 1–9, 2007.
- [20] T. Touya and D. Auroux. Control and topological optimization of a large multibeam array antenna. In A. Daryoush, editor, *Proc. ARP 2008 - Antennas, Radar, and Wave Propagation*. ACTA Press, 2008.
- [21] D. Auroux, P. Bansart, and J. Blum. An easy-to-implement and efficient data assimilation method for the identification of the initial condition: the Back and Forth Nudging (BFN) algorithm. *J. Physics Conf. Series*, 2008. In press.

## Preprints

- [22] D. Auroux, L. Jaafar Belaid, and B. Rjaibi. Application of the topological gradient method to color image restoration. Submitted, 2008.
- [23] D. Auroux and J. Fehrenbach. Identification of velocity fields for geophysical fluids from a sequence of images. Submitted, 2008.
- [24] D. Auroux and M. Nodet. The back and forth nudging algorithm for data assimilation problems: theoretical results on transport equations. Submitted, 2008.
- [25] D. Auroux and S. Bonnabel. Symmetry-preserving nudging: theory and application to a shallow water model. Submitted, 2008.
- [26] Y. Ahipo, D. Auroux, L. Cohen, and M. Masmoudi. A hybrid scheme for contour detection and completion based on topological gradient and fast marching algorithms - an application to image inpainting and segmentation. Preprint, 2008.



# References

- [27] R. Adrian. Particle imaging techniques for experimental fluid mechanics. *Ann. Rev. Fluid Mech.*, 23:261–304, 1991.
- [28] N. Aghannan and P. Rouchon. On invariant asymptotic observers. In *Proc. of the 41st IEEE Conf. on Decision and Control*, volume 2, pages 1479–1484, 2002.
- [29] N. Aghannan and P. Rouchon. An intrinsic observer for a class of lagrangian systems. *IEEE Trans. Autom. Contr.*, 48(6):936–945, 2003.
- [30] G. Alessandrini, E. Beretta, and S. Vessela. Determining linear cracks by boundary measurements: Lipschitz stability. *SIAM J. Math. Anal.*, 27(2):361–375, 1996.
- [31] G. Alessandrini and A. Diaz Valenzuela. Unique determination of multiple cracks by two measurements. *SIAM J. Control Optim.*, 34(3):913–921, 1996.
- [32] G. Allaire. Shape optimization by the homogenization method. *Applied Mathematical Sciences*, 146, Springer-Verlag, 2002.
- [33] G. Allaire, F. Jouve, and A. M. Taoder. A level set method for shape optimization. *C. R. Acad. Sci. Paris Ser. I*, 334:1125–1130, 2002.
- [34] G. Allaire and R. Kohn. Optimal design for minimum weight and compliance in plane stress using extremal microstructures. *Eur. J. Mech. A Solids*, 12:839–878, 1993.
- [35] L. Alvarez, C. Castaño, M. Garcia, K. Krissian, L. Mazorra, A. Salgado, and J. Sanchez. *A variational approach for 3D motion estimation of incompressible PIV flows*, pages 837–847. Springer, Berlin, 2007.
- [36] H. Ammari, M. S. Vogelius, and D. Volkov. Asymptotic formulas for perturbations in the electromagnetic fields due to the presence of inhomogeneities of small diameter ii - the full maxwell equations. *J. Math. Pures Appl.*, 80(8):769–814, 2001.
- [37] S. Amstutz, I. Horchani, and M. Masmoudi. Crack detection by the topological gradient method. *Control and Cybernetics*, 34(1):119–138, 2005.

- [38] S. Amstutz, M. Masmoudi, and B. Samet. The topological asymptotic for the helmoltz equation. *SIAM J. Control Optim.*, 42(5):1523–1544, 2003.
- [39] P. Ananden. A computational framework and an algorithm for the measurement of visual motion. *Int. J. Comput. Vis.*, 2:283–310, 1989.
- [40] S. Andrieux and A. Ben Abda. Identification of planar cracks by complete overdetermined data: inversion formulae. *Inverse Problems*, 12:553–563, 1996.
- [41] M. Arnold and B. N. Datta. Single-input eigenvalue assignment algorithms: A close look. *SIAM J. Matrix Anal. Appl.*, 19(2):444–467, 1998.
- [42] G. Aubert and J.-F. Aujol. Optimal partitions, regularized solutions, and application to image classification. *Applicable Analysis*, 84(1):15–35, 2005.
- [43] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2001.
- [44] G. Aubert and L. Vese. A variational method in image recovery. *SIAM J. Numer. Anal.*, 34(5):1948–1979, 1997.
- [45] J.-F. Aujol, G. Aubert, and L. Blanc-Féraud. Wavelet-based level set evolution for classification of textured images. *IEEE Trans. Image Proc.*, 12(12):1634–1641, 2003.
- [46] J.-W. Bao and R. M. Errico. An adjoint examination of a nudging method for data assimilation. *Month. Weather Rev.*, 125:1355–1373, 1997.
- [47] S. Beauchemin and J. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.
- [48] Z. Belhachmi and D. Bucur. Stability and uniqueness for the crack identification problem. *SIAM J. Control Optim.*, 46(1):253–263, 2007.
- [49] A. Ben Abda, H. Ben Ameer, and M. Jaoua. Identification of 2d cracks by elastic boundary measurements. *Inverse Problems*, 15:67–77, 1999.
- [50] A. Ben Abda, M. Kallel, J. Leblond, and J.-P. Marmorat. Line-segment cracks recovery from incomplete boundary data. *Inverse Problems*, 18:1057–1077, 2002.
- [51] M. Bendsoe. Optimal topology design of continuum structure: an introduction. Technical report, Department of Mathematics, Technical University of Denmark, Lyngby, Denmark, 1996.
- [52] M. Bendsoe and P. Kikuchi. Generating optimal topologies in structural design using an homogenisation method. *Comput. Methods Appl. Mech. Engin.*, 71:197–224, 1988.
- [53] A. F. Bennett. *Inverse Modeling of the Ocean and Atmosphere*. Cambridge University Press, Cambridge, 2002.

- [54] M. Berthod, Z. Kato, S. Yu, and J. Zerubia. Bayesian image classification using markov random fields. *Image Vision Comput.*, 14(4):285–293, 1996.
- [55] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.
- [56] E. Blayo, S. Durbiano, P. A. Vidard, and F.-X. Le Dimet. *Reduced order strategies for variational data assimilation in oceanic models*. Springer-Verlag, 2003.
- [57] E. Blayo, J. Verron, and J.-M. Molines. Assimilation of TOPEX/POSEIDON altimeter data into a circulation model of the North Atlantic. *J. Geophys. Res.*, 99(C12):24691–24705, 1994.
- [58] S. Bonnabel, Ph. Martin, and P. Rouchon. Non-linear symmetry-preserving observers on Lie groups. *IEEE Trans. Autom. Contr.*, 2008. accepted for publication.
- [59] S. Bonnabel, Ph. Martin, and P. Rouchon. Symmetry-preserving observers. *IEEE Trans. Automat. Contr.*, 2008. In press.
- [60] F. Bonnans and P. Rouchon. *Commande et optimisation de systèmes dynamiques*. Les Éditions de l'École Polytechnique, Palaiseau, 2005.
- [61] C. A. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Trans. Image Proc.*, 3:162–177, 1994.
- [62] F. Bouttier and G. Kelly. Observing system experiments with the ECMWF data assimilation system. *Quart. J. Roy. Meteor. Soc.*, 127:1469–1488, 2001.
- [63] M. Bruhl, M. Hanke, and M. Pidcock. Crack detection using electrostatic measurements. *Math. Model. Numer. Anal.*, 35:595–605, 2001.
- [64] K. Bryan and M. S. Vogelius. A review of selected works on crack identification. In *Proc. IMA workshop on Geometric Methods in Inverse Problems and PDE Control*, August 2001.
- [65] A.-P. Calderón. On an inverse boundary value problem. In *Seminar on Numerical Analysis and Its Applications to Continuum Physics*, pages 65–73, Rio de Janeiro, Brasil, 1980. Soc. Brasil. Mat.
- [66] F. Catté, T. Coll, P.-L. Lions, and J.-M. Morel. Image selective smoothing and edge detection by non linear diffusion. *SIAM J. Numer. Anal.*, 29:182–193, 1992.
- [67] T. Chan and J. Shen. Mathematical models for local deterministic inpaintings. Technical Report 00-11, UCLA CAM, March 2000.
- [68] T. Chan and J. Shen. Non-texture inpainting by curvature-driven diffusions (ccd). Technical Report 00-35, UCLA CAM, September 2000.

- [69] T. Chan and J. Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.
- [70] P. Charbonnier. *Reconstruction d'images: régularisation avec prise en compte des discontinuités*. PhD thesis, Université de Nice Sophia-Antipolis, 1994.
- [71] L. D. Cohen. Multiple contour finding and perceptual grouping using minimal paths. Technical report, Les Cahiers du CEREMADE, Université Paris Dauphine, 2001.
- [72] L. D. Cohen. *Minimal paths and fast marching methods for image analysis*. Mathematical models in computer vision: the handbook, N. Paragios, Y. Chen and O. Faugeras (Eds). Springer, 2005.
- [73] L. D. Cohen and R. Kimmel. Global minimum for active contour models: a minimal path approach. *Int. J. Computer Vision*, 24(1):57–78, 1997.
- [74] J. D. Cole. On a quasilinear parabolic equation occurring in aerodynamics. *Quart. Appl. Math.*, 9(3):225–236, 1951.
- [75] Coriolis rotating platform, website <http://www.coriolis-legi.org/>.
- [76] T. Corpetti, E. Mémin, and P. Pérez. Dense estimation of fluid flows. *IEEE Trans. Pattern Anal. Mach. Intel.*, 24:365–380, 2002.
- [77] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Volume II*. Wiley-Interscience, 1962.
- [78] P. Courtier. Dual formulation of four-dimensional variational assimilation. *Q. J. R. Meteorol. Soc.*, 123:2449–2461, 1997.
- [79] A. Cuzol, P. Hellier, and E. Mémin. A low dimensional fluid motion estimator. *Int. J. Comput. Vis.*, 75:329–349, 2007.
- [80] B. N. Datta. An algorithm to assign eigenvalues in a hessenberg matrix: Single-input case. *IEEE Trans. Automatic Control*, 32(5):414–417, 1987.
- [81] R. Dautray and J.-L. Lions. *Analyse mathématique et calcul numérique pour les sciences et les techniques*. Collection CEA. Masson, Paris, 1987.
- [82] T. Deschamps and L. D. Cohen. Minimal path in 3D images and application to virtual endoscopy. Technical report, Les Cahiers du CEREMADE, Université Paris Dauphine, 2000.
- [83] S. Di Zenzo. A note on the gradient of a multi-image. *Comput. Vision Graph. Image Proc.*, 33:116–125, 1986.
- [84] J. Dicker. *Fast marching methods and level set methods: an implementation*. PhD thesis, University of British Columbia, 2006.

- [85] F. Dubois, N. Petit, and P. Rouchon. Motion planing and nonlinear simulations for a tank containing a fluid. In *European Control Conference, Karlsruhe*, 1999.
- [86] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Proc.*, 15(12):3736–3745, 2006.
- [87] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *J. Appl. Comput. Harmonic Anal.*, 19:340–358, 2005.
- [88] H.W. Engl and C.W. Groetsch. *Inverse and Ill Posed Problems*. Academic Press, New York, 1987.
- [89] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, 1998.
- [90] G. Evensen and P. J. van Leeuwen. An ensemble Kalman smoother for nonlinear dynamics. *Mon. Wea. Rev.*, 128:1852–1867, 1999.
- [91] J. Fehrenbach and M. Masmoudi. A fast algorithm for image registration. *C. R. Acad. Sci. Paris, Ser. I*, 346:593–598, 2008.
- [92] M. Fisher and P. Courtier. Estimating the covariance matrix of analysis and forecast error in variational data assimilation. Technical Report 220, ECMWF, 1995.
- [93] J. M. Fitzpatrick. A method for calculating velocity in time dependent images based on the continuity equation. In *Proc. Conf. Comp. Vision Pattern Rec.*, pages 78–81, San Francisco, USA, 1985.
- [94] J. M. Fitzpatrick. The existence of geometrical density-image transformations corresponding to object motion. *Comput. Vis. Grap. Imag. Proc.*, 44:155–174, 1988.
- [95] J. B. Flor and I. Eames. Dynamics of monopolar vortices on the beta plane. *J. Fluid Mech.*, 456:353–376, 2002.
- [96] A. Friedman and M. S. Vogelius. Determining cracks by boundary measurements. *Indiana Univ. Math. J.*, 38(3):527–556, 1989.
- [97] A. Friedman and M. S. Vogelius. Identification of small inhomogeneities of extreme conductivity by boundary measurements: a theorem of continuous dependence. *Arch. Rational Mech. Anal.*, 105(4):299–326, 1989.
- [98] S. Garreau, P. Guillaume, and M. Masmoudi. The topological asymptotic for pde systems: The elasticity case. *SIAM J. Control Optim.*, 39(6):1756–1778, 2001.

- [99] M. Ghil. Meteorological data assimilation for oceanographers. part i: description and theoretical framework. *Dyn. Atmos. Oceans*, 13:171–218, 1989.
- [100] M. Ghil and P. Manalotte-Rizzoli. Data assimilation in meteorology and oceanography. *Adv. Geophys.*, 23:141–265, 1991.
- [101] J.-Ch. Gilbert and C. Lemaréchal. Some numerical experiments with variable-storage quasi-Newton algorithms. *Math. Prog.*, 45:407–435, 1989.
- [102] J. Giroire and J.-C. Nédélec. Numerical solution of an exterior neumann problem using a double layer potentiel. *Math. Comput.*, 32(144):973–990, 1978.
- [103] GOES geostationnary satellite server <http://www.goes.noaa.gov/>.
- [104] P. Guillaume and K. Sididris. The topological asymptotic expansion for the dirichlet problem. *SIAM J. Control Optim.*, 41(4):1042–1072, 2002.
- [105] P. Guillaume and K. Sididris. The topological sensitivity and shape optimization for the stokes equations. *SIAM J. Control Optim.*, 43(1):1–31, 2004.
- [106] J. Hoke and R. A. Anthes. The initialization of numerical models by a dynamic initialization technique. *Month. Weather Rev.*, 104:1551–1556, 1976.
- [107] W. R. Holland. The role of mesoscale eddies in the general circulation of the ocean. *J. Phys. Ocean.*, 8(3):363–392, 1978.
- [108] E. Hopf. The partial differential equation  $u_t + uu_x = \mu u_{xx}$ . *Comm. Pure Appl. Math.*, 3:201–230, 1950.
- [109] B. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [110] E. Huot, T. Isambert, I. Herlin, J.-P. Berroir, and G. Korotaev. Data assimilation of satellite images within an oceanographic circulation model. In *Proc. Int. Conf. Acoustics Speech Sign. Proc.*, Toulouse, France, 2006.
- [111] T. Isambert, I. Herlin, and J.-P. Berroir. Fast and stable vector spline method for fluid flow estimation. In *Proc. Int. Conf. Image Proc.*, pages 505–508, San Antonio, USA, 2007.
- [112] L. Jaafar-Belaid, M. Jaoua, M. Masmoudi, and L. Siala. Image restoration and edge detection by topological asymptotic expansion. *C. R. Acad. Sci., Ser. I*, 342(5):313–318, 2006.
- [113] S. Jiang and M. Ghil. Tracking nonlinear solutions with simulated altimetric data in a shallow-water model. *J. Phys. Oceanogr.*, 27(1):72–95, 1997.
- [114] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME - J. Basic Engin.*, 82:35–45, 1960.



- [115] E. Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2003.
- [116] E. Kalnay, S. Ki Park, Z.-X. Pu, and J. Gao. Application of the quasi-inverse method to data assimilation. *Month. Weather Rev.*, 128:864–875, 2000.
- [117] A. Khludnev and V. Kovtunenکو. *Analysis of cracks in solids*. WIT Press, Southampton-Boston, 2000.
- [118] R. Kohn and M. Vogelius. Relaxation of a variational method for impedance computed tomography. *Comm. Pure Appl. Math.*, 40(6):745–777, 1987.
- [119] V. Komornik and P. Loreti. *Fourier Series in Control Theory*. Springer Monographs in Mathematics. Springer-Verlag, New York, 2005.
- [120] S. Kubo and K. Ohji. *Inverse problems and the electric potential computed tomography method as one of their application*. Mechanical Modeling of New Electromagnetic Materials. Elsevier Science Publisher, 1990.
- [121] I. Larrabide, R. A. Feijóo, A. A. Novotny, E. Taroco, and M. Masmoudi. An image segmentation method base on a discrete version of the topological derivative. In *Proc. 6th World Congress on Structural and Multidisciplinary Optimization*, Brazil, June 2005.
- [122] R. Larsen, K. Conradsen, and B. K. Ersboll. Estimation of dense image flow fields in fluids. *IEEE Trans. Geoscience Remote Sensing*, 36(1):256–264, 1998.
- [123] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus*, 38A:97–110, 1986.
- [124] Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Trans. Image Proc.*, 5:987–995, 1996.
- [125] R. L. Lillie. *Whole Earth Geophysics: an introduction textbook for geologists and geophysicists*. Prentice Hall, 1999.
- [126] A. C. Lorenc, R. S. Bell, and B. Macpherson. The meteorological office analysis correction data assimilation scheme. *Quart. J. Royal Meteor. Soc.*, 117:59–89, 1991.
- [127] E. N. Lorenz. Deterministic non periodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.
- [128] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, 1981.
- [129] D. Luenberger. Observers for multivariable systems. *IEEE Trans. Autom. Contr.*, 11:190–197, 1966.

- [130] B. Luong, J. Blum, and J. Verron. A variational method for the resolution of a data assimilation problem in oceanography. *Inverse Problems*, 14:979–997, 1998.
- [131] W. H. Lyne, R. Swinbank, and N. T. Birch. A data assimilation experiment and the global circulation during the fgge special observing periods. *Quart. J. Roy. Meteor. Soc.*, 108:575–594, 1982.
- [132] J. Ma, A. Antoniadis, and F.-X. Le Dimet. Curvlets-based snake for multiscale detection and tracking of geophysical fluids. *IEEE Trans. Geoscience Remote Sensing*, 45(1):3626–3638, 2006.
- [133] M. Masmoudi. *The topological asymptotic*, volume 16 of *Computational Methods for Control Applications*, R. Glowinski, H. Karawada, and J. Périaux (Eds.), pages 53–72. GAKUTO Internat. Ser. Math. Sci. Appl., Tokyo, Japan, 2001.
- [134] E. Mémin and P. Perez. Optical flow estimation and object-based segmentation with robust techniques. *IEEE Trans. Image Proc.*, 7(5):703–719, 1998.
- [135] Y. Michel and F. Bouttier. Automatic tracking of dry intrusions on satellite water vapour imagery and model output. *Quart. J. Roy. Meteor. Soc.*, 132:2257–2276, 2006.
- [136] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42(5):577–685, 1989.
- [137] N. Nishimura and S. Kobayashi. A boundary integral equation method for an inverse problem related to crack detection. *Int. J. Num. Methods Engrg.*, 32:1371–1387, 1991.
- [138] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Comput.*, 35:773–782, 1980.
- [139] P. J. Olver. *Equivalence, Invariants, and Symmetry*. Cambridge University Press, 1995.
- [140] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. Comput. Vision*, 46(3):223–247, 2002.
- [141] T. Pavlidis and Y.-T. Liow. Integrating region growing and edge detection. *IEEE Trans. Pattern Anal. Mach. Intel.*, 12(3):225–233, 1990.
- [142] P. Perona and J. Malik. Scale space and edge detection detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intel.*, 12:629–639, 1990.
- [143] C. Pires, R. Vautard, and O. Talagrand. On extending the limits of variational assimilation in nonlinear chaotic systems. *Tellus*, 48A:96–121, 1996.

- [144] F. Rabier, H. Järvinen, E. Klinker, J.-F. Mahfouf, and A. Simmons. The ECMWF operational implementation of four-dimensional assimilation. i: Experimental results with simplified physics. *Q. J. R. Meteorol. Soc.*, 126:1143–1170, 2000.
- [145] N. Rawlinson and M. Sambridge. The fast marching method: an effective tool for tomographic imaging and tracking multiple phases in complex layered media. *Explor. Geophys.*, 36:341–350, 2005.
- [146] P. Ruhnau, T. Kohlberger, C. Schnörr, and H. Nobach. Variational optical flow estimation for particle image velocimetry. *Experiments in Fluids*, 38(1):21–32, 2005.
- [147] D. L. Russell. Controllability and stabilizability theory for linear partial differential equations: recent progress and open questions. *SIAM Rev.*, 20(4):639–739, 1978.
- [148] B. Samet, S. Amstutz, and M. Masmoudi. The topological asymptotic for the helmholtz equation. *SIAM J. Control Optim.*, 42(5):1523–1544, 2003.
- [149] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia. A level set model for image classification. *Int. J. Comput. Vision*, 40(3):187–197, 2000.
- [150] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia. A variational model for image classification and restoration. *IEEE Trans. Pattern Anal. Mach. Intel.*, 22(5):460–472, 2000.
- [151] F. Santosa. A level-set approach for inverse problems involving obstacles. *ESAIM Control Optim. Calc. Var.*, 1:17–33, 1996.
- [152] F. Santosa and M. Vogelius. A computational algorithm to determine cracks from electrostatic boundary measurements. *Int. J. Eng. Sci.*, 29:917–937, 1991.
- [153] G. Sapiro. *Differential Equations and Image Analysis*. Cambridge University Press, Cambridge, 2001.
- [154] A. Schumacher. *Topologieoptimisierung von Bauteilstrukturen unter Verwendung von Topchpositionierungskriterien*. PhD thesis, Universität-Gesamthochschule-Siegen, Germany, 1995.
- [155] L. Schwartz. Opérateurs invariants par rotations. Fonctions métaharmoniques. In *Séminaire Schwartz*, volume 2(9), pages 1–5, 1954.
- [156] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, Cambridge, 1999.

- [157] J. Sokolowski and A. Zochowski. On the topological derivative in shape optimization. *SIAM J. Control Optim.*, 37:1241–1272, 1999.
- [158] J. Sokolowski and A. Zochowski. Topological derivatives of shape functionals for elasticity systems. *Int. Ser. Numer. Math.*, 139:231–244, 2002.
- [159] D. R. Stauffer and J. W. Bao. Optimal determination of nudging coefficients using the adjoint equations. *Tellus*, 45A:358–369, 1993.
- [160] D. R. Stauffer and N. L. Seaman. Use of four dimensional data assimilation in a limited area mesoscale model - part 1: Experiments with synoptic-scale data. *Month. Weather Rev.*, 118:1250–1277, 1990.
- [161] O. Talagrand. On the mathematics of data assimilation. *Tellus*, 33A:321–339, 1981.
- [162] O. Talagrand. A study of the dynamics of four-dimensional data assimilation. *Tellus*, 33A:43–60, 1981.
- [163] O. Talagrand. Assimilation of observations, an introduction. *J. Met. Soc. Japan*, 75(1B):191–209, 1997.
- [164] A. Telea and J. van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *Proc. Eurographics IEEE-TCVG Symposium on Visualization*, Vienna, 2002. Springer.
- [165] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Pattern Anal. Mach. Intel.*, 8(4):413–424, 1986.
- [166] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill Posed Problems*. Winston and sons, Washington, 1997.
- [167] E. Trélat. *Contrôle optimal: théorie et applications*. Vuibert, Paris, 2005.
- [168] J. Verron. Altimeter data assimilation into an ocean circulation model: sensitivity to orbital parameters. *J. Geophys. Res.*, 95(C7):443–459, 1990.
- [169] J. Verron, L. Gourdeau, D. T. Pham, R. Murtugudde, and A. J. Busalacchi. An extended Kalman filter to assimilate satellite altimeter data into a nonlinear numerical model of the tropical pacific ocean: Method and validation. *J. Geophys. Res.*, 104:5441–5458, 1999.
- [170] J. Verron and W. R. Holland. Impact de données d’altimétrie satellitaire sur les simulations numériques des circulations générales océaniques aux latitudes moyennes. *Annales Geophysicae*, 7(1):31–46, 1989.
- [171] J. Verron, J.-M. Molines, and E. Blayo. Assimilation of geosat data into a quasigeostrophic model of the north atlantic between 20°n and 50°n: preliminary results. *Oceanol. Acta*, 15(5):575–583, 1992.

- [172] P.-A. Vidard, F.-X. Le Dimet, and A. Piacentini. Determination of optimal nudging coefficients. *Tellus*, 55A:1–15, 2003.
- [173] M. Y. Wang, X. M. Wang, and D. M. Guo. A level set method for structural topology optimization. *Comput. Meth. Appl. Mech. Engin.*, 192:227–246, 2003.
- [174] J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. In *Theoretical Foundations of Computer Vision*, volume 11, pages 221–236, Dagstuhl, Germany, 1994.
- [175] J. Weickert. *Anisotropic diffusion in image processing*. PhD thesis, University of Kaiserslautern, Germany, 1996.
- [176] J. Weickert. Efficient image segmentation using partial differential equations and morphology. *Pattern Recognition*, 34(9):1813–1824, 2001.
- [177] P. Wen, X. Wu, and C. Wu. An interactive image inpainting method based on RBF networks. In *Third International Symposium on Neural Networks*, pages 629–637, 2006.
- [178] T. Zhou, F. Tang, J. Wang, Z. Wang, and Q. Peng. Digital image inpainting with radial basis functions. *J. Image Graphics (In Chinese)*, 9(10):1190–1196, 2004.
- [179] F. Zhu and J. Tian. Modified fast marching methods and level set method for medical image segmentation. *J. X-ray Sci. Tech*, 11(4):193–204, 2003.
- [180] X. Zou, I. M. Navon, and F.-X. Le Dimet. An optimal nudging data assimilation scheme using parameter estimation. *Q. J. R. Meteorol. Soc.*, 118:1163–1186, 1992.





## Résumé:

Dans une première partie, nous avons étudié des problèmes de traitement d'images. Nous avons utilisé l'analyse asymptotique topologique pour la détection des contours d'une image. Cela permet de considérer alors plusieurs applications: restauration/débruitage, classification. L'inpainting est traité d'une façon un peu différente, et la double donnée Dirichlet et Neumann sur le bord du domaine caché permet de reconstruire les contours dans la partie cachée de l'image. Enfin la segmentation peut être traitée comme limite de la classification, en s'appuyant sur des résultats d'analycité de la solution quand on fait tendre un paramètre vers 0. La rapidité de cette méthode permet de traiter ces différents cas en temps réel, y compris pour des films.

Dans une seconde partie, nous avons abordé l'assimilation de données, le but étant d'identifier la condition initiale d'un système à partir d'observations partielles. Nous avons défini un nouvel algorithme, basé sur le "nudging" (méthode de relaxation consistant à rajouter un terme de rappel vers les observations directement dans l'équation afin de tirer la solution vers les observations). En considérant itérativement et alternativement des résolutions du système direct et rétrograde en temps, avec à chaque fois un terme de rappel vers les observations, on peut améliorer l'estimation de la condition initiale. Là encore, la méthode est performante et extrêmement rapide, comme de nombreux tests numériques l'ont démontré. En parallèle, plusieurs résultats théoriques de convergence ont été obtenus dans des cas simplifiés.

Enfin, une étude a été réalisée à l'interface de ces deux thématiques: l'extraction de données, et plus précisément de champs de vitesses, à partir de séquences d'images météorologiques ou océanographiques. L'idée consiste à chercher un champ de vitesse (ou déplacement) qui transporte une image sur la suivante. L'approche considérée est variationnelle, et basée sur la minimisation d'une fonctionnelle non linéaire dépendant du champ de vitesse. Une approche multi-grille permet d'obtenir très rapidement des champs de vitesse. Ces vitesses peuvent alors être assimilées directement dans un système d'assimilation.

## Summary:

In a first chapter, we consider image processing problems. We applied the topological asymptotic analysis to the edge detection problem. Once the edges are identified, one can easily consider the restoration/enhancement and classification problems. The inpainting problem has also been considered, but from a slightly different point of view given the Dirichlet and Neumann conditions on the boundary of the unknown part of the image, the topological gradient allows one to retrieve the missing edges of the hidden zone, and then to reconstruct an unblurred image. Finally, the segmentation problem has been considered with the same mathematical tools, using the analyticity of the enhanced solution with respect to a small parameter. All these algorithms are extremely efficient and fast, and allows us to process images and even movies in real time.

The second chapter is devoted to data assimilation. We developed a new algorithm: the Back and Forth Nudging (BFN). The standard nudging technique consists in adding to the equations of the model a relaxation term that is supposed to force the observations to the model. The BFN algorithm consists in repeatedly performing forward and backward integrations of the model with relaxation (or nudging) terms, using opposite signs in the direct and inverse integrations, so as to make the backward evolution numerically stable. Extensive numerical experiments have been performed on several simplified geophysical models, showing the efficiency of this easy-to-implement and fast approach. Moreover, several theoretical results of convergence have been obtained in simple situations.

Finally, we also worked at the interface of these two topics and considered image data assimilation. The idea is to extract velocity fields from a sequence of oceanographic or meteorological images. A variational approach has been proposed, in which the minimization of a nonlinear cost function provides a displacement (or velocity) field between two images. A multi-grid approach and an appropriate minimization process, allow us to extract the information very quickly. These "pseudo"-observations can then be directly assimilated as the velocity is usually a model variable.