Numerical simulation of shallow water equations and some physical models in image processing

by

Gloria Haro Ortega

Ph.D. Thesis

Doctorate in Computer Science and Digital Communication
Department of Technologies

Advisors: Vicent Caselles Costa
Rosa Donat Beneito

Universitat Pompeu Fabra

Barcelona, April 2005

Dipòsit legal: B.40014-2005
ISBN: 84-689-3831-9

Agradecimientos

Quisiera agradecer a mis directores de tesis, Vicent Caselles y Rosa Donat, por su disponibilidad, dedicación, y por sus amplios conocimientos que tan bien me han sabido transmitir. En particular a Vicent Caselles por su apoyo constante y por haber confiado en mi desde un principio. Sin duda alguna ha sido un gran privilegio el ternerlos como directores. Esta tesis ha sido posible gracias a ellos.

Muchas gracias a quienes también han contribuido en la realización de este trabajo: Andrés Almansa, Marcelo Bertalmío, Guillermo Sapiro y Joan Verdera.

No puedo olvidarme tampoco del resto de integrantes del grupo de imagen: Coloma Ballester, Jordi Faro, Lluís Garrido, Laura Igual y Enric Meinhardt, siempre dispuestos a prestarme su ayuda. Incluyo también en estas líneas a Andreu Solé, porque a todos nos gustaría que siguiera aquí.

Gracias a mis compañeros de doctorado (los de la UPF y los de la UPC), por compartir inquietudes, por los buenos momentos y por hacer más llevaderos los más difíciles.

Ya por último, aunque sin restarles por ello protagonismo, gracias a mis padres, Diego y M. Carmen, a mi hermano Pedro, por supuesto también a Rebeca, y a mis abuelos: Carmen, Gloria y Pedro. De manera muy especial, gracias a Amador: por su paciencia y su apoyo incondicional.

Abstract

There are two main subjects in this thesis: the first one deals with the numerical simulation of shallow water equations, the other one is the resolution of some problems in image processing. As a common scenario of the image problems we can say that all of the approaches use the restrictions given by their corresponding acquisition models.

With the aim of solving numerically the shallow water equations we extend a high resolution numerical scheme adapted to homogeneous systems of hyperbolic equations [66]. One of the interests of this scheme is the use of the Marquina's flux splitting technique, introduced by Donat and Marquina [59]. The potential of the Marquina's technique with respect to the schemes based on a single flux decomposition has been shown experimentally in the context of gas dynamics. The shallow water equations are described by means of a non homogeneous system, the presence of the source term is due to the topography over which the water runs. It is of great importance in the resolution of the shallow water equations that the scheme maintains the steady state without introducing spurious oscillations. Once we have extended the Marquina's technique to the non homogeneous case, the analysis of the scheme in the steady state of water at rest shows that the use of two different Jacobians in each interface prevents the scheme from verifying the exact C-property (conservation property) introduced by Bermúdez and Vázquez [19]. However, the approximate C-property is verified if we use the high order versions of the scheme. The same analysis reveals that we can achieve the exact C-property if we use one Jacobian (a single flux decomposition). In light of these conclusions we propose a combined scheme that uses the Marquina's double flux decomposition when adjacent states are not close and a single decomposition otherwise. The combination of both techniques permits to benefit from the advantages of both schemes. On the

other hand, we propose a special treatment of the numerical scheme at wet/dry fronts and at dry bed generation situations. Finally, the performance of the scheme is evaluated through classical experiments in the shallow water literature. In addition, we show particular instances where the combined scheme performs better than the Marquina's technique and the classical approach of a single decomposition.

The second subject treated in this thesis is the digital simulation of the Day for Night (or American Night in Europe). The inappropriateness or impossibility to shot some scenes at night leads to the use, in film industry, of the Day for Night technique. Then, the scenes are shot at daytime, placing a blue filter in front of the lens and underexposing the film. In this thesis we propose an algorithm that simulates a night image coming from a day image. The algorithm considers some aspects that contribute to the difference in visual perception at daytime with respect to night perception: change of illuminant, modification of chromaticity, of luminance and of contrast as well as the loss of visual acuity. In order to simulate this last aspect we introduce a novel diffusion Partial Differential Equation (PDE) that produces an anisotropic diffusion which is controlled by the local luminances. This PDE simulates the spatial summation principle of the photoreceptors in the retina, a principle that has been analyzed by means of psychophysic experiments. As a difference with other techniques present in the literature with the same purpose, this equation does not produce the ringing phenomenon.

The gap restoration (inpainting) on surfaces is the object of the third part of this dissertation. As an introductory level, we revise the use of the elastica in image inpainting, as an interpretation of the Gestalt theory on perception, and its relation with the Willmore functional (the integral of the square mean curvature). We also revise some of the existing methods for surface inpainting. We improve with an automatic initialization method the variational problem presented in [139]. Then we propose other geometrical approaches for 3D inpainting. The two first techniques are based on the mean curvature: one of them minimizes the Laplacian of a distance function and the other makes a diffusion of the mean curvature reconstructing afterwards the surface with the prescribed curvature. On the other hand, we also use two interpolation methods: the resolution of the Laplace equation, and an Absolutely Minimizing Lipschitz Extension (AMLE). These approaches are evaluated with synthetic and real surfaces.

Finally, we solve the restoration of satellite images. The variational problem that we propose is based on the complete model of acquisition system which considers the

following aspects: noise, modulation transfer function of the system and sampling process (regular or irregular). The functional to minimize contains also a term based on the Total Variation in order to regularize the solution. With the aim to solve the variational problem we extend the technique proposed by Chambolle [40] for the restoration of noised images and we also use the algorithm of Gröchenig and Strohmer [80] that recovers a uniform sampling from irregular samples. The restoration method that we propose manages to do irregular to regular sampling, denoising, deconvolution and zoom at the same time.

Resumen

Los temas tratados en esta tesis son, por un lado, la simulación numérica de las ecuaciones de aguas someras ("shallow waters") y por otro, la resolución de algunos problemas de procesamiento de imágenes. Como marco común de estos problemas en imagen podemos decir que todos ellos se abordan usando las rectricciones dadas por sus correspondientes modelos de adquisición.

Con el objetivo de resolver numéricamente las ecuaciones de aguas someras, extendemos un esquema numérico de alta resolución, pensado originalmente para sistemas homogéneos de ecuaciones hiperbólicas [66]. Es de especial interés en este esquema la utilización de la técnica de Marquina, introducida por Donat y Marquina [59], y que utiliza una separación de flujos en cada una de las interfases. El potencial del esquema de Marquina respecto a las técnicas basadas en una única descomposición del flujo ha sido demostrada con experimentos en el contexto de la dinámica de gases. Las ecuaciones de aguas someras se describen mediante un sistema no homogéneo, la presencia del término fuente es debida a la topografía sobre la que avanza el agua. En la resolución de las ecuaciones de aguas someras es de vital importancia que el esquema numérico mantenga el estado estacionario sin introducir ondas espúreas. Una vez propuesta la extensión de la técnica de Marquina al caso no homogéneo, el análisis del esquema en el estado estacionario de aguas en calma demuestra que el uso de dos Jacobianos diferentes en cada interfase impide que el sistema verifique la propiedad de conservación C exacta introducida por Bermúdez y Vázquez [19]. En cambio, la propiedad C aproximada se verifica si se usan las versiones del esquema que usan un orden elevado de interpolación. El mismo análisis nos desvela que se puede conseguir la propiedad C exacta si usamos un solo Jacobiano (una única descomposición de flujos). A la vista de estos

resultados, se propone un esquema combinado que usa la técnica de doble descomposición de flujos de Marquina cuando los dos estados adyacentes no están próximos y una única descomposición en caso contrario. La combinación de estas dos técnicas permite valernos de las ventajas de ambos esquemas. Por otro lado, se propone un tratamiento especial del esquema numérico en los frentes seco/mojado y en las situaciones de generación de zona seca. Finalmente, el correcto comportamiento del esquema se evalúa a través de experimentos clásicos en la literatura de aguas someras. También se ponen de manifiesto casos concretos donde el esquema combinado funciona mejor que el esquema de Marquina o la técnica clásica de descomposición única.

El segundo tema tratado es la simulación digital de la Noche Americana ("Day for Night"). La improcedencia o imposibilidad de rodar algunas escenas durante la noche conlleva, en el cine, a la utilización de la técnica de la Noche Americana. Las escenas se ruedan de día poniendo un filtro azul en la cámara y subexponiendo la película. En la tesis proponemos un algoritmo que simula una imagen nocturna a partir de una imagen diurna. El algoritmo considera varios aspectos que intervienen en la diferencia de percepción visual nocturna respecto a la diurna: cambio de iluminante, modificación de la cromaticidad, de la luminancia y del contraste así como la pérdida de agudeza visual. Para simular este último aspecto se propone una ecuación en derivadas parciales (EDP) que produce una difusión anisotrópica en función de la luminancia local. La EDP simula el principio de sumación espacial de los fotoreceptores situados en la retina y que ha sido analizado mediante varios experimentos psicofísicos. A diferencia de otras técnicas presentes en la literatura y con el mismo propósito, esta ecuación no produce el fenómeno de ringing.

La restauración de agujeros ("inpainting") en superficies es objeto de la tercera parte de la tesis. A nivel introductorio se revisa el uso de la elástica en el inpainting de imágenes, como interpretación de la teoría de percepción de la Gestalt, y su relación con el funcional de Willmore (la integral de la curvatura media al cuadrado); así como también algunos de los métodos existentes de inpainting en superficies. Primeramente se mejora el problema variacional presentado en [139] mediante un método de inicialización automático. Seguidamente se proponen otros enfoques geométricos para abordar el problema del inpainting en tres dimensiones. Las dos primeras técnicas están basadas en la curvatura media: una de ellas minimiza el Laplaciano de una función distancia y la otra realiza una difusión de la curvatura media reconstruyendo posteriormente la superficie asociada a dicha curvatura. Por otro lado también se utilizan dos métodos de

interpolación: la resolución de la equación de Laplace y el método AMLE (Absolutely Minimization Lipschitz Extension). La verificación de estas técnicas se realiza tanto en superficies sintéticas como en reales.

Por último, dedicamos una parte a la restauración de imágenes satelitales. El problema variacional propuesto con este objetivo se basa en el modelo completo de adquisición de imágenes que contiene los siguientes aspectos: ruido, función de transferencia del sistema y muestreo (regular o irregular). El funcional a minimizar también contiene un término basado en la Variación Total de cara a regularizar la solución. La resolución del problema variacional se realiza extendiendo la técnica propuesta por Chambolle [40] para la restauración de imágenes con ruido y mediante el algoritmo de Gröchenig y Strohmer [80] para recuperar un muestreo uniforme a través de muestras irregulares. El método de restauración propuesto consigue obtener una colección de muestras regulares a partir de un muestreo irregular, eliminando a la vez el ruido, deconvolucinando la imagen y haciendo un zoom.

Contents

Chapter 1

Outline of Dissertation

This thesis is organized in four parts, each one of them forms an autonomous part which can be read independently from the others. The first one deals with the numerical simulation of shallow water equations. In the other three parts we treat some problems that arise in image processing and that can be modeled using the restrictions imposed by their corresponding acquisition models. An introduction and motivation of the different problems can be found at the beginning of each part. In what follows, we give a detailed description of the contents of each chapter.

PART I: Numerical simulation of Shallow Water Equations

Chapter 2: We revise some theoretical aspects of the shallow water equations: hypothesis considered by the derivation of the model from the Navier-Stokes equations, the equations that describe the model and the eigen-decomposition of their associated Jacobian. Then, we recall the Riemann problem of two adjacent states as well as their possible solutions: rarefaction waves and shock waves. Finally, we study the conditions that lead to the formation of a dry zone between two wet states.

Chapter 3: In this chapter we present an extension of Marquina's flux formula [59] to the non homogeneous case in order to simulate the shallow water equations. We analyze the performance of the proposed numerical scheme at the steady state and the conditions under which the approximate and the exact C-properties are verified. In view of this analysis we propose a combined scheme that uses one or two Jacobians, each approach when most adequate. In addition, we propose a special treatment of the scheme at wet/dry fronts as well as at situations of dry bed generation. We test the performance of the proposed scheme with classical experiments in the shallow water literature and show particular instances where the combined scheme performs better

than the Marquina's technique and the classical approach of a single decomposition.

Appendix A: It contains some remarks on the conservativity of the scheme and proposes an alternative way to treat the source term integrals that provides a fully-conservative scheme.

Part II: Day for night

Chapter 4: We introduce a digital day for night algorithm. The algorithm transforms an image taken at daytime into a simulated night image. This is achieved in five steps: change of illuminant, modification of chromaticity, modification of luminance, change of contrast and simulation of loss of visual acuity. In the last step we introduce a novel diffusion PDE that models the spatial summation principle, takes luminance into account and produces no ringing. Some examples of the algorithm are present at the end of the chapter.

Appendix B: The proofs of theorems in chapter 4 are contained in this appendix.

PART III: Inpainting Surface Holes

Chapter 5: Firstly, we revise the use of the Euler's elastica in image inpainting and its relation with the Willmore functional. Some of the existent methods for surface inpainting are also described. In particular, the variational problem introduced in [139] is formalized and improved here with an automatic initialization method. Then, other geometric variational approaches for inpainting surface holes are proposed here. Two of them are based on the mean curvature. The first one minimizes the Laplacian of a distance function. The second one is a diffusion of the mean curvature followed by the reconstruction of the distance function with the prescribed curvature. The other two approaches are interpolation techniques: the resolution of the Laplace equation, and an Absolutely Minimizing Lipschitz Extension (AMLE). We compare the results of these approaches with synthetic and real surfaces.

PART IV: Restoration of irregularly sampled images

Chapter 6: This chapter is devoted to the restoration of satellite images. The proposed variational model uses the Total Variation as a regularizing term and it is also based on the complete acquisition model of a satellite image. We concentrate on the special case of irregular sampling. Thus, we recall the basic concepts of irregular sampling and the necessary conditions for recovering a uniform sampling from irregular samples. Then we review the irregular to regular sampling algorithm of Gröchenig and Strohmer [80] which we shall incorporate in our proposed algorithm. In order to solve

the full restoration model, i.e. irregular to regular sampling, denoising, deconvolution and zoom, we extend the technique introduced by Chambolle [40] which denoises an image using the Total Variation. We compare the restoration results of our proposed algorithm with those obtained by the algorithm of Gröchenig and Strohmer.

Appendix C: We explain how a general Modulation Transfer Function (MTF) of a satellite can be modeled.

Appendix D: In this appendix we recall the definition of functions of bounded variation.

Chapter 7: This is the last chapter of the dissertation. We give the main conclusions of this work and describe the perspectives and the future work.

Part I

Numerical simulation of Shallow Water Equations

Chapter 2

Shallow Water Equations

ABSTRACT

This chapter is intended to revise some theoretical aspects of the shallow water equations. The formation of different types of waves as a solution of the Riemann problem of two adjacent states is also considered. Finally, we study how and when a dry state is formed between two wet states.

2.1 Introduction

Many practical problems involving shallow water flow in oceanography and atmospheric sciences are conveniently modeled by considering the one-dimensional or two-dimensional Saint-Venant, or shallow water, system. This is a hyperbolic system of conservation laws that approximately describes various geophysical flows, such as rivers, coastal areas, dam-breaks and even oceans when completed with a Coriolis term. For practical applications, the inclusion of a non-flat bottom topography is required, which leads to the occurrence of source terms of geometrical nature. The shallow water model is widely used in many different applications so as to make predictions and simulations of practical interest. As examples of applications we can mention the simulation of internal tides in the Strait of Gibraltar [38, 74], the dam-break simulation of a real case occurred in Aznalcóllar [74] and also transport of sediments [39, 52].

The system of shallow water equations can be written as follows

$$U_t + \text{div}(F(U)) = S(\mathbf{x}, U). \tag{2.1}$$

These equations are derived by depth averaging the Navier-Stokes equations, neglecting diffusion of momentum by viscous and turbulent effects. Ignoring also friction losses, the source term is only due to the geometry of the bottom topography and the resulting system of equations in the two dimensional case becomes (see [134] and references therein)

$$U_t + F(U)_x + E(U)_y = S \tag{2.2}$$

$$\begin{pmatrix} h \\ q_1 \\ q_2 \end{pmatrix}_t + \begin{pmatrix} q_1 \\ \frac{q_1^2}{h} + \frac{1}{2}gh^2 \\ \frac{q_1 q_2}{h} \end{pmatrix}_x + \begin{pmatrix} q_2 \\ \frac{q_1 q_2}{h} \\ \frac{q_2^2}{h} + \frac{1}{2}gh^2 \end{pmatrix}_y = \begin{pmatrix} 0 \\ -ghz_x \\ -ghz_y \end{pmatrix}$$

where h is the water depth, q_1 and q_2 are the two components of the discharge, and z is the bottom topography. Indeed, we make an abuse of notation by letting $\mathrm{div}(F(U)) = F(U)_x + E(U)_y$. The corresponding eigenvalues of the Jacobian matrices of flux components F and E are:

$$\begin{aligned} \lambda_1^F &= u_1 - c & \lambda_2^F &= u_1 & \lambda_3^F &= u_1 + c \\ \lambda_1^E &= u_2 - c & \lambda_2^E &= u_2 & \lambda_3^E &= u_2 + c \end{aligned}$$

where $u_1 = q_1/h$ and $u_2 = q_2/h$ are the components of the fluid velocity and $c = \sqrt{gh}$ is the sound velocity. The superscripts F and E refer respectively to the fluxes $F(U)$ and $E(U)$ in (2.2), which are the two components of the flux vector. The matrices of right eigenvectors R^F and R^E, as well as the matrices of left eigenvectors L^F and L^E are:

$$R^F = \begin{pmatrix} 1 & 0 & 1 \\ \lambda_1^F & 0 & \lambda_3^F \\ u_2 & 1 & u_2 \end{pmatrix} \quad L^F = \begin{pmatrix} \lambda_3^F/(2c) & -1/(2c) & 0 \\ -u_2 & 0 & 1 \\ -\lambda_1^F/(2c) & 1/(2c) & 0 \end{pmatrix}$$

$$R^E = \begin{pmatrix} 1 & 0 & 1 \\ u_1 & 1 & u_1 \\ \lambda_1^E & 0 & \lambda_3^E \end{pmatrix} \quad L^E = \begin{pmatrix} \lambda_3^E/(2c) & 0 & -1/(2c) \\ -u_1 & 1 & 0 \\ -\lambda_1^E/(2c) & 0 & 1/(2c) \end{pmatrix}$$

2.2 Wave formation

2.2.1 Introduction

In this section we are going to study the solution of the Riemann problem formed by two adjacent states. We will see that there exists a solution provided that the initial states are sufficiently close in some sense. This solution consists of a single wave,

namely a shock wave or a rarefaction wave. Here, we are going to develop the solution of the Riemann problem starting from general known results and basic concepts of the theory of hyperbolic systems of conservation laws. A detailed and rigorous study on such a topic can be found in [130], a mathematical theory is also revised in [101] and [72]. In [134] the study is directly done on the shallow water equations.

Before analyzing each type of solution we recall the system of equations that define the shallow water in one dimension over flat topographies:

$$\begin{aligned} h_t &+ q_x = 0 \\ q_t &+ \left(\tfrac{q^2}{h} + \tfrac{1}{2}gh^2\right)_x = 0 \end{aligned} \tag{2.3}$$

The system of conservation laws (2.3) together with initial-value data of the form

$$U(x,0) = \begin{cases} U_L = (h_L, q_L) & \text{if } x < 0 \\ U_R = (h_R, q_R) & \text{if } x > 0 \end{cases} \tag{2.4}$$

is known as the Riemann problem.

For a general system of conservation laws $U_t + F(U)_x = 0$ where $U(x,t) \in \mathbb{R}^N$, we call J the Jacobian matrix of F which can be decomposed in its eigenvalues $\lambda_k(U)$ and its associated eigenvectors $r_k(U)$ ($k = 1, ..., N$). Then the *characteristic variables* are defined as LU. The eigenvalues λ_k are called the *characteristic velocities*.

2.2.2 Rarefaction waves

Riemann solutions always remain constant along all rays of the form $x = \epsilon t$, where $\epsilon \in \mathbb{R}$. A rarefaction wave that lies between the states U_L and U_R also has this property and takes the form:

$$U(x,t) = \begin{cases} U_L & \text{if } x \leq \epsilon_1 t \\ W(x/t) & \text{if } \epsilon_1 t < x < \epsilon_2 t \\ U_R & \text{if } x \geq \epsilon_2 t \end{cases}$$

where W is a smooth function with $W(\epsilon_1) = U_L$ and $W(\epsilon_2) = U_R$, so a rarefaction wave is a continuous solution. If we differentiate $U(x,t) = W(x/t)$ in both space and time:

$$\begin{aligned} U_x(x,t) &= \frac{1}{t} W'(x/t) \\ U_t(x,t) &= -\frac{x}{t^2} W'(x/t) \end{aligned} \tag{2.5}$$

If we substitute (2.5) in $U_t + F'(U)U_x = 0$ and multiply by t we get

$$F'(W(\epsilon))W'(\epsilon) = \epsilon W'(\epsilon) \tag{2.6}$$

The equation (2.6) admits two solutions. One of them is $W'(\epsilon) = 0$, i.e., W is constant. If $W'(\epsilon) \neq 0$ then W' is proportional to an eigenvector of F' and ϵ is its corresponding eigenvalue:

$$W'(\epsilon) = \alpha(\epsilon) r_k(W(\epsilon)) \tag{2.7}$$

$$\epsilon = \lambda_k(W(\epsilon)) \tag{2.8}$$

Since we are working in one dimension $k = 1, 2$. As a consequence of (2.7) all values lie along some integral curve of r_k. The states $U_L = W(\epsilon_1)$ and $U_R = W(\epsilon_2)$ also lie on the same integral curve. This a necessary condition for the existence of a rarefaction wave connecting U_L and U_R but not sufficient. We need $\epsilon = x/t$ to be monotonically increasing as $W(\epsilon)$ moves from U_L to U_R along the integral curve. By (2.8), monotonicity of ϵ is equivalent to monotonicity of $\lambda_k(W)$ and so

$$\lambda_k(U_L) < \lambda_k(U_R) \tag{2.9}$$

There are two families of rarefaction waves, each one of them corresponds to the characteristic family of the k eigenvector. In addition, a k-rarefaction has the property that k-Riemann invariants remain constant across the wave. A k-Riemann invariant is defined as the smooth function $w : \mathbb{R}^N \to \mathbb{R}$ that satisfies

$$< r_k(U), \nabla w(U) >= 0 \qquad \text{for any } U \in \mathbb{R}^N.$$

The 1-Riemann invariant is then $u + 2\sqrt{gh} = \textit{constant}$ and the 2-Riemann invariant is $u - 2\sqrt{gh} = \textit{constant}$ (remember that $u = q/h$).

In order to determine the function $W(\epsilon)$, we determine the scale factor $\alpha(\epsilon)$ by differentiating (2.8) with respect to ϵ:

$$1 = \nabla \lambda_k(W(\epsilon)) \cdot W'(\epsilon)$$

making use of (2.7)

$$\alpha(\epsilon) = \frac{1}{\nabla \lambda_k(W(\epsilon)) \cdot r_k(W(\epsilon))} \, .$$

Hence,

$$W'(\epsilon) = \frac{r_k(W(\epsilon))}{\nabla \lambda_k(W(\epsilon)) \cdot r_k(W(\epsilon))} \qquad \epsilon_1 \leq \epsilon \leq \epsilon_2 \tag{2.10}$$

gives a system of ODEs for $W(\epsilon)$ with initial data:

$$W(\epsilon_1) = U_L \qquad \text{where } \epsilon_1 = \lambda_k(U_L) \text{ and } \epsilon_2 = \lambda_k(U_R)$$

Since we fix U_L as initial condition, we are going to find the states that can be connected to U_R by a rarefaction wave. Note that the denominator in (2.10) is finite for $\epsilon_1 \leq \epsilon \leq \epsilon_2$ only if λ_k is monotone between ϵ_1 and ϵ_2.

Solving (2.10) for the first eigenvalue and its corresponding eigenvector,

$$\lambda_1(W) = u - \sqrt{gh} \qquad r_1(W) = (1, \lambda_1)^T; \qquad \text{where } u = q/h$$

gives the one parameter family of 1-rarefaction waves:

$$\begin{aligned} \sqrt{gh} &= \frac{1}{3}\left(2\sqrt{gh_L} + u_L - \frac{x}{t}\right) \\ u &= \frac{1}{3}\left(2\sqrt{gh_L} + u_L + 2\frac{x}{t}\right) \end{aligned} \tag{2.11}$$

where $u_L = q_L/h_L$. Thus, the set of states which can be connected to the right of U_L by a 1-rarefaction lie in the curve

$$u - u_L = 2\left(\sqrt{gh_L} - \sqrt{gh}\right) \equiv R_1(h; U_L)\ . \tag{2.12}$$

In the same manner for the second eigenvalue and eigenvector,

$$\lambda_2(W) = u + \sqrt{gh} \qquad r_2(W) = (1, \lambda_2)^T$$

the family of 2-rarefaction waves associated to them is:

$$\begin{aligned} \sqrt{gh} &= \frac{1}{3}\left(2\sqrt{gh_L} - u_L + \frac{x}{t}\right) \\ u &= \frac{1}{3}\left(-2\sqrt{gh_L} + u_L + 2\frac{x}{t}\right) \end{aligned} \tag{2.13}$$

Now, the curve that defines the states which can be connected to U_L by a 2-rarefaction on the right is

$$u - u_L = -2\left(\sqrt{gh_L} - \sqrt{gh}\right) \equiv R_2(h; U_L)\ . \tag{2.14}$$

We have seen that a k-rarefaction must satisfy (2.9), then if U_R is connected to U_L by a 1-rarefaction, it must verify:

$$u_L - \sqrt{gh_L} < u_R - \sqrt{gh_R}$$

where $u_L = q_L/h_L$ and $u_R = q_R/h_R$. In addition, setting $U = U_R$ in (2.12) and substituting it into this last inequality we obtain

$$h_R < h_L \tag{2.15}$$

If we return to (2.12) and make use of condition (2.15) we obtain the second condition that must verify the state U_R

$$u_R > u_L \tag{2.16}$$

In a similar way, a 2-rarefaction must satisfy:

$$u_L + \sqrt{gh_L} < u_R + \sqrt{gh_R}$$

that added to the curve (2.14) for $U = U_R$ gives the two conditions on U_R

$$h_R > h_L \tag{2.17}$$

$$u_R > u_L \tag{2.18}$$

Note that if the relation of velocities of states U_R and U_L is (2.18) they are joined by a k-rarefaction, then the complementary, $u_R < u_L$, is satisfied for states that are connected by a k-shock.

2.2.3 Shock waves

Let us start by considering a general system of conservation laws $U_t + F(U)_x = 0$ where $U(x,t) \in \mathbb{R}^N$. We fix the state $U_L = (h_L, q_L)$ and compute the possible states $U = (h, q)$ which can be connected to U_L by a shock wave. In such a case both states much satisfy the Rankine-Hugoniot conditions:

$$F(U) - F(U_L) = s(U - U_L) \tag{2.19}$$

Since the state U_L is fixed this gives a system of N equations and $N + 1$ unknowns: the N components of U and the shock velocity s. If we again parameterize the states by $U = W(\epsilon)$ with $W(0) = U_L$ as well as the velocity $s(\epsilon)$, substitute them in (2.19) and differentiate with respect to ϵ and make $\epsilon \to 0$ we obtain the following expression:

$$F'(W(0))W'(0) = s(0)W'(0)$$

We arrive then to a similar condition as in the rarefaction waves, if we consider that $W'(0) \neq 0$, W' has to be proportional to an eigenvector of F' and s has to be its eigenvalue.

More particularly, in our case $N = 2$ and the Rankine-Hugoniot conditions are:

$$\begin{aligned} s(h - h_L) &= q - q_L \\ s(q - q_L) &= \left(\frac{q^2}{h} + \frac{1}{2}gh^2\right) - \left(\frac{q_L^2}{h_L} + \frac{1}{2}gh_L^2\right) \end{aligned}$$

Then we have a system of two equations with three unknowns, so we can find one parameter family of solutions (note that$(h - h_L)$ and $(q - q_L)$ are not zero since the initial states are different):

$$q = q_L h/h_L \pm h(h - h_L)\sqrt{\frac{g}{2}\frac{(h + h_L)}{hh_L}} \tag{2.20}$$

The $\pm$ signs in this equation give two solutions, one for each characteristic wave determined by each eigenvalue. Since q is parameterized by h we can also parameterize this curve by taking for example $h = h_L(1+\epsilon)$ and we have,

$$U = U_L + \epsilon \begin{bmatrix} h_L \\ q_L - (1+\epsilon)h_L\sqrt{\frac{g}{2}\frac{2+\epsilon}{1+\epsilon}h_L} \end{bmatrix} \tag{2.21}$$

$$U = U_L + \epsilon \begin{bmatrix} h_L \\ q_L + (1+\epsilon)h_L\sqrt{\frac{g}{2}\frac{2+\epsilon}{1+\epsilon}h_L} \end{bmatrix} \tag{2.22}$$

As we can see, the first family (2.21) comes from curve (2.20) with the minus sign because differentiating it with respect to ϵ and letting $\epsilon \to 0$ the result is proportional to the first eigenvector associated with the eigenvalue $\lambda_1 = u - \sqrt{gh}$. So, we can define the 1-shock curve as:

$$u - u_L = u_L - (h - h_L)\sqrt{\frac{g}{2}\frac{(h+h_L)}{hh_L}} \equiv S_1(h; U_L) \tag{2.23}$$

By analogy, the family (2.22) is associated with the eigenvalue $\lambda_2 = u + \sqrt{gh}$ and comes from (2.20) with the plus sign. The 2-shock curve is defined as:

$$u - u_L = u_L + (h - h_L)\sqrt{\frac{g}{2}\frac{(h+h_L)}{hh_L}} \equiv S_2(h; U_L) \tag{2.24}$$

In addition shocks must satisfy the *entropy condition* (see [130] for more details):

$$\lambda_k(U_L) > s > \lambda_k(U_R)$$

Moreover, we have seen in the precedent section that the condition on U_R for the shock waves is

$$u_r < u_L \tag{2.25}$$

Thus, applying condition (2.25) on the 1-shock wave (2.23) we obtain the second condition that must verify the right state U_R:

$$h_R < h_L \tag{2.26}$$

The same condition (2.25) applied on the 2-shock wave (2.24) gives:

$$h_R > h_L \tag{2.27}$$

2.2.4 Integral curves

We have seen in the precedent section that both k-shocks and k-rarefaction curves defined by the parameterized curve $W(\epsilon)$ have its tangent vector proportional to the k eigenvector of the flux function. These curves are then called integral curves.

We can put all of these curves together in the $h-u$ plane as in figure 2.1. This figure shows that the $h-u$ plane is divided into four disjoint regions I,II,III and IV. Moreover, the curves R_1 and S_1 have second-order contact (their first two derivatives are equal) at point U_L, this happens also with R_2 and S_2 and can be justified as a consequence of a general theorem in [130].

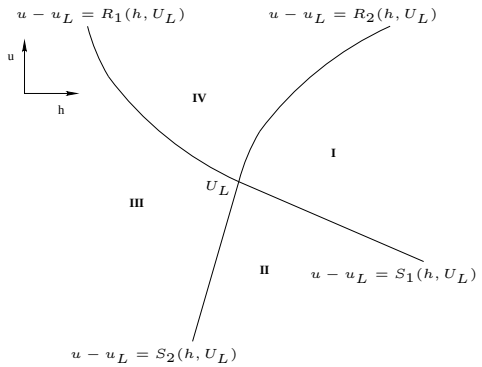


Figure 2.1: Integral curves for the state U_L

Consider the point U_L as fixed, the Riemann problem (2.3)-(2.4) and let us allow U_R to vary. If U_R lies on any of the four studied curves (2.12),(2.14),(2.23) or (2.24) then the solution is completely defined by one of these curves and both states are joined by either a k-shock or a k-rarefaction wave depending on the type of curve the point U_R belongs to. On the other hand, if U_R does not lie on any of the four curves it belongs to any of the four open regions I, II, III or IV. Let us define, for $\bar{U} \in \mathbb{R}^2$ the curves:

$$\begin{aligned} W_i^S(\bar{U}) &= \left\{(h,u) : u = u_L + S_i(h,\bar{U})\right\}, \qquad i = 1,2 \\ W_i^R(\bar{U}) &= \left\{(h,u) : u = u_L + R_i(h,\bar{U})\right\}, \qquad i = 1,2 \\ W_i(\bar{U}) &= W_i^S(\bar{U}) \cup W_i^R(\bar{U}), \qquad i = 1,2 \end{aligned}$$

and then for a fixed $U_L \in \mathbb{R}^2$, we consider the family of curves

$$\mathfrak{F} = \left\{W_2(\bar{U}) : \bar{U} \in W_1(U_L)\right\}$$

Making the hypothesis that the $h-u$ plane is covered univalently by the family of curves $\mathfrak{F}$, i.e., through each point U_R there passes exactly one curve of $\mathfrak{F}$. Then the solu-

tion to the Riemann problem (2.3)-(2.4) consists on three states U_L, U_R and $\bar{U}$ connected by two different waves. We connect $\bar{U}$ to U_L on the right by a backward wave (1-shock or 1-rarefaction), and U_R to $\bar{U}$ on the right by a forward wave (2-shock or 2-rarefaction).

For example, if U_R is in region II, there is a unique point $\bar{U}$ for which the curve $W_2(\bar{U})$ is in $\mathfrak{F}$ and passes through U_R. Since $\bar{U} \in W_1^S(U_L)$ and $U_R \in W_2^S(\bar{U})$ then $\bar{U}$ is connected to U_L on the right by a back shock and U_R is connected to $\bar{U}$ on the right by a front shock, see figure 2.2.

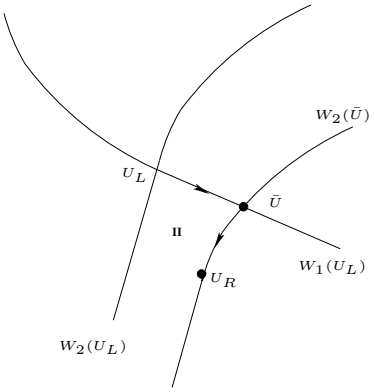


Figure 2.2: Intermediate state $\bar{U}$ when U_R lies in region II

Remember that we assumed that the $h-u$ plane is covered univalently by the family of curves $\mathfrak{F}$. The proof is split into two cases: in the first case, we consider that U_R lies in one of the regions I, II or III; and in the second case, U_R lies in region IV. For the first case, the proof follows the same arguments as a similar proof in [130] but for the gas dynamics equations. We will see now that in the second case there are points in this region IV that cannot be covered by an element of $\mathfrak{F}$.

Consider a point $\bar{U} = (\bar{h}, \bar{u}) \in W_1^R(U_L)$, as the water depth is restricted to $h \geq 0$, then from (2.12) we have

$$\bar{u} = u_L + 2\sqrt{gh_L} - 2\sqrt{g\bar{h}} \leq u_L + 2\sqrt{gh_L} \tag{2.28}$$

On the other hand, a point $U_R = (h_R, u_R) \in W_2^R(\bar{U})$ and connected to $\bar{U}$ is defined by (2.14) with $U_L = \bar{U}$ obtaining

$$u_R = \bar{u} - 2\sqrt{g\bar{h}} + 2\sqrt{gh_R}$$

and using again the positivity restriction of h and the inequality (2.28) we obtain

$$u_R - u_L \leq 2\sqrt{gh_L} + 2\sqrt{gh_R} \tag{2.29}$$

This is the condition that states in region IV must verify to be covered by curves in $\mathfrak{F}$. If we impose h to be strictly positive (to ensure a solution without dry zones) then condition (2.29) is an strict inequality and it is called *depth positivity condition* [134].

2.2.5 Occurrence of dry bed

In case that the velocities of the initial states of the Riemann problem (2.3),(2.4) are sufficiently large and condition (2.29) does not hold, then a dry bed between both states is formed. The vacuum is joined to U_L and U_R by a back rarefaction and a forward rarefaction respectively.

Indeed, a shock wave cannot be adjacent to a region of dry bed. This is proved in [134] in a very simple manner by considering a wet bed U_L and a dry bed U_R as the initial data (2.4) for the Riemann problem. After applying Rankine-Hugoniot conditions, a contradiction with the initial data is obtained.

The speed of the wet/dry back front s_L is obtained using the first equation in (2.11) for the 1-rarefaction wave, and considering U_L as left state and $h = 0$ in the right state. Thus, we obtain

$$s_L^* = u_L + 2c_L, \qquad c_L = \sqrt{gh_L}$$

In the same way, the wet/dry forward front can be obtained from 2-rarefaction equations (2.13) with $h = 0$ in the left state and U_R as the right state. We have

$$s_R^* = u_R - 2c_R, \qquad c_R = \sqrt{gh_R}$$

We can now define the solution for the Riemann problem (2.3),(2.4) whose initial states U_L and U_R do not verify condition (2.29)

$$U(x,t) = \begin{cases} U_L & \text{if} \quad x/t \leq u_L - c_L \\ U_L^* & \text{if} \quad u_L - c_L \leq x/t \leq s_L^* \\ U_0 & \text{if} \quad s_L^* \leq x/t \leq s_R^* \\ U_R^* & \text{if} \quad s_R^* \leq x/t \leq u_R + c_R \\ U_R & \text{if} \quad u_R + c_R \leq x/t \end{cases} \tag{2.30}$$

where U_L^* is defined by equations (2.11), U_R^* by equations (2.13) and U_0 represents the dry state.

Chapter 3

Numerical simulation

ABSTRACT

We present an extension of Marquina's flux formula [59] for the shallow water system. We show that the use of two different Jacobians at cell interfaces prevents the scheme from satisfying the exact C-property while the approximate C-property is satisfied for higher order versions of the scheme. The use of a single Jacobian in Marquina's flux splitting formula leads to a numerical scheme satisfying the exact C-property, hence we propose a combined technique that uses Marquina's two sided decomposition when the two adjacent states are not close and a single decomposition otherwise. Finally, we propose a special treatment at wet/dry fronts and situations of dry bed generation. We show the performance of the proposed scheme with classical experiments in the literature.

3.1 Introduction

The general form of a conservation law equation in his differential form is:

$$u_t + \text{div}(f(u)) = 0$$

where u is the conserved variable and $f(u)$ is the flux.

Nevertheless, a physical system is described not only with a unique conservation equation but with a system of conservation laws formed by the conservation of mass, momentum and energy. We can write these systems as,

$$U_t + \text{div}(F(U)) = 0 \tag{3.1}$$

we can see that the structure is the same as in the single equation but now U represents a vector of conserved quantities.

The flux transport of the conserved quantity can be convective or diffusive. Conservation laws with only convective fluxes are known as *hyperbolic* conservation laws. Convective transport requires specialized numerical treatment while diffusive terms can be treated by standard numerical methods.

There are special difficulties associated with solving hyperbolic equations: shocks formation, rarefactions and contact discontinuities. A good numerical implementation must overcome these problems. Methods based on naive finite difference approximations may work well for smooth solutions but not when discontinuities are present. For further details on these topics, we refer the reader to the books [72, 101] which provide a general overview to the properties of conservation laws equations and the basic numerical methods for solving them.

There exist many different numerical methods to solve hyperbolic systems of conservation laws. Among these, we have implemented the one proposed by Fedkiw et al. in [66] because it provides high accuracy without spurious oscillations and shock capturing. This method is based on a Shu-Osher finite difference Essentially Non-Oscillatory (ENO) combined with Marquina's flux splitting technique.

ENO methods present high accuracy in smooth regions and capture steep gradients in the flow without creating spurious oscillations. This is performed using an adaptative polynomial interpolation based on smoothness considerations that, in addition, doesn't need tunable parameters because smoothness is measured by the size of the k^{th} derivate in a k degree interpolant. To make possible shock capturing and for reasons of physical consistency and stability, this interpolation is based on upwind biased differencing. Upwind methods are always used in numerical schemes for conservation laws because they extrapolate information from data from the direction of propagation.

Originally, ENO schemes were based on the conservative control volume discretization of the equations. However, this formulation requires complicated transfers between cell averages and cell center nodal values and this transfers process becomes particularly complicated in two and higher spatial dimensions. To overcome such problems, Shu and Osher [126] developed a conservative finite difference form of the ENO method, using only nodal values of the conserved variables. Finally, they used a Runge-Kutta time integration scheme that is stable and also has the Total Variation Diminishing (TVD) property. The TVD property prevents the time stepping scheme from introducing spurious spatial oscillations into upwind-biased spatial discretizations.

In some situations, ENO methods present spurious oscillations. They are caused by the fact of evaluating the flux at a cell that separates two very different states, i.e., at a cell where the characteristic velocity changes sign (this is called a sonic point). The flux at the cell is computed by a linear average of the flux at nodes adjacent to the midpoint. This interpolation performs well in a smooth region but it may cause problems between nodes in an unresolved steep gradient, which would lead to oscillations in the solution. To avoid that, Donat and Marquina [59] introduce the technique of Flux Splitting and which is used in [66]. It consists of making use of the left and right values of the flux separately, in an upwind fashion, rather than using a single midpoint interpolation of both values. In particular, they propose to use two sided characteristic information at each interface. Thus, a left flux is evaluated with a left side biased reconstruction (ENO interpolation in [66]) over the left characteristic variables. They proceed in a similar way in the analogous case: the right flux computation. It is seen in [59, 66] that this numerical flux can be used to design robust High Resolution Shock Capturing (HRSC) schemes with good properties with respect to certain numerical pathologies.

Our aim in this chapter is to extend the numerical method in [66] so as to include the presence of the source term arising in shallow water equations. For that in section 3.2 we recall the numerical method introduced by Fedkiw et al. in [66]. In section 3.3 we will extend this method to the non homogeneous case. Numerical experiments that validate the proposed scheme are shown in section 3.4. Finally, in section 3.5 we give some conclusions.

3.2 Numerical scheme for Hyperbolic Systems of Conservation Laws

3.2.1 Time discretization

In [66], they use the *method of lines* approach. We explain now what it consists of. Once we have a numerical approximation $A(U)$ of the spatial derivative terms (as well as source terms in our case) in (2.2) the equation to solve becomes a system of Ordinary Differential Equations (ODEs):

$$U_t + A(U) = 0$$

In order to solve the time integration of these ODEs, a third order Total Variation Diminishing (TVD) Runge-Kutta method is used [126]. Being U^n the solution at the n

time step, the numerical scheme is the following:

$$\begin{aligned} U^* &= U^n - \Delta t A(U^n) \\ U^{**} &= U^n - \Delta t \left(\frac{1}{4} A(U^n) + \frac{1}{4} A(U^*) \right) \\ U^{n+1} &= U^n - \Delta t \left(\frac{1}{6} A(U^n) + \frac{1}{6} A(U^*) + \frac{2}{6} A(U^{**}) \right) \end{aligned}$$

where Δt is the time resolution. Indeed, we can also use a second or first order Runge-Kutta (see [66, 126] for more details) and this choice depends on the order of the ENO interpolation that we use in the spatial discretization.

3.2.2 Spatial discretization

Let us consider a two dimensional conservation law system of N equations which can be written in the general form (3.1), where $\text{div}(F(U)) = F(U)_x + E(U)_y$ (by abuse of notation on F). The discretization is applied independently in both dimensions (what is called dimension by dimension manner). Firstly, ENO scheme is performed on the horizontal rows from top to bottom in order to obtain the numerical approximation of $F(U)_x$. Similarly, the numerical approximation of $E(U)_y$ is obtained applying ENO on the vertical rows from left to right.

ENO schemes and most of the methods to solve conservation laws have been studied in the case of scalar equations. In case of systems, the most frequent approach is the *characteristic based scheme*. This method makes use of the fact that physical systems of conservation laws are in general hyperbolic, that is, all the eigenvalues of the Jacobian matrix of the flux function are real and the matrix is diagonalizable, i.e., there is a complete set of N linearly independent eigenvectors. The characteristic scheme then reduces the original (nonlinear) system into a system of (hopefully) nearly independent scalar equations. These can be independently discretized in an upwind manner and then the system is transformed back into the original variables using the same eigen-decomposition.

Now we are going to explain the characteristic based scheme in more detail. First, let us consider a system like (3.1) but in one spatial dimension:

$$U_t + F(U)_x = 0 \tag{3.2}$$

If the flow is locally smooth we can use the chain rule to expand the derivate in x,

$$U_t + JU_x = 0$$

where $J = \frac{\partial F}{\partial U}$ is the Jacobian matrix of the flux function. If the system is hyperbolic the matrix J has N real eigenvalues λ^p, $p = 1, ..., N$ and its corresponding N eigenvectors. The Jacobian matrix can be diagonalized using the matrices L and R where the eigenvectors are written as rows and columns respectively

$$LJR = \Lambda$$

where, Λ is a diagonal matrix with the N eigenvalues of J (λ_p, $p = 1, ..., N$) as diagonal elements. If we multiply equation (3.2) by the matrix $L(U_i)$ which diagonalizes J locally at point x_i, we obtain

$$[L(U_i)U]_t + [L(U_i)F(U)]_x = 0 \tag{3.3}$$

The vector $L(U_i)U$ is called the vector of *characteristic variables* and $L(U_i)F(U)$ is its corresponding vector of *characteristic fluxes*. The system (3.3) is a system of N independent scalar conservation equations, each of them denoted with the subindex p. Each p-th equation is solved at point x_i with the ENO method in an upwind fashion determined by the sign of the local characteristic velocity. Further on this section, the ENO method is explained in more detail.

Due to the discretization (3.5) of the spatial derivative in (3.2), we will have to estimate the numerical flux function at cell wall points $x_{i+1/2}$. Thus, the Jacobian must be evaluated at these points. The characteristic velocity of the flow of the p-th conserved variable at the cell wall $i + 1/2$ is $\lambda^p(U_{i+1/2})$ with left eigenvector $L^p(U_{i+1/2})$ and right eigenvector $R^p(U_{i+1/2})$. The scalar equation that we will obtain by projecting to the p-th characteristic field is then:

$$w_t + f(w)_x = 0$$

where $w = L^p(U_{i+1/2})U$ and $f(w) = L^p(U_{i+1/2})F(U)$. The velocity term $\lambda^p(U_{i+1/2})$ will play an essential role in the ENO scheme, more precisely for the upwind interpolation, since it determines the direction of propagation. The numerical flux function at the cell wall, $\mathcal{F}_{i+1/2}$, estimated by the ENO method at the p-th characteristic field, is multiplied by its corresponding right eigenvector to recover its value in the original variables,

$$F^p_{i+1/2} = \mathcal{F}_{i+1/2} R^p(U_{i+1/2}) \tag{3.4}$$

We call $F^p_{i+1/2}$ the quantity of the numerical flux $F_{i+1/2}$ that comes from the p-th characteristic field. Adding the contributions of all fields we obtain the estimation of the numerical flux:

$$F_{i+1/2} = \sum_p F^p_{i+1/2}$$

Once we have this estimation of the numerical flux at cell walls, we compute numerically the flux derivative in the following way

$$F(U)_x = \frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} \tag{3.5}$$

where Δx is the spatial resolution.

As observed in [66], the Jacobian matrix of the convective flux vector is quite important to any characteristic based scheme, as it defines the local linearization of the nonlinear problem. It determines the transformation to the local characteristic fields, and thus the local upwind directions, as well as what quantities are to be upwind differenced.

In a typical characteristic-based scheme, only the values of the solution at cell centers are known, while the numerical flux functions are computed at cell boundaries. Some form of reconstruction (usually polynomial) is then required to compute the interface values required by the flux computation. While it makes little difference what sort of (consistent) approximation is used in smooth regions, the situation can change drastically between nodes in an unresolved steep gradient, or when the left and right states have significantly different properties. The results in [66, 59] demonstrate that the use of averaged Jacobians, interpolated from left and right nodal states, can lead to numerically pathological behavior in the approximate solution. Instead, Marquina's flux formula uses the unambiguous values for the left and right reconstructed states ($U^L_{i+1/2}$ and $U^R_{i+1/2}$) to compute *two* characteristic decompositions at each cell interface. The upwind information is extracted from considering non-averaged information, which seems to help in avoiding (or diminishing) the aforementioned pathologies.

In this way, the flux computation (3.4) at the $i + 1/2$ interface at p-th characteristic field is now:

$$F^p_{i+1/2} = \mathcal{F}^L_{i+1/2} R^p(U^L_{i+1/2}) + \mathcal{F}^R_{i+1/2} R^p(U^R_{i+1/2})$$

where $\mathcal{F}^L_{i+1/2}$ and $\mathcal{F}^R_{i+1/2}$ are the numerical fluxes computed in the scalar field resulting from the projected variables and fluxes $L^p(U^L_{i+1/2})U$,$L^p(U^L_{i+1/2})F(U)$ and $L^p(U^R_{i+1/2})U$, $L^p(U^R_{i+1/2})F(U)$ respectively. $L(U^L_{i+1/2})$ and $R(U^L_{i+1/2})$ (resp. $L(U^R_{i+1/2})$ and $R(U^R_{i+1/2})$) are

the matrices of eigenvectors of the Jacobian matrix $J(U^L_{i+1/2})$ ($J(U^R_{i+1/2})$) evaluated at left (right) reconstructed state at the interface $i+1/2$.

ENO-Roe discretization (third order accurate)

ENO reconstruction is based on the idea consisting of finding the flux at cell walls as a derivative of a polynomial interpolation of its primitive, P (more details can be found in [66, 126]). In such an interpolation process, $D^1_i P$ is calculated in the upwind direction. In turn, higher order divided differences are chosen by taking the smallest in absolute value between the two possible choices. These are the relations between derivatives of P and the flux function

$$D^1_i P = \frac{P(x_{i+1/2}) - P(x_{i-1/2})}{\Delta x} = f(w(x_i))$$

$$D^2_{i+1/2} P = \frac{1}{2} D^1_{i+1/2} f$$

$$D^3_i P = \frac{1}{3} D^2_i f$$

Hence, for a specific cell wall located at $x_{i+1/2}$, the scheme for finding the associated numerical flux function of the characteristic variable $\mathcal{F}_{i+1/2}$ consists of the following steps:

1. If $\lambda^p(U_{i+1/2}) > 0$ then $k = i$. Otherwise, $k = i+1$.
   Define $\mathcal{Q}_1(x) = (D^1_k P)(x - x_{i+1/2})$

2. If $|D^2_{k-1/2} P| \leq |D^2_{k+1/2} P|$ then $c = D^2_{k-1/2} P$ and $k^* = k-1$. Otherwise, $c = D^2_{k+1/2} P$ and $k^* = k$.
   Define $\mathcal{Q}_2(x) = c(x - x_{k-1/2})(x - x_{k+1/2})$

3. If $|D^3_{k^*} P| \leq |D^3_{k^*+1} P|$ then $c^* = D^3_{k^*} P$. Otherwise, $c^* = D^3_{k^*+1} P$.
   (Using *Nonoscillatory interpolation* of Harten [86], if $D^3_{k^*} P \, D^3_{k^*+1} P < 0$ then $c^* = 0$).
   Define $\mathcal{Q}_3(x) = c^*(x - x_{k^*-1/2})(x - x_{k^*+1/2})(x - x_{k^*+3/2})$

4. Then, $\mathcal{F}_{i+1/2} = P'(x_{i+1/2}) = \mathcal{Q}'_1(x_{i+1/2}) + \mathcal{Q}'_2(x_{i+1/2}) + \mathcal{Q}'_3(x_{i+1/2})$
   which simplifies to $\mathcal{F}_{i+1/2} = D^1_k P + c(2(i-k)+1)\Delta x + c^*(3(i-k^*)^2 - 1)(\Delta x)^2$

Local Lax-Friedrichs' construction: Entropy Fixed version of the ENO-Roe

The ENO-Roe discretization could create entropy violating expansion shocks near sonic points. In that case and in that specific points, the solution is smoothed slightly

to break up any entropy violating expansion shocks there. So, high order dissipation is added to the calculation of F at the specific cell walls in the vicinity of the sonic point.

The scheme of the Entropy Fix ENO-Roe is:

1. Compute the divided differences of the primitive functions P^+ and P^- defined as:

$$D_i^1 P^{\pm} = \frac{1}{2} f(w_i) \pm \frac{1}{2} \alpha_{i+1/2} w_i$$

   the parameter $\alpha_{i+1/2}$ will be defined later.

2. Compute the second differences $D_{i+1/2}^2 P^{\pm}$ and the third ones $D_i^3 P^{\pm}$ in the standard way, like those of P.

3. For P^+ set $k = i$. Use the rest of ENO-Roe replacing P with P^+ to obtain $\mathcal{F}_{i+1/2}^+$

4. For P^- set $k = i+1$. Use the rest of ENO-Roe replacing P with P^- to obtain $\mathcal{F}_{i+1/2}^-$

5. Then, $\mathcal{F}_{i+1/2} = \mathcal{F}_{i+1/2}^+ + \mathcal{F}_{i+1/2}^-$

Local sided interface states computation

We have already said that this method makes use of the left and right cell wall fluxes in an upwind mode to improve the interpolation at sonic points. Therefore, we need the approximations U^L and U^R of the conserved variables interpolated from the left and right sides of the cell wall.

For each variable v of the vector of conserved variables U, the left and right estimations at cell walls are calculated in the following form:

1. Compute $D_{i+1/2}^1 v$ and $D_i^2 v$ in the standard way, like those of P.

2. For $v_{i+1/2}^L$ set $k = i$. For $v_{i+1/2}^R$ set $k = i + 1$.
   Define $\mathcal{Q}_0(x) = D_k^0 v = v_k$

3. If $|D_{k-1/2}^1 v| \leq |D_{k+1/2}^1 v|$ then $c = D_{k-1/2}^1 v$ and $k^* = k - 1$. Otherwise, $c = D_{k+1/2}^1 v$ and $k^* = k$.
   Define $\mathcal{Q}_1(x) = c(x - x_k)$

4. If $|D_{k^*}^2 v| \leq |D_{k^*+1}^2 v|$ then $c^* = D_{k^*}^2 v$. Otherwise, $c^* = D_{k^*+1}^2 v$.
   (Using *Nonoscillatory interpolation* of Harten [86], if $D_{k^*}^2 v \, D_{k^*+1}^2 v < 0$ then $c^* = 0$).
   Define $\mathcal{Q}_2(x) = c^*(x - x_{k^*})(x - x_{k^*+1})$

5. Then, $v_{i+1/2} = \mathcal{Q}_0(x_{i+1/2}) + \mathcal{Q}_1(x_{i+1/2}) + \mathcal{Q}_2(x_{i+1/2})$
   which simplifies to

$$\begin{aligned} v^L_{i+1/2} &= v_i + \frac{c\Delta x}{2} + c^*((i-k^*)^2 - \frac{1}{4})(\Delta x)^2 \\ v^R_{i+1/2} &= v_{i+1} - \frac{c\Delta x}{2} + c^*((i-k^*)^2 - \frac{1}{4})(\Delta x)^2 \end{aligned}$$

ENO-RF construction: The complete numerical scheme for the spatial term

1. Compute $\lambda^p(U^L_{i+1/2})$ and $\lambda^p(U^R_{i+1/2})$ using the the variables $U^L_{i+1/2}$ and $U^L_{i+1/2}$ obtained by the flux splitting scheme.

2. If $\lambda^p(U^L_{i+1/2}) > 0$ and $\lambda^p(U^R_{i+1/2}) > 0$: the upwind is from the left, then

   $\mathcal{F}^L_{i+1/2}$ using ENO-Roe.
   $\mathcal{F}^R_{i+1/2} = 0$.

3. If $\lambda^p(U^L_{i+1/2}) < 0$ and $\lambda^p(U^R_{i+1/2}) < 0$: the upwind is from the right, then

   $\mathcal{F}^L_{i+1/2} = 0$.
   $\mathcal{F}^R_{i+1/2}$ using ENO-Roe.

4. If $\lambda^p(U^L_{i+1/2})\lambda^p(U^R_{i+1/2}) \leq 0$: there is a sonic point nearby, we define

$$\alpha_{i+1/2} = \max(|\lambda^p(U^L_{i+1/2})|, |\lambda^p(U^R_{i+1/2})|)$$

   Then,

   $\mathcal{F}^L_{i+1/2}$ using entropy fix ENO with $\mathcal{F}^-_{i+1/2} = 0$.
   $\mathcal{F}^R_{i+1/2}$ using entropy fix ENO with $\mathcal{F}^+_{i+1/2} = 0$.

5. $F^p_{i+1/2} = \mathcal{F}^L_{i+1/2} R^p(U^L_{i+1/2}) + \mathcal{F}^R_{i+1/2} R^p(U^R_{i+1/2})$

3.3 Numerical extension in the presence of Source Terms

3.3.1 Introduction

Very often, numerical simulations for non homogeneous systems of the form (2.1) are accomplished by using a fractional splitting technique, in which one alternates between solving the homogeneous system of conservation laws

$$U_t + \mathrm{div}(F(U)) = 0$$

and the system of ordinary differential equations,

$$U_t = S(\mathbf{x}, U).$$

This approach, however, performs very poorly in those situations where U_t is small relative to the other two terms, in particular when steady or quasi-steady solutions are being sought. For such solutions, highly accurate numerical simulations can only be obtained from numerical methods that 'respect' the balance that occurs between the flux gradient and the source term when U_t is small, and it is known [102] that this balance is not likely to be respected when using a fractional step approach.

The shallow water system is used to model real-life applications in which the flow regime is steady or quasi steady, and much effort has been devoted to design numerical techniques that are capable to preserve steady states at the discrete level as well as to accurately compute the evolution of small dynamical perturbations of these. The inclusion of the source term in a direct discretization of system (2.1) becomes then a non-trivial issue. Roe showed very early [124] that pointwise evaluation of the source term is not a suitable strategy, and the necessity of an *upwind* discretization of the source term is now widely recognized [124, 19, 102, 141].

For shallow water flow, the idea of 'source-term upwinding' lead Bermúdez and Vázquez-Cendón [19] to formulate the so-called C-property (for *Conservation* property), which prevents the propagation of parasitic waves in steady and quasi-steady flows. Independently, Greenberg and LeRoux [77] coined the term "well-balanced" for schemes that preserve steady states at the discrete level. These ideas have been explored and developed for shallow water flows in the recent literature [75, 12, 141, 119] and a well established strategy is to combine a conservative scheme for homogeneous conservation laws with an 'appropriate' *upwind* discretization of the source term. The properties of the homogeneous 'solver' are important in the overall performance of the scheme and many of the main homogeneous solvers have been extended to the shallow water system. To name but a few, Bermúdez and Vázquez-Cendón [19] used Roe's scheme as the homogeneous solver. Vukovic and Sopta extended their approach [141] using the ENO and WENO schemes described in [86, 85, 93]. Relaxation schemes have also been considered [12, 57], etc.

In [59] Donat and Marquina develop a new numerical scheme that avoids the use of averaged quantities in computing the numerical flux function at cell interfaces. Marquina's numerical flux formula is based on the use of two jacobian matrices at each interface, from which unmixed, sided, characteristic information is extracted. We seek

to obtain an extension of Marquina's flux formula for non-homogeneous conservation laws by incorporating the idea of flux gradient and source term balancing.

Since the source-term/flux-gradient balancing in [19, 141] is linked to the use of one Jacobian matrix at the interface, we follow a different strategy, described by Gascón and Corberán in [71]. There, the authors propose to write the source term in divergence form so that it can be incorporated into the flux vector of the homogeneous system to be later discretized in an upwind manner. To design the numerical technique to be applied to system (2.1) we proceed as in [126, 59]: we first examine the scalar case and then construct the numerical flux function for the system case by implementing the scalar numerical flux in each (local) characteristic field.

In studying the C-property for our extension of Marquina's scheme, we realize that the *exact* C-property is linked to the use of a unique Jacobian at each cell interface. It is interesting to notice that these 1-Jacobian schemes (1J in the rest of the chapter) can also be considered as a flux-gradient/source-term balanced version of the Shu-Osher ENO schemes in [126], when applied to the shallow water system with topography. The extension turns out to be different from that in [141], since our 1J schemes satisfy the exact C property *independently* of the average interface state, while in [141], the exact C property is only obtained when the average interface state is the arithmetic mean of the left and right interface states.

The use of two Jacobian matrices prevents the scheme from satisfying the *exact* C-property, but the approximate C-property is satisfied as long as the order of accuracy is at least two. Since the use of two Jacobians at cell interfaces is still advantageous in situations where the left and right states are very different, we propose a *combined* 1J-2J scheme which seems to behave well in all cases. In addition, the 2J philosophy serves to design a simple correction that is able to handle dry states (existing or being formed) in a rather straightforward way.

3.3.2 The basic strategy

As described in [66, 59], Marquina's scheme can be interpreted as a *characteristic based scheme* that avoids the use of any artificially constructed averaged quantities at cell interfaces. For full details on the scheme for homogeneous systems (with applications), we refer the reader to [66, 59, 109]. Our aim is to carry out the extension of Marquina's scheme [66, 59] to the shallow water system (2.2) with non-flat bottom topography.

The key step in the extension to (2.2) is the discretization of the source term, which

can be split as follows,

$$S = S_1 + S_2 = \begin{pmatrix} 0 \\ -ghz_x \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -ghz_y \end{pmatrix}$$

We follow [71], and define the functions $B(x,y,t)$ and $C(x,y,t)$ as

$$B(x,y,t) = -\int_{\bar{x}}^{x} S_1(s,y,t)ds = \begin{pmatrix} 0 \\ \beta(x,y,t) \\ 0 \end{pmatrix}, \quad C(x,y,t) = -\int_{\bar{y}}^{y} S_2(x,s,t)ds = \begin{pmatrix} 0 \\ \kappa(x,y,t) \\ 0 \end{pmatrix}$$

with $\beta(x,y,t) = \int_{\bar{x}}^{x} gh(s,y,t)z_x(s,y)ds$ and $\kappa(x,y,t) = \int_{\bar{y}}^{y} gh(x,s,t)z_y(x,s)ds$. Then, we write system (2.2) as follows,

$$U_t + (F+B)_x + (E+C)_y = 0 \tag{3.6}$$

The numerical technique we propose follows the general design principles in [126] and explained in section 3.1. As the spatial terms are treated in a dimension by dimension fashion, it is sufficient to give a detailed description of the technique for the one-dimensional system (2.3). As in [66], we consider first the case of a scalar equation in one dimension.

3.3.3 Scalar equation

Let us consider the scalar, one dimensional version of system (2.3),

$$w_t + f(w)_x = s(x,w). \tag{3.7}$$

Following [71], we define $b(x,t) = \int_{\bar{x}}^{x} -s(y,w(y,t))dy$, and rewrite (3.7) as

$$w_t + g_x = 0 \tag{3.8}$$

with the *combined flux* function $g := f + b$.

We follow a *method of lines* approach and discretize the space variable in (3.8) in a 'conservative' fashion,

$$w_t + \frac{G_{i+1/2} - G_{i-1/2}}{\Delta x} = 0 \tag{3.9}$$

where $G_{i+1/2}$ is the numerical *combined-flux* function.

Dropping the time variable for the sake of simplicity, we note as in [126], that if $\phi(\xi)$ satisfies

$$g(w(x),x) = \frac{1}{\Delta x}\int_{x-\Delta x/2}^{x+\Delta x/2}\phi(\xi)d\xi$$

then

$$g(w(x),x)_x = \frac{1}{\Delta x}\left(\phi(x+\Delta x/2)-\phi(x-\Delta x/2)\right)$$

Thus, we seek to construct the numerical flux function $G_{i+1/2}$ as an approximation to $\phi(x_{i+1/2})$ of the appropriate order of accuracy. Following the construction *via primitive function* [126] we can accomplish this task simply by knowing $g_i = g(w_i, x_i) = f_i + b_i$.

It is well known that 'upwinding' is an essential part of any numerical scheme for hyperbolic conservation laws. In our construction of $G_{i+1/2}$, the upwind direction is determined by the sign of $f'(w)$ in $[x_i, x_{i+1}]$. This choice is simpler from that in [71], where the quantities that determine the upwind directions are numerically computed from g, but it is consistent with other implementations [124, 19, 102, 141] and justified by the examination of the particular case $f(w) = aw$, $s(x,w) = k(x)w$ [124]

$$w(x,t)_t + aw(x,t)_x = k(x)w(x,t) \tag{3.10}$$

whose the solution is

$$w(x,t) = w(x-at,0) + \int_{\bar{x}}^{t} k(x-a(t-s))w(x-a(t-s),s)ds \tag{3.11}$$

and where each term on the RHS exhibits an upwind domain of dependence, which is determined by the wind direction $f'(w) = a$.

With these observations, we propose to use the ENO-RF construction (see section 3.1 and [66, 126] or Algorithm ENO-3 below for specific details) directly on the *combined flux* data $g_i = f_i + b_i$, that is

$$G_{i+1/2} = \begin{cases} \hat{g}^{Roe}_{i+1/2} & \text{if } f' \text{ does not change sign in } [w_i, w_{i+1}] \\ \hat{g}^{LLF}_{i+1/2} & \text{else} \end{cases} \tag{3.12}$$

Here $\hat{g}^{Roe}$ refers to the ENO-Roe numerical flux construction and $\hat{g}^{LLF}$ the 'Local Lax-Friedrichs' construction in [126] (explained in section 3.1). For the sake of completeness and ease of reference, we give the explicit formulas for the third order ENO reconstruction (see also [66]). For the ENO-Roe numerical flux we have

$$\hat{g}^{Roe}_{i+1/2} = g_k + c(2(i-k)+1)\Delta x + c_*(3(i-k_*)^2 - 1)\Delta x^2 \tag{3.13}$$

where the indexes k and k_*, and the quantities c and c_* are computed by the ENO-3 Algorithm.

Algorithm ENO-3

Step 1. If $f' > 0$ then $k = i$

else $k = i + 1$

Step 2. If $|g[k, k+1]| \leq |g[k-1, k]|$ then $k_* = k$, $c = \frac{1}{2}g[k, k+1]$

else $k_* = k - 1$, $c = \frac{1}{2}g[k-1, k]$

Step 3. If $|g[k_*, k_*+1, k_*+2]| \leq |g[k_*-1, k_*, k_*+1]|$ then $c_* = \frac{1}{3}g[k_*, k_*+1, k_*+2]$

else $c_* = \frac{1}{3}g[k_*-1, k_*, k_*+1]$

For the LLF-flux we need to define

$$g_l^+ = 0.5(g_l + \alpha_{i+1/2} w_l), \qquad l = i-2, \ldots, i+2 \tag{3.14}$$

$$g_l^- = 0.5(g_l - \alpha_{i+1/2} w_l), \qquad l = i-1, \ldots, i+3 \tag{3.15}$$

where

$$\alpha_{i+1/2} = \max\{|f'(w)|, w \in [w_i, w_{i+1}]\}$$

then

$$\hat{g}_{i+1/2}^{LLF} = \hat{g}_{i+1/2}^+ + \hat{g}_{i+1/2}^- \tag{3.16}$$

where

$$\hat{g}_{i+1/2}^+ = g_i^+ + c^+ \Delta x + c_*^+ (3(i - k_*^+)^2 - 1)\Delta x^2 \tag{3.17}$$

$$\hat{g}_{i+1/2}^- = g_{i+1}^- - c^- \Delta x + c_*^- (3(i - k_*^-)^2 - 1)\Delta x^2 \tag{3.18}$$

the quantities c^+, k_*^+ and c_*^+ are computed by setting $k = i$ in Step 1 of Algorithm ENO-3 and proceeding with g^+ instead of g as specified in Steps 2 and 3. For $\hat{g}_{i+1/2}^-$ we set $k = i + 1$ in Step 1 and proceed analogously with g^-.

Taking into account formulas (3.13) to (3.18) we see that the numerical flux function $G_{i+1/2}$ in (3.12) can be written as follows:

$$G_{i+1/2} = \mathcal{G}_{i+1/2} + \Delta x C_{i+1/2} + \Delta x^2 D_{i+1/2} = \mathcal{G}_{i+1/2} + HOT_{i+1/2}, \tag{3.19}$$

where the term $\mathcal{G}_{i+1/2}$ collects the first order contribution while the terms containing $C_{i+1/2}$ and $D_{i+1/2}$ are second-order and third-order correction terms. Using equations (3.13) to (3.18) we can write

$$\mathcal{G}_{i+1/2} = \begin{cases} f_i + b_i & \text{if } f' > 0 \text{ in } [w_i, w_{i+1}] \\ f_{i+1} + b_{i+1} & \text{if } f' < 0 \text{ in } [w_i, w_{i+1}] \\ \frac{1}{2}(f_i + f_{i+1}) + \frac{1}{2}(b_i + b_{i+1}) + \frac{1}{2}\alpha_{i+1/2}(w_i - w_{i+1}) & \text{else} \end{cases} \tag{3.20}$$

Notice that given an index l,

$$g[l,l+1] = \frac{g_{l+1}-g_l}{\Delta x} = f[l,l+1] + \frac{1}{\Delta x}\int_{x_l}^{x_{l+1}} -s(w(x),x)dx$$

so that all the terms in $HOT_{i+1/2}$ involve only quantities of the form

$$b_{l,l+1} = \int_{x_l}^{x_{l+1}} -s(w(x),x)dx \tag{3.21}$$

From a numerical point of view, these quantities admit a better numerical treatment than the integral expressions of the form $b_l = \int_{x_{-1/2}}^{x_l} -s(w(x),x)dx$ appearing in (3.20). In order to design a computationally convenient method, we shall deduce an equivalent expression for the flux difference $\mathcal{G}_{i+1/2} - \mathcal{G}_{i-1/2}$ in terms of the quantities $b_{l,l+1}$. Observe that

$$\mathcal{G}_{i+1/2} - \frac{b_i}{2} - \frac{b_{i+1}}{2} = \begin{cases} f_i - \frac{1}{2}b_{i,i+1} & \text{if } f' > 0 \text{ in } [w_i, w_{i+1}] \\ f_{i+1} + \frac{1}{2}b_{i,i+1} & \text{if } f' < 0 \text{ in } [w_i, w_{i+1}] \\ \frac{1}{2}(f_i + \alpha_{i+1/2}w_i) + \frac{1}{2}(f_{i+1} - \alpha_{i+1/2}w_{i+1}) & \text{else} \end{cases} \tag{3.22}$$

where the RHS of expression (3.22) involves only integrals between two consecutive cells, i.e. $b_{l,l+1}$ terms.

Taking into account (3.22) and that $b_{i,j} = b_j - b_i$, we can re-write the first order flux difference as follows,

$$\begin{aligned} \mathcal{G}_{i+1/2} - \mathcal{G}_{i-1/2} &= \left(\mathcal{G}_{i+1/2} - \tfrac{1}{2}b_i - \tfrac{1}{2}b_{i+1}\right) - \left(\mathcal{G}_{i-1/2} - \tfrac{1}{2}b_i - \tfrac{1}{2}b_{i-1}\right) + \tfrac{1}{2}b_{i+1} - \tfrac{1}{2}b_{i-1} \\ &= \left(\mathcal{G}_{i+1/2} - \tfrac{1}{2}b_i - \tfrac{1}{2}b_{i+1} + \tfrac{1}{2}b_{i,i+1}\right) - \left(\mathcal{G}_{i-1/2} - \tfrac{1}{2}b_i - \tfrac{1}{2}b_{i-1} - \tfrac{1}{2}b_{i-1,i}\right) \\ &= \mathcal{G}^+_{i+1/2} - \mathcal{G}^-_{i-1/2} \end{aligned}$$

Notice that the *modified fluxes* $\mathcal{G}^{\pm}$ only involve integral expressions of the form $b_{l,l+1}$. Indeed, carrying out the algebra we easily arrive at

$$\mathcal{G}^+_{i+1/2} = \begin{cases} f_i & \text{if } f' > 0 \text{ in } [w_i, w_{i+1}] \\ f_{i+1} + b_{i,i+1} & \text{if } f' < 0 \text{ in } [w_i, w_{i+1}] \\ \frac{1}{2}(f_i + \alpha_{i+1/2}w_i) + \frac{1}{2}(f_{i+1} - \alpha_{i+1/2}w_{i+1}) + \frac{1}{2}b_{i,i+1} & \text{else} \end{cases} \tag{3.23}$$

$$\mathcal{G}^-_{i+1/2} = \begin{cases} f_i - b_{i,i+1} & \text{if } f' > 0 \text{ in } [w_i, w_{i+1}] \\ f_{i+1} & \text{if } f' < 0 \text{ in } [w_l, w_r] \\ \frac{1}{2}(f_i + \alpha_{i+1/2}w_i) + \frac{1}{2}(f_{i+1} - \alpha_{i+1/2}w_{i+1}) - \frac{1}{2}b_{i,i+1} & \text{else} \end{cases} \tag{3.24}$$

From these expressions, we see that $\mathcal{G}^+_{i+1/2}$ collects a contribution from the source term *only if* there is wind coming from the right at the interface. On the other hand, $\mathcal{G}^-_{i+1/2}$ collects a source term contribution *only if* there is wind coming from the left at the interface. Notice that

$$\mathcal{G}^+_{i+1/2} - \mathcal{G}^-_{i+1/2} = b_{i,i+1}$$

hence the *signed* numerical fluxes, $\mathcal{G}^\pm_{i+1/2}$, provide an effective *upwind* splitting of the source term contribution at the $i+1/2$ interface

It is illustrative to write down the flux difference for the particular case (3.10). For $a > 0$ (analogously for $a < 0$) we have

$$\mathcal{G}_{i+1/2} - \mathcal{G}_{i-1/2} = \mathcal{G}^+_{i+1/2} - \mathcal{G}^-_{i-1/2} = f_i - f_{i-1} + \int_{x_{i-1}}^{x_i} s(z, w(z))dz$$

this expression clearly displays the *upwind* contribution of the source term.

When the flux $f(w)$ is nonlinear, the expressions above provide easily computable numerical flux functions that incorporate an upwind source-term contribution. We refer to Figure 3.1 for a better understanding of which cell contains source term contributions, according to the sign of f'.

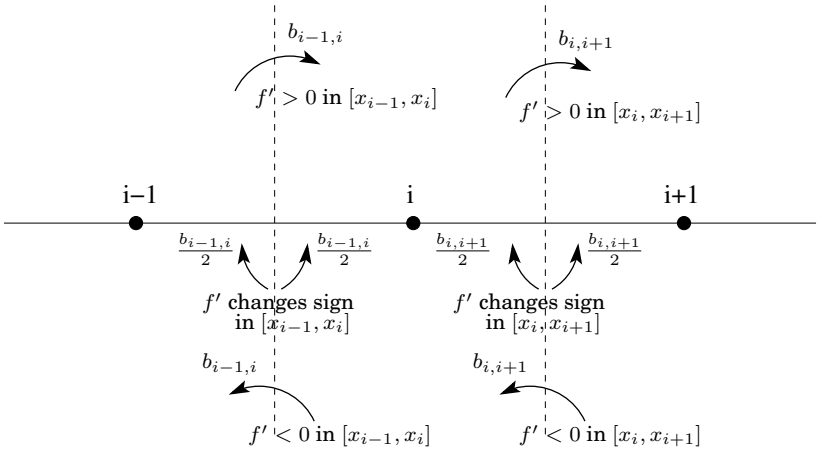


Figure 3.1: Contribution of source terms (at first order) involving the cell i and its both contiguous cells according to the sign of the characteristic velocities at the vicinity of each cell wall.

To obtain a high resolution scheme, let us define

$$G^+_{i+1/2} = \mathcal{G}^+_{i+1/2} + HOT_{i+1/2}, \qquad G^-_{i+1/2} = \mathcal{G}^-_{i+1/2} + HOT_{i+1/2}$$

where $HOT_{i+1/2} = G_{i+1/2} - \mathcal{G}_{i+1/2}$ are the higher order terms obtained from the polynomial expressions (3.13),(3.17) and (3.18). Thus

$$G_{i+1/2} - G_{i-1/2} = G^+_{i+1/2} - G^-_{i-1/2} \tag{3.25}$$

Using (3.25), the semi-discrete formulation (3.9) can be equivalently written as

$$w_t + \frac{G^+_{i+1/2} - G^-_{i-1/2}}{\Delta x} = 0 \tag{3.26}$$

where the computation of the combined fluxes $G^{\pm}_{i+1/2}$ only involve integrals of the source term between consecutive cell centers. Equation (3.25) will be the starting point for our extension to the system case.

A fully discrete expression is obtained by numerically approximating the integral terms $b_{i,i+1}$. It should be noted that the numerical integration technique employed here has an influence on the global order of accuracy. In the next section we shall describe our choice for the shallow water case.

3.3.4 Extension to nonlinear systems

To carry out the extension to the 1D system (2.3), we write it in the form $U_t + G_x = 0$, with $G(U,x) = F(U) + B(U,x)$ and $B(U,x) = (0, \int_{\bar{x}}^{x} ghz_x ds)^T$. We propose to use a semi-discrete formulation of the type

$$U_t + \frac{G^+_{i+1/2} - G^-_{i-1/2}}{\Delta x} = 0, \tag{3.27}$$

where the computation of $G^{\pm}_{i+1/2}$ only involves integral terms over consecutive cell centers and follows the basic design strategy in Marquina's flux formula: two states are computed at each side of a cell-interface, U^L and U^R, and the numerical flux functions are obtained by applying the scalar algorithm to "sided" local characteristic fluxes.

The states U^L and U^R at each side of a given interface are obtained by ENO interpolation of the physical variables as specified in section 3.1 and [66]. Unless specifically stated, the order of the interpolation used to compute these states is the same as the order of the scheme.

Given $U^L = U^L_{i+1/2}$ and $U^R = U^R_{i+1/2}$, the left and right states at the $i+1/2$ cell-interface, the flux functions $G^{\pm}_{i+1/2}$ shall be defined as

$$G^{\pm}_{i+1/2} = \sum_{p=1}^{2} (\tilde{G}^{\pm}_{i+1/2})^{p,L} R^p(U^L) + (\tilde{G}^{\pm}_{i+1/2})^{p,R} R^p(U^R) \tag{3.28}$$

where $L^p(U^L)$, $R^p(U^L)$ ($L^p(U^R)$, $R^p(U^R)$), $p = 1, 2$, are the left and right eigenvectors of the Jacobian matrix $J(U) = F'(U)$, associated to the eigenvalues $\lambda^p(U^L)$ ($\lambda^p(U^R)$), and the local *modified* characteristic fluxes $(\tilde{G}^{p,\pm})^{L,R}$ are computed *as in the scalar case*. We give next a precise description of the computation of these numerical fluxes. In what

follows $B_{j,j+1} = B_{j+1} - B_j = (0, \int_{x_j}^{x_{j+1}} ghz_x ds)^T$. In the fully discrete scheme, the integral in the second component is substituted by the discrete expression derived in the next section. We recall that for a scheme of order r ($r = 1, 2$ or 3 in this work), the index j below runs from $j = i - r, \ldots, i + r$.

2J-Numerical flux function: For $p = 1, 2$

- If $\lambda^p(U^L) > 0$ and $\lambda^p(U^R) > 0$ then wind from the left: use ENO-Roe construction
  - Compute the first order contributions, $\mathcal{G}^{\pm}_{i+1/2}$, from (3.23) and (3.24), setting $f' > 0$; $f_j = L^p(U^L) \cdot F_j$, $b_{j,j+1} = L^p(U^L) \cdot B_{j,j+1}$.
  - Compute the $HOT^L_{i+1/2}$ terms using Algorithm ENO-3 with $k = i$, $g_j = L^p(U^L) \cdot (F_j + B_j)$.

  Then define
  $$(\tilde{G}^{p,\pm}_{i+1/2})^L = \mathcal{G}^{\pm}_{i+1/2} + HOT^L_{i+1/2}, \qquad (\tilde{G}^{p,\pm}_{i+1/2})^R = 0.$$
- If $\lambda^p(U^L) < 0$ and $\lambda^p(U^R) < 0$ then wind from the right: Use ENO-Roe construction.
  - Compute the first order contributions, $\mathcal{G}^{\pm}_{i+1/2}$, from (3.23) and (3.24), setting $f' < 0$; $f_j = L^p(U^R) \cdot F_j$, $b_{j,j+1} = L^p(U^R) \cdot B_{j,j+1}$.
  - Compute the $HOT^R_{i+1/2}$ terms using Algorithm ENO-3 with $k = i + 1$, $g_j = L^p(U^R) \cdot (F_j + B_j)$

  Then define
  $$(\tilde{G}^{p,\pm}_{i+1/2})^L = 0, \qquad (\tilde{G}^{p,\pm}_{i+1/2})^R = \mathcal{G}^{\pm}_{i+1/2} + HOT^R_{i+1/2};$$
- If $\lambda^p(U^L)\lambda^p(U^R) \leq 0$ then mixed wind: Use the LLF construction.
  - Define $\alpha_{i+1/2} = \max(|\lambda^p(U^L)|, |\lambda^p(U^R)|)$
  - Define
    $(\tilde{G}^{p,+}_{i+1/2})^L = L^p(U^L) \cdot \frac{1}{2}\left(F_i + \alpha_{i+1/2}U_i\right) + HOT^L_{i+1/2}$
    $(\tilde{G}^{p,-}_{i+1/2})^L = L^p(U^L) \cdot \frac{1}{2}\left(F_i + \alpha_{i+1/2}U_i - B_{i,i+1}\right) + HOT^L_{i+1/2}$
    Compute the $HOT^L_{i+1/2}$ terms as in (3.17) with $g^+_j = \frac{1}{2}L^p(U^L)(F_j + B_j + \alpha_{i+1/2}U_j)$.
    $(\tilde{G}^{p,+}_{i+1/2})^R = L^p(U^R) \cdot \frac{1}{2}\left(F_{i+1} - \alpha_{i+1/2}U_{i+1} + B_{i,i+1}\right) + HOT^R_{i+1/2}$
    $(\tilde{G}^{p,-}_{i+1/2})^R = L^p(U^R) \cdot \frac{1}{2}\left(F_{i+1} - \alpha_{i+1/2}U_{i+1}\right) + HOT^R_{i+1/2}$
    Compute the $HOT^L_{i+1/2}$ terms as in (3.18) with $g^-_j = \frac{1}{2}L^p(U^R)(F_j + B_j - \alpha_{i+1/2}U_j)$.

The extension just described complies with the basic design principles of Marquina's flux formula: the superscript L refers to characteristic information carried by a left-wind, while R refers to right-wind driven information. On the other hand, the superscripts $\pm$ in the local characteristic fluxes refer to the main (first-order) source term contribution in each characteristic field. As in the scalar case, the $+$ fluxes ($-$ fluxes) in the pth field collect source term contributions only if there is wind coming from the right (left) at the interface. [1]

We also remark that when $U^L = U^R$ (for example when $U^L = U^R = U^A$ an average interface state), then the 2J-numerical flux function just described becomes a 1J numerical flux and (3.28) reduces to

$$G^{\pm}_{i+1/2} = \sum_p (\tilde{G}^{p,\pm}_{i+1/2})^A R^p(U^A_{i+1/2}) \tag{3.29}$$

This 1J-numerical flux becomes an extension of the ENO numerical fluxes in [126] to the shallow water system. As we shall see shortly, this extension turns out to be different from that in [141].

Numerical approximation of the source term contributions

For the shallow water system, the source term contribution involves the numerical approximation of integrals such as $\int_{x_i}^{x_{i+1}} ghz_x dx$. Assuming that the integrand is smooth and applying the trapezoidal rule

$$\beta_{i,i+1} = \int_{x_i}^{x_{i+1}} ghz_x dx = g[(hz_x)_i + (hz_x)_{i+1}]\frac{\Delta x}{2} - (hz_x)_{xx}(\xi)\frac{\Delta x^3}{12} \tag{3.30}$$

If the topography and the flow are smooth, we can write

$$(hz_x)_i = h_i(z_x)_i = h_i\left(\frac{z_{i+1}-z_i}{\Delta x} - \frac{\Delta x}{2}(z_{xx})_i + O(\Delta x^2)\right)$$
$$(hz_x)_{i+1} = h_{i+1}(z_x)_{i+1} = h_{i+1}\left(\frac{z_{i+1}-z_i}{\Delta x} + \frac{\Delta x}{2}(z_{xx})_{i+1} + O(\Delta x^2)\right)$$

Replacing these terms in (3.30) we obtain

$$\begin{aligned}\beta_{i,i+1} &= \frac{g}{2}(z_{i+1}-z_i)(h_{i+1}+h_i) + \frac{\Delta x^2}{4}g(-h_i(z_{xx})_i + h_{i+1}(z_{xx})_{i+1}) + O(\Delta x^3)\\ &= \frac{g}{2}(z_{i+1}-z_i)(h_{i+1}+h_i) + O(\Delta x^3) = \bar{\beta}_{i,i+1} + O(\Delta x^3)\end{aligned}$$

[1] We have also tested $(\tilde{G}^{p,\pm}_{i+1/2})^L = L^p(U^L)\cdot\frac{1}{2}\left(F_i + \alpha_{i+1/2}U_i \pm \frac{1}{2}B_{i,i+1}\right) + HOT^L_{i+1/2}$ and $(\tilde{G}^{p,\pm}_{i+1/2})^R = L^p(U^R)\cdot\frac{1}{2}\left(F_{i+1} - \alpha_{i+1/2}U_{i+1} \pm \frac{1}{2}B_{i,i+1}\right) + HOT^R_{i+1/2}$ without finding differences in the experimental results.

Thus, $\bar{\beta}_{i,i+1} = \frac{g}{2}(z_{i+1} - z_i)(h_{i+1} + h_i)$ provides a third order approximation to the integral of the source term between two consecutive cells, for smooth flows and smooth topography. This is compatible with the third order accurate flux computations carried out by the ENO-3 algorithm described above and is the choice used in our numerical schemes.

Study of the C-property

Numerical schemes specifically designed for the simulation of shallow water flows must be able to compute accurately steady states and small dynamical perturbations of these. Schemes that have this property are named *well balanced* by Le Roux and collaborators [77]. In [19], Bermúdez and Vázquez Cendón identify a key property that the scheme must satisfy in order to prevent the formation of numerically driven parasitic waves. This is the so called C property.

A scheme is said to satisfy the exact C-property if it is exact when applied to the stationary case $q \equiv 0$ and $h + z \equiv constant$ and if it is not exact but accurate to order $O(\Delta x^2)$ it is said to satisfy the approximate C-property.

To determine the behaviour of our 2J scheme with respect to the C-property, we study the flux difference $G^+_{i+1/2} - G^-_{i-1/2}$ for the quiescent stationary solution $q = 0, h + z = C$. Notice that in this case $\nabla(z + h) = 0$, thus

$$F(U) = \begin{pmatrix} 0 \\ \frac{1}{2}gh^2 \end{pmatrix} \quad B(U,x) = \begin{pmatrix} 0 \\ \int_{\bar{x}}^{x} ghz_x ds \end{pmatrix} = \begin{pmatrix} 0 \\ -\int_{\bar{x}}^{x} ghh_x ds \end{pmatrix}$$

hence

$$B_j = \begin{pmatrix} 0 \\ -\frac{g}{2}h_j^2 + constant \end{pmatrix} \quad G_j = F_j + B_j = \begin{pmatrix} 0 \\ constant \end{pmatrix}$$

and

$$B_{i,i+1} = B_{i+1} - B_i = \begin{pmatrix} 0 \\ -\frac{g}{2}(h_i^2 - h_{i+1}^2) \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{g}{2}(z_{i+1} - z_i)(h_{i+1} + h_i)) \end{pmatrix},$$

since $h_i + z_i = h_{i+1} + z_{i+1}$. This implies that that $\beta_{i,i+1} = \bar{\beta}_{i,i+1}$, i.e. the numerical approximation to the integral values are exact.

For quiescent steady state flows, the eigenvalues of the one dimensional system are: $\lambda_1(U) = -c$ and $\lambda_2(U) = c = \sqrt{gh}$, and the right and left eigenvectors $R^m(U) = (1, \lambda_m(U))^T$, $L^m(U) = \frac{1}{2}(1, \lambda_m^{-1}(U))$.

When $h > 0$, $\lambda_1 < 0$ and $\lambda_2 > 0$ always, and the 2J-algorithm easily leads to:

$$G^+_{i+1/2} - G^-_{i-1/2} = (\tilde{G}^{1,+}_{i+1/2})^R R^{1,R}_{i+1/2} + (\tilde{G}^{2,+}_{i+1/2})^L R^{2,L}_{i+1/2} - (\tilde{G}^{1,-}_{i-1/2})^R R^{1,R}_{i-1/2} - (\tilde{G}^{2,-}_{i-1/2})^L R^{2,L}_{i-1/2}$$

Since G_j is constant $\forall j$, all divided differences of the characteristic fluxes: $g_j^{p,L} = L^{p,L}G_j$ and $g_j^{p,R} = L^{p,R}G_j$ for $p = 1, 2$ are zero, hence all HOT terms are zero. In addition, the eigenvalues do not change sign, i.e. the flux functions are always computed with the ENO-Roe Algorithm and one can easily check that

$$(\tilde{G}^{1,+}_{i+1/2})^R = L^{1,R}_{i+1/2}(F_{i+1} + B_{i,i+1}) = -\frac{1}{4}g\frac{h_i^2}{c^R_{i+1/2}} \qquad (\tilde{G}^{2,+}_{i+1/2})^L = L^{2,L}_{i+1/2}F_i = \frac{1}{4}g\frac{h_i^2}{c^L_{i+1/2}}$$

$$(\tilde{G}^{2,-}_{i-1/2})^L = L^{2,L}_{i-1/2}(F_{i-1} - B_{i-1,i}) = \frac{1}{4}g\frac{h_i^2}{c^L_{i-1/2}} \qquad (\tilde{G}^{1,-}_{i-1/2})^R = L^{1,R}_{i-1/2}F_i = -\frac{1}{4}g\frac{h_i^2}{c^R_{i-1/2}}$$

Then, carrying out the algebra we get

$$\begin{pmatrix} \frac{G^+_{i+1/2} - G^-_{i-1/2}}{\Delta x} \end{pmatrix} = \frac{gh_i^2}{4\Delta x}\begin{pmatrix} -\frac{1}{c^R_{i+1/2}} + \frac{1}{c^L_{i+1/2}} + \frac{1}{c^R_{i-1/2}} - \frac{1}{c^L_{i-1/2}} \\ 0 \end{pmatrix} \tag{3.31}$$

We can observe that all terms in the second component cancel out. For the first component, assuming that h is a smooth function, the ENO interpolation procedure of order r would lead to the following estimate

$$h^{L,R}_{i+1/2} = h(x_{i+1/2}) + O(\Delta x^r) \tag{3.32}$$

Hence, for $h > 0$ and sufficiently smooth, we can conclude that

$$(h^{L,R}_{i+1/2})^{-1/2} = (h(x_{i+1/2}))^{-1/2} + O(\Delta x^r) \quad \Rightarrow \quad (c^{L,R}_{i+1/2})^{-1} = (c(x_{i+1/2}))^{-1} + O(\Delta x^r)$$

so that the approximate C-property is satisfied provided that $r \geq 3$.

Notice that for the first order scheme we have

$$\begin{pmatrix} \frac{G^+_{i+1/2} - G^-_{i-1/2}}{\Delta x} \end{pmatrix} = \frac{gh_i^2}{4\Delta x}\begin{pmatrix} -\frac{1}{c_{i+1}} + \frac{1}{c_i} + \frac{1}{c_i} - \frac{1}{c_{i-1}} \\ 0 \end{pmatrix} = \frac{gh_i^2}{4\Delta x}\begin{pmatrix} -\left(\frac{1}{c}\right)_{xx}\Delta x^2 \\ 0 \end{pmatrix} = \begin{pmatrix} O(\Delta x) \\ 0 \end{pmatrix}$$

Thus, it seems reasonable to expect that second order accuracy would also ensure the approximate C-property, and this seems to be so computationally (see section 3.4), but we cannot ensure it analytically.

Notice that if $U^L = U^R$, i.e. we use only one Jacobian at each cell interface, the first term in (3.31) also cancels out, and that this cancellation occurs *independently of the interface state* and of the order of the scheme. Therefore, our 1J (one-Jacobian) scheme verifies the exact C-property. As mentioned in the introduction, the 1J scheme can be considered as a flux gradient-source term balanced extension of the ENO schemes in [126], different from that in [141].

Remark 1 *Our analysis shows that the cancellations necessary to obtain the C property must be accomplished at each cell boundary. The two essential ingredients at the $i+1/2$ interface are: a) only one Jacobian is computed at the interface and b) $\bar{\beta}_{i,i+1} = \beta_{i,i+1}$.*

Combined 1J-2J scheme

In light of the remarks above, It would seem that the use of two Jacobian evaluations at cell walls is not to be recommended for numerical simulations of steady or quasi-steady shallow water flows, unless the scheme is at least third order accurate. However, we have found that using two jacobians can still be advantageous in certain situations that are of interest in shallow water flow and that involve two very different states at both sides of a numerical interface (see experiments: Transcritical flow with shock in section 3.4.1 and the ones in section 3.4.3.

We then propose a scheme that combines the use of one or two Jacobians, when either choice is most appropriate: When the interpolated states at each cell-wall, $U^L_{i+1/2}$ and $U^R_{i+1/2}$, are 'close', then, a single Jacobian decomposition is used at the $i+1/2$ interface. If there is a significant difference between the left and right states at the interface we construct the combined flux function using two Jacobian decompositions. However, if there is a sonic point nearby, in practice when $\lambda^p(U_i)\lambda^p(U_{i+1}) \leq 0$, then even if the states are close, we still use the two sided decomposition. In this way, we expect to get both the benefits of *essentially* satisfying the exact C-property, by using mostly the 1J-numerical flux, and of the 2J scheme, at a few particular locations where it is known that Marquina's flux splitting technique has a better performance.

The combined 1J-2J numerical flux function we propose can be explicitly described as follows,

Algorithm 1J-2J : At the $i+1/2$ interface,

- Compute $U^L = U^L_{i+1/2}$ and $U^R = U^R_{i+1/2}$ with left and right-biased ENO interpolation, as in [66].
- Compute the eigenvalues $\lambda^p(U^L)$, $\lambda^p(U^R)$ for $p = 1, 2$.
- If $||U^L - U^R|| < \Delta x^s$ [2] and $\lambda^p(U^L)\lambda^p(U^R) > 0$ for $p = 1$ and 2 then (contiguous states are very close and no sonic point nearby: 1J-numerical flux)

[2] s is a power less than the order of the scheme r. In practice, we use $s = 1/2$ for $r = 1$ and $s = 1$ for $r = 2, 3$.

Define the average state at the interface, e.g.[3] $U^A_{i+1/2} = \frac{1}{2}(U^L + U^R)$

$$G^{\pm}_{i+1/2} = \sum_p (\tilde{G}^{p,\pm}_{i+1/2})^A R^p(U^A_{i+1/2})$$

- Else, (contiguous states not close, or sonic nearby: 2J-numerical flux)

$$G^{\pm}_{i+1/2} = \sum_p (\tilde{G}^{p,\pm}_{i+1/2})^L R^p(U^L) + (\tilde{G}^{p,\pm}_{i+1/2})^R R^p(U^R)$$

3.3.5 Treatment of dry zones

One of the major problems that arises when simulating the movement of a fluid is the presence/occurrence of dry zones. Dry areas in shallow-water flow simulations can be handled in various ways, but Toro [134] warns that converting dry areas into *slightly wet* areas is both inappropriate and numerically 'dangerous' (see [134] for details).

Front tracking seems to be an adequate procedure but, since it is difficult to implement in several dimensions, other alternatives have also been explored in the literature (see [33, 32],[129, 147], [70] and references therein).

In this work we follow the approach in [31]. To deal with existing dry areas, we give a threshold of minimum water depth below which h and q are set to zero and the cell is considered a dry cell. Wetting fronts are then included in the ordinary cell procedure in a through calculation that assumes zero water depth for the dry cells. The key point in this approach is the use of a numerical flux function that can cope with zero-depth cells while maintaining, at the same time, the C-property.

Since the advancement of the wet/dry front should be determined by the wet state at the front, we adopt the following special treatment at cell-walls separating wet and dry states.

- If $h_i \neq 0$ and $h_{i+1} = 0$: (*wet*/*dry* front) then
  - Compute $U^L_{i+1/2}$ with ENO interpolation. Set $U^R_{i+1/2} = U^L_{i+1/2}$.
- If $h_i = 0$ and $h_{i+1} \neq 0$: (*dry*/*wet* front) then
  - Compute $U^R_{i+1/2}$ with ENO interpolation. Set $U^L_{i+1/2} = U^R_{i+1/2}$.

The interface flux always uses one Jacobian, evaluated at the wet state at the interface, hence the first condition to fulfill the exact C property is respected (see Remark 1).

[3] in our numerical simulations we have found no difference between the arithmetic mean or the Roe mean.

Source term at wet/dry fronts

Notice that the discrete expression $\bar{\beta}_{i,i+1}$ in section 1 is not the exact value of the integral source term contribution $\beta_{i,i+1}$ at the boundary between a wet and a dry cell. In order to fulfill the second condition in Remark 1 we re-derive the discrete value of the integral contribution taking into account the existence of a dry cell.

We consider first a *wet*/*dry* interface (i.e. $h_i \neq 0$, $h_{i+1} = 0$). Let x_c be the the point where h becomes null. Then,

$$\beta_{i,i+1} = \int_{x_i}^{x_{i+1}} ghz_x dx = \int_{x_i}^{x_c} ghz_x dx = \frac{g}{2}(h_c + h_i)(z_c - z_i) + O(\Delta x^3) \tag{3.33}$$

where $h_c = h(x_c) = 0$ and $z_c = z(x_c)$. Let us assume that z and h are sufficiently smooth in $[x_i, x_c]$, then:

$$h_c = h_i + h_x(i)(x_c - x_i) + O(\Delta x^2) \tag{3.34}$$

$$z_c = z_i + z_x(i)(x_c - x_i) + O(\Delta x^2) \tag{3.35}$$

Assuming that $h_x(i) = O(1)$ we can write

$$z_c = z_i - z_x(i)\frac{h_i}{h_x(i)} + O(\Delta x^2) \tag{3.36}$$

Replacing z_c from (3.36) in (3.33), we obtain

$$\beta_{i,i+1} = -\frac{g}{2}\frac{z_x(i)}{h_x(i)}h_i^2 + O(\Delta x^2) \tag{3.37}$$

To obtain a discrete approximation, we substitute $z_x(i)$ and $h_x(i)$ by discrete estimations from 'wet' variables at cell centers, i.e.

$$\bar{\beta}_{i,i+1} = -\frac{g}{2}\frac{z_i - z_{i-1}}{h_i - h_{i-1}}h_i^2 \tag{3.38}$$

This approximation provides a discrete value for the source term contribution that satisfies the second condition in remark 1: If $h_i + z_i = h_{i-1} + z_{i-1}$ at the *wet* side, then $\bar{\beta}_{i,i+1} = \beta_{i,i+1} = -gh_i^2/2$. Hence the numerical flux function complies with the necessary conditions to obtain the cancellations necessary for the C-property.

For a *dry*/*wet* boundary, an analogous computation leads easily to

$$\beta_{i,i+1} = \frac{g}{2}\frac{z_x(i+1)}{h_x(i+1)}h_{i+1}^2 + O(\Delta x^2) \tag{3.39}$$

As before, approximating the derivatives from *wet* values leads to the following discrete source term contribution (that satisfies also the second condition in remark 1)

$$\bar{\beta}_{i,i+1} = \frac{g}{2}\frac{z_{i+2} - z_{i+1}}{h_{i+2} - h_{i+1}}h_{i+1}^2 \tag{3.40}$$

Note that in the redefinition of the integral (3.33) x_c is a point between x_i and x_{i+1} so the possible values of h_x in (3.34) are restricted to

$$h_x(i) \leq -\frac{h_i}{\Delta x} \text{ in the wet/dry front} \tag{3.41}$$

$$h_x(i+1) \geq \frac{h_{i+1}}{\Delta x} \text{ in the dry/wet front} \tag{3.42}$$

If inequality (3.41) (or (3.42)) does not hold, we assume that $x_c = x_i$ ($x_c = x_{i+1}$) and we use $\bar{\beta}_{i,i+1} = \frac{g}{2}(h_{i+1}+h_i)(z_{i+1}-z_i)$ as usually. Note also that inequalities (3.41) and (3.42) avoid automatically the particular cases $h_x(i) = 0$ and $h_x(i+1) = 0$.

Occurrence of Dry states from Wet flow

In section 2.2.4 we recall that dry areas appear in the solution of Riemann problems for shallow water flows with flat topography when the *depth positivity condition*

$$u^R - u^L \leq 2c^L + 2c^R$$

is not satisfied. As noticed by Toro [134] the occurrence of dry areas is particularly hard to simulate with conventional *wet* Riemann solvers.

Throughout our numerical experimentation we have found that the type of viscosity ($\alpha_{i+1/2}$) considered in the LLF portion of Marquina's flux splitting formula is most important when dealing with the numerical formation of dry areas. Moreover, the way in which the interface states are computed seems to be also very important: U^L and U^R should be computed avoiding any mixed information. In numerical simulations involving dry states formed from wet states by rarefaction separation we have found a numerical treatment which seems to give consistent and reliable numerical results in all cases. The numerical treatment is as follows,

If $u_i - u_{i+1} > 2c_i + 2c_{i+1}$ holds,

- Compute U^L by an ENO interpolation *considering that the states at the right of the* $i+1/2$ *interface are dry cells*.
- Compute U^R by an ENO interpolation *considering that the states to the left of the interface are dry cells*.
- For every p-th field the LLF construction is used substituting the numerical viscosity $\alpha_{i+1/2} = \max(|\lambda^p(U^L)|, |\lambda^p(U^R)|)$ by the following definition

$$\alpha_{i+1/2} := \frac{\hat{\lambda}^p_{i+1/2}(\lambda^p(U^R) + \lambda^p(U^L)) - 2\lambda^p(U^R)\lambda^p(U^L)}{\lambda^p(U^R) - \lambda^p(U^L)} \tag{3.43}$$

where $\hat{\lambda}^p_{i+1/2}$ is the p-th eigenvalue of the Jacobian matrix evaluated at the Roe's average of contiguous states, i.e.

$$\hat{h}_{i+1/2} = \sqrt{h_i h_{i+1}}, \quad \hat{c}_{i+1/2} = \sqrt{g\frac{h_i + h_{i+1}}{2}} \quad \text{and} \quad \hat{u}_{i+1/2} = \frac{\sqrt{h_i}u_i + \sqrt{h_{i+1}}u_{i+1}}{\sqrt{h_i} + \sqrt{h_{i+1}}}$$

The choice for $\alpha_{i+1/2}$ is inspired in the entropy fix for Roe's scheme proposed by Harten and Hyman. For the scalar conservation law, the entropy-fixed flux can be written as ([101])

$$\begin{aligned} \tilde{F}(w^L, w^R) &= f^L + \lambda^L \frac{\lambda^R - \hat{\lambda}}{\lambda^R - \lambda^L}(w^R - w^L) \\ &= f^R - \lambda^R \frac{\hat{\lambda} - \lambda^L}{\lambda^R - \lambda^L}(w^R - w^L) \end{aligned} \tag{3.44}$$

where $\hat{\lambda}$ is a characteristic speed at the interface. Adding the two expressions for the RHS we get

$$\tilde{F}(w^L, w^R) = \frac{1}{2}(f^L + \alpha w^L) + \frac{1}{2}(f^R - \alpha w^R)\ ; \quad \alpha = \frac{\hat{\lambda}(\lambda^R + \lambda^L) - 2\lambda^R\lambda^L}{\lambda^R - \lambda^L}$$

For systems, $\hat{\lambda}^p$ is the pth eigenvalue of the Roe-matrix constructed from w^L and w^R. In figure 3.11(a) we show an example of the type of instabilities that occur when these corrections are not implemented.

3.4 Experiments

The following series of numerical experiments illustrate various features of the scheme. The test cases are standard in the literature. In all numerical simulations we use a CFL coefficient of 0.8 and a threshold of water depth equal to 10^{-4}.

3.4.1 Steady flow: Relevance of the C-property

Maintaining Steady flow

To give a numerical validation of our study on the C-property, we follow [141] and consider a smooth topography in a $20m$ channel given by

$$z(x) = 0.2e^{-(\frac{2}{5}(x-10)^2)}$$

and the two initial states shown in Figure 3.2. The left plot corresponds to a quiescent state ($u = 0$) and the right plot to a non-quiescent state with constant discharge 4.42 m^2/s. In table 3.1, we show measurements of the L^1-error and the numerical order at

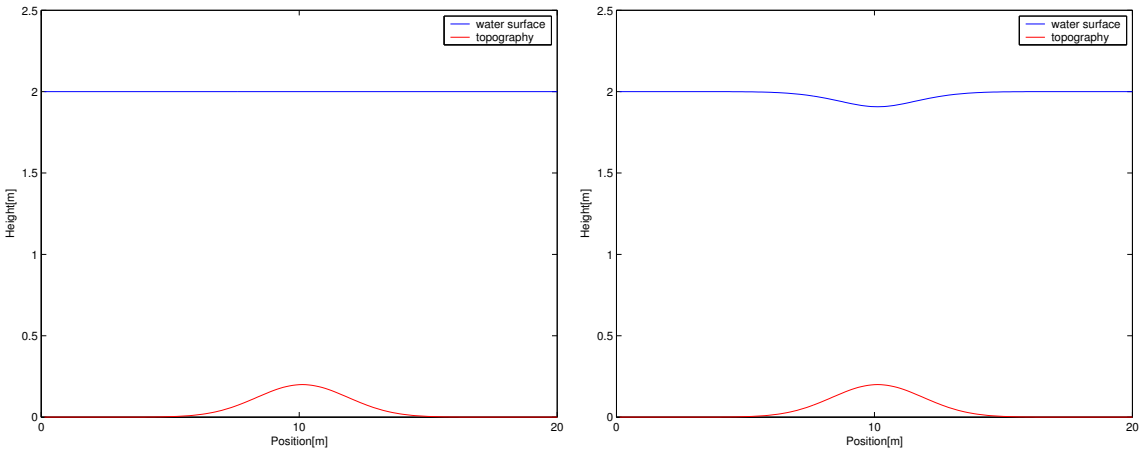


Figure 3.2: Exact water surface and topography at steady state (left: quiescent state, right: non-quiescent state).

$T = 50s$ when using the 2J-scheme. We can see that the order of accuracy obtained in the quiescent simulation agrees with the theoretical analysis in section 1: the global error behaves like $O(\Delta x)$ for $r = 1$, and it seems to be of order larger than 2 for the third order scheme. It also shows that the second order method satisfies the approximate C-property although we could not ensure it theoretically. We obtain similar results for the non-quiescent case, as shown in table 3.1, even though no analytical study was presented for this case.

If we repeat the calculations using the 1J-2J scheme, then the error is within machine precision, i.e. the scheme satisfies the exact C-property, in the quiescent case. The table for the non-quiescent case deserves some explanation: We can observe that all the errors in this case are the same for any r. This $O(\Delta x^2)$ error actually comes from the trapezoidal rule used to approximate the integral source term contribution. We have verified experimentally that if we increase the order of the numerical integration, using for example Simpson's rule, the approximation order of the non-quiescent state is improved. We thus conjecture that the 1J-scheme preserves *all* steady states up to the order of the numerical integration procedure.

Usually, the bottom topography is not smooth. With the objective of evaluating the discretization of the source term in the presence of complex and possibly non-smooth geometry, the following experiment was proposed in a workshop on dam-break wave simulation [76]. The topography is tabulated in [76] and shown in figure 3.3. The initial condition is the water at rest at a level of 12m. The boundary conditions are a water level of 12m and no discharge. Numerical results obtained after a simulation of 200s

QUIESCENT STATE

cells	n=1		n=2		n=3	
	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$
20		$1.9 \cdot 10^{-3}$		$5.47 \cdot 10^{-4}$		$2.62 \cdot 10^{-4}$
40	1.57	$6.48 \cdot 10^{-4}$	2.47	$9.84 \cdot 10^{-5}$	1.85	$7.24 \cdot 10^{-5}$
80	1.35	$2.53 \cdot 10^{-4}$	2.33	$1.95 \cdot 10^{-5}$	1.52	$2.52 \cdot 10^{-5}$
160	1.10	$1.17 \cdot 10^{-4}$	1.89	$5.26 \cdot 10^{-6}$	3.14	$2.84 \cdot 10^{-6}$
320	0.99	$5.95 \cdot 10^{-5}$	1.37	$2.02 \cdot 10^{-6}$	3.96	$1.81 \cdot 10^{-7}$

NON-QUIESCENT STATE

cells	n=1		n=2		n=3	
	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$
20		$1.7 \cdot 10^{-3}$		$7.26 \cdot 10^{-4}$		$2.92 \cdot 10^{-4}$
40	0.97	$8.46 \cdot 10^{-4}$	2.04	$1.76 \cdot 10^{-4}$	1.24	$1.23 \cdot 10^{-4}$
80	1.00	$4.22 \cdot 10^{-4}$	2.13	$4.00 \cdot 10^{-5}$	1.30	$5.01 \cdot 10^{-5}$
160	0.99	$2.11 \cdot 10^{-4}$	2.07	$9.5 \cdot 10^{-6}$	1.88	$1.36 \cdot 10^{-5}$
320	0.99	$1.06 \cdot 10^{-4}$	2.04	$2.30 \cdot 10^{-6}$	2.42	$2.53 \cdot 10^{-6}$

Table 3.1: Steady state with smooth topography. Error order measured at T=$50s$ (2J-scheme).

QUIESCENT STATE

cells	n=1		n=2		n=3	
	$L^1 error$	$L^\infty error$	$L^1 error$	$L^\infty error$	$L^1 error$	$L^\infty error$
20	$2.22 \cdot 10^{-16}$	$6.66 \cdot 10^{-16}$	$3.44 \cdot 10^{-16}$	$4.44 \cdot 10^{-16}$	$6.10 \cdot 10^{-16}$	$8.88 \cdot 10^{-16}$
40	$1.28 \cdot 10^{-16}$	$4.44 \cdot 10^{-16}$	$7.22 \cdot 10^{-17}$	$2.22 \cdot 10^{-16}$	$3.77 \cdot 10^{-16}$	$8.88 \cdot 10^{-16}$
80	$1.23 \cdot 10^{-15}$	$1.78 \cdot 10^{-15}$	$1.64 \cdot 10^{-16}$	$4.44 \cdot 10^{-16}$	$1.03 \cdot 10^{-15}$	$1.55 \cdot 10^{-15}$
160	$1.57 \cdot 10^{-16}$	$6.66 \cdot 10^{-16}$	$6.41 \cdot 10^{-16}$	$1.33 \cdot 10^{-15}$	$1.44 \cdot 10^{-15}$	$2.00 \cdot 10^{-15}$
320	$5.76 \cdot 10^{-16}$	$1.11 \cdot 10^{-15}$	$2.89 \cdot 10^{-16}$	$8.88 \cdot 10^{-16}$	$1.66 \cdot 10^{-15}$	$2.22 \cdot 10^{-15}$

NON-QUIESCENT STATE

cells	n=1		n=2		n=3	
	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$
20		$1.15 \cdot 10^{-5}$		$1.15 \cdot 10^{-5}$		$1.15 \cdot 10^{-5}$
40	1.93	$3.00 \cdot 10^{-6}$	1.93	$3.00 \cdot 10^{-6}$	1.93	$3.00 \cdot 10^{-6}$
80	1.98	$7.59 \cdot 10^{-7}$	1.98	$7.59 \cdot 10^{-7}$	1.98	$7.59 \cdot 10^{-7}$
160	2.00	$1.90 \cdot 10^{-7}$	2.00	$1.90 \cdot 10^{-7}$	2.00	$1.90 \cdot 10^{-7}$
320	2.00	$4.76 \cdot 10^{-8}$	2.00	$4.76 \cdot 10^{-8}$	2.00	$4.76 \cdot 10^{-8}$

Table 3.2: Steady state with smooth topography. Error order measured at T=$50s$ (1J-2J scheme).

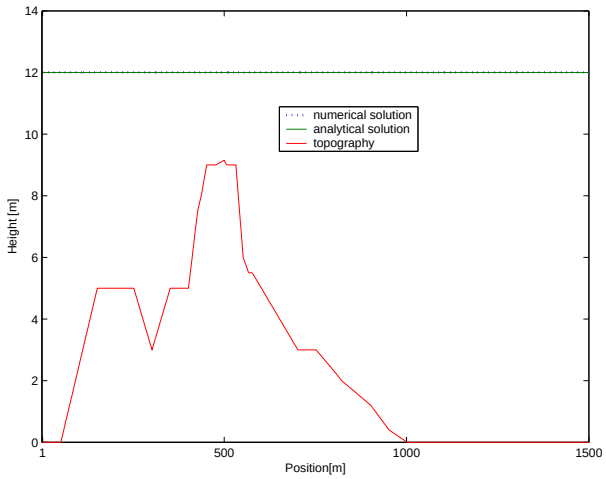


Figure 3.3: Water at rest over a complex geometry (T=200s, 300 nodes, 2nd order 2J-scheme)

are displayed in Figure 3.3. We can observe that the second order numerical scheme has not produced any visually detectable artificial movement of water. In table 3.3 we present the L^∞ errors in water depth and velocity after $T = 10.8s$ (600 nodes) with three different variants of the scheme.

Num. scheme	order	L^∞ error in h	L^∞ error in u
2J-scheme	r=1	$2.875 \cdot 10^{-1}$	2.0542
	r=2	$1.922 \cdot 10^{-1}$	$3.103 \cdot 10^{-1}$
	r=3	$4.870 \cdot 10^{-2}$	$4.030 \cdot 10^{-2}$
1J-scheme (Roe's average)	r=1	$3.553 \cdot 10^{-15}$	$3.780 \cdot 10^{-15}$
	r=2	$1.777 \cdot 10^{-15}$	$2.114 \cdot 10^{-15}$
	r=3	$3.553 \cdot 10^{-15}$	$2.534 \cdot 10^{-15}$
1J-2J scheme	r=1	$3.553 \cdot 10^{-15}$	$3.780 \cdot 10^{-15}$
	r=2	$1.777 \cdot 10^{-15}$	$2.556 \cdot 10^{-15}$
	r=3	$3.553 \cdot 10^{-15}$	$3.315 \cdot 10^{-15}$

Table 3.3: Quiescent state proposed in a workshop on dam-break wave simulation [76]. L^∞ error measured at T=$10.8s$.

Steady flow over a hump

This series of tests were proposed in the workshop [76] with the aim to evaluate the ability of numerical schemes to arrive and maintain the steady state over a non-flat

topography. In these simulations a bottom topography of 25m length is defined as:

$$z(x) = \begin{cases} 0.2 - 0.005(x-10)^2 & \text{if} \quad 8\,m < x < 12\,m \\ 0 & \text{otherwise} \end{cases} \tag{3.45}$$

In all cases, the initial conditions are $h + z = constant$ and $q = 0$. Depending on the initial and boundary conditions, the resulting flow could be at rest, subcritical or transcritical with or without shock. The analytical solution is computed with the Bernouilli equation:

$$\frac{q^2}{2gh^2} + h + z = H_a \qquad \text{where } H_a = \frac{q^2}{2gh_b^2} + h_b$$

where the discharge q is constant and is determined by the value imposed at the boundary, h_b is the water depth at the boundary which is also constant. For these experiments we use a grid with 100 points.

Subcritical flow The initial conditions are $h + z = 2m$, $q = 0$. For the boundary conditions:

$$\begin{aligned} \text{downstream:} \quad & h = \; 2\,m \\ \text{upstream:} \quad & q = \; 4.42\,m^2/s \end{aligned}$$

In figure 3.4 we display the results obtained with the second order 2J scheme. No significant differences can be seen when using $r = 1$ or $r = 3$ or when using the 1J-2J scheme. We note that the oscillations observed in the discharge are of the order of $O(\Delta x^r)$.

Transcritical flow without shock Initial conditions: $h+z = 0.66m$, $q = 0$. Boundary conditions:

$$\begin{aligned} \text{downstream:} \quad & h = \; 0.66\,m \text{ only when } F_r < 1 \\ \text{upstream:} \quad & q = \; 1.53\,m^2/s \end{aligned}$$

where $F_r = u/\sqrt{gh}$ is the *Froude number*.

The results obtained after 200s of are shown in Figure 3.5. In the second order simulation shown in Figure 3.5 we appreciate a slight 'dog-leg'-like effect at the maximum of the topography, which is no longer visible in the 1J-2J simulation. The glitch is improved when using the 2J third order scheme which fits correctly the analytical solution (as the 1J-2J scheme).

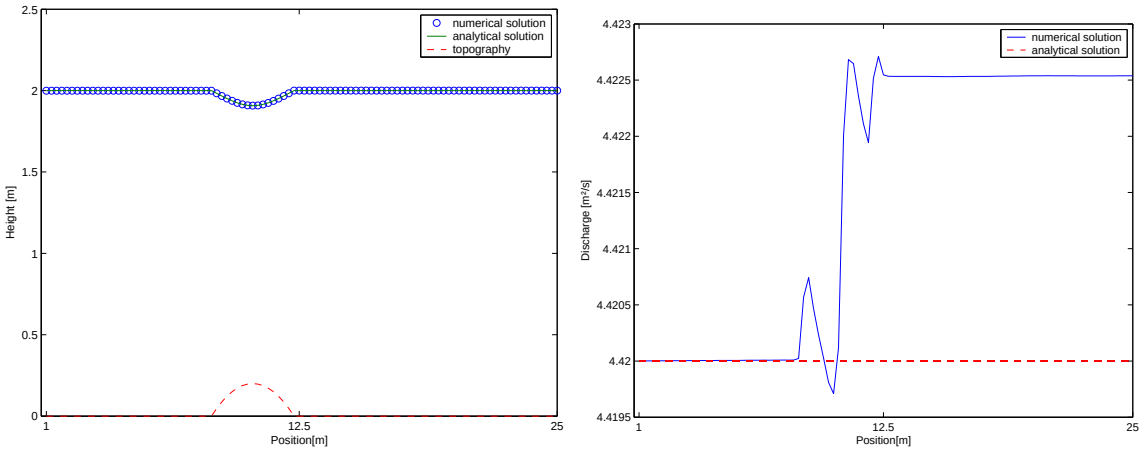


Figure 3.4: Subcritical flow over a hump (T=200s, 100 nodes, 2J-scheme $r = 2$). Water depth (left) and discharge (right)

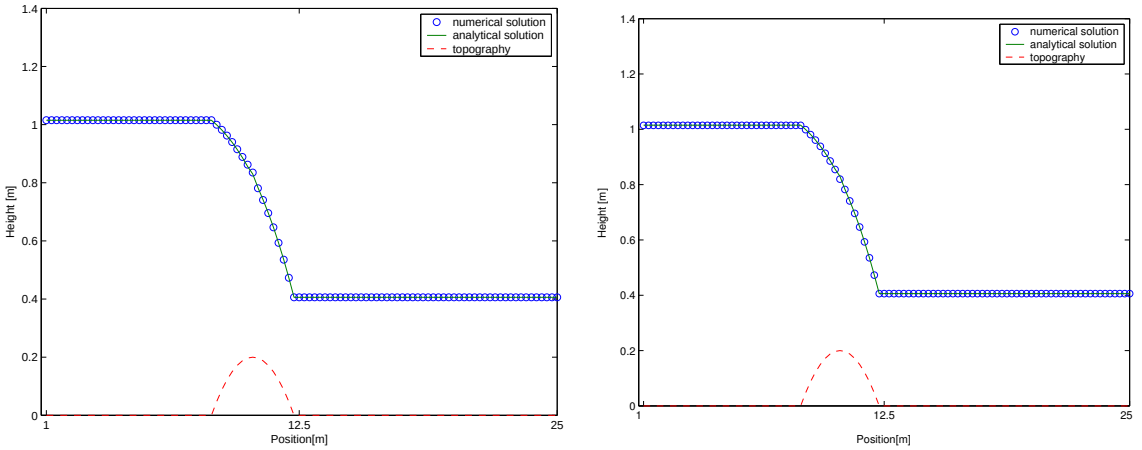


Figure 3.5: Transcritical flow without shock over a hump (T=200s, 100 nodes, 2nd order). Water depth: 2J-scheme (left) and 1J-2J scheme (right)

Transcritical flow with shock Initial conditions: $h + z = 0.33m$, $q = 0$. Boundary conditions:

$$\begin{aligned} \text{downstream:} \quad & h = \; 0.33\, m \\ \text{upstream:} \quad & q = \; 0.18\, m^2/s \end{aligned}$$

This test is particularly well suited to display the behavior of the schemes we propose. In Figure 3.6 we compare the results of the 1J-2J scheme with the 2J scheme for $r = 1, 2, 3$. We readily note that the 2J scheme gives a sharper resolution at the shock, however the glitches observed in the previous transcritical simulation distort in a noticeable manner the profile of the water surface in the first order simulation. The glitches improve with the order of accuracy, and the 1J-2J scheme produces the best resolution for each r.

3.4.2 Quasi stationary flow

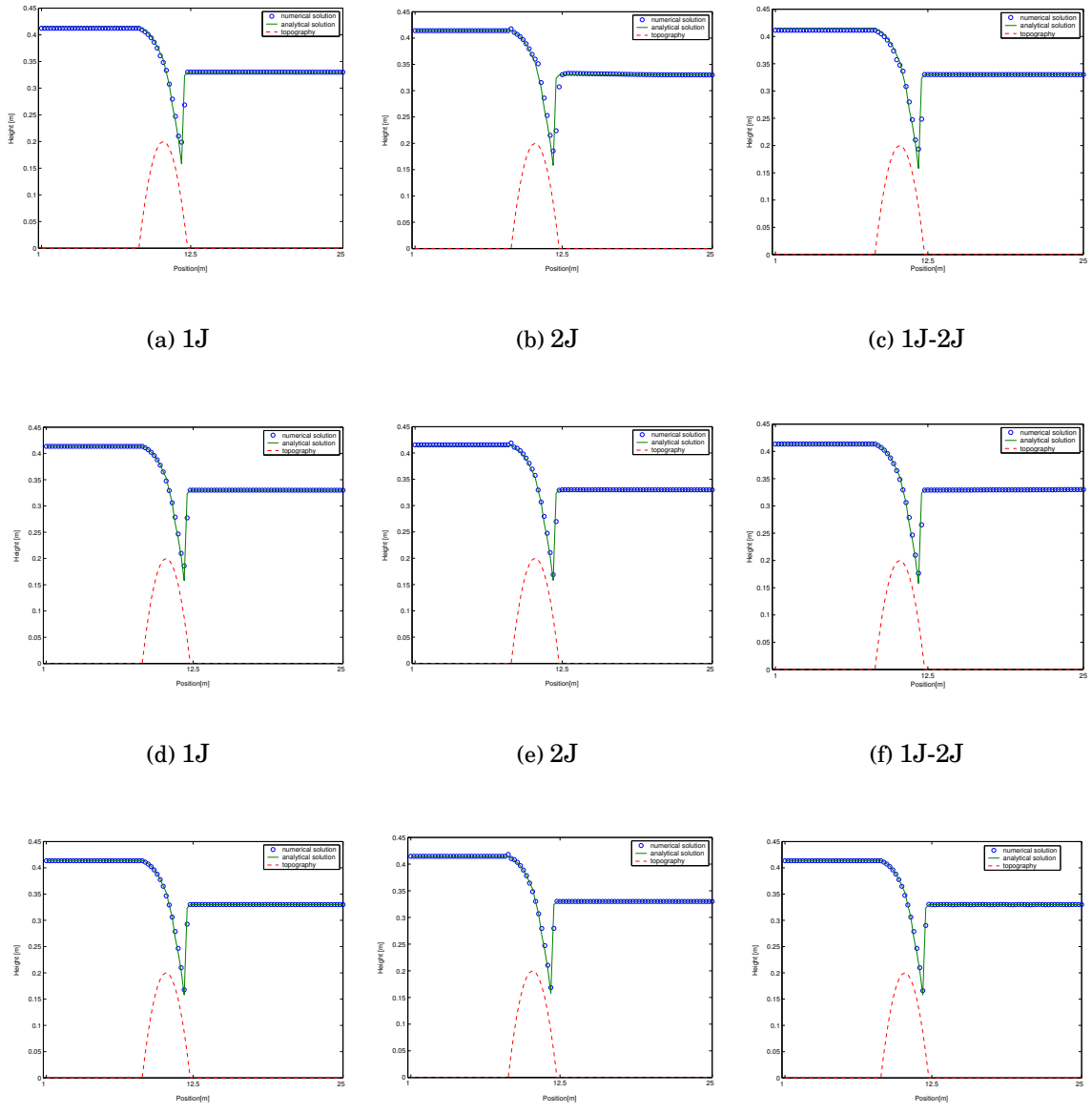
In [102], Le Veque proposes a test involving a quasi-stationary flow in order to evaluate the capability of the scheme to accurately compute small perturbations of the water surface over a variable topography. Le Veque uses this test to show the disadvantages of schemes that do not preserve steady states. The bottom topography is given as

$$z(x) = \begin{cases} 0.25(\cos(\pi(x - 0.5)/0.1) + 1) & \text{if} \quad |x - 0.5| < 0.1 \\ 0 & \text{otherwise} \end{cases}$$

on $0 < x < 1$ with $g = 1$. The initial conditions are $q = 0$ and

$$h(x) := 1 - z(x) + \epsilon \text{ for} \quad 0.1 < x < 0.2,$$

which represents a small hump perturbation of the quiescent steady state $(h, u) = (1 - z, 0)$. The initial disturbance splits into two waves propagating with left and right characteristics speeds $\pm c$. A magnified view of the water surface after $0.7s$ with $\epsilon = 10^{-3}$, with 300 nodes is shown in Figure 3.7. The negative effect of not satisfying the exact or approximate C- property in the first order 2J scheme is revealed in spurious oscillations (of the order of the perturbation itself) over the topography (Fig. 3.7(a)). Mesh refinement (not shown) improves the results and the numerical solution converges to the true solution. The numerical results for the second and third order 2J scheme, which satisfy the *approximate* C-property, display a much smaller level of numerically generated noise. On the other hand, the results of the 1J-2J scheme are accurate and without spurious oscillations (Figures 3.7(b)).

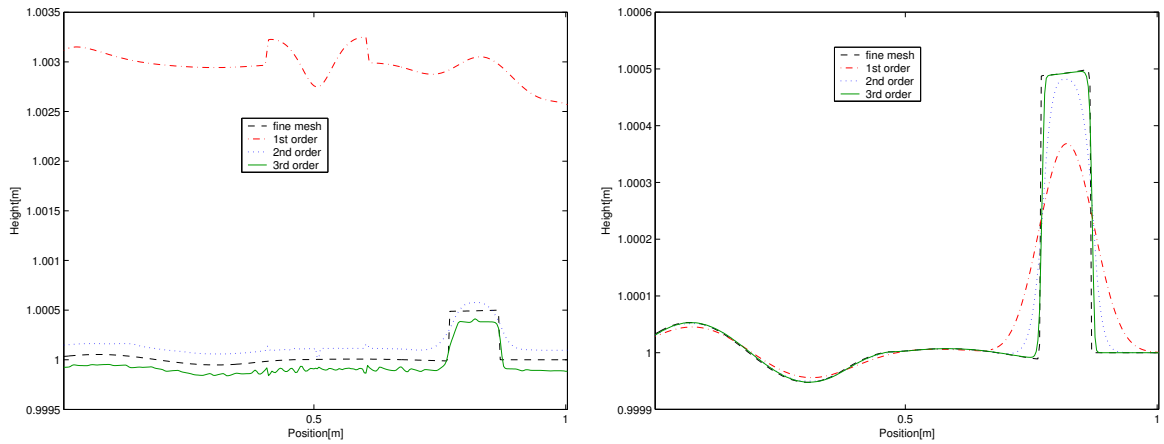


(a) 1J (b) 2J (c) 1J-2J

(d) 1J (e) 2J (f) 1J-2J

(g) 1J (h) 2J (i) 1J-2J

Figure 3.6: Transcritical flow with shock over a hump (T=200s, 100 nodes). Comparison of 1J and 2J scheme with the combined 1J-2J scheme. Top: 1st order. Middle: 2nd order. Bottom: 3rd order.



(a) 2J scheme (300 nodes)

(b) 1J-2J scheme (300 nodes)

Figure 3.7: Quasi stationary case, water surface (T=0.7s with $\epsilon = 10^{-3}$)

3.4.3 Wet/dry fronts and dry bed generation

Drain on a non-flat bottom

This simulation was proposed in [70]. The bottom topography is the same as in (3.45) and the initial condition is $h + z = 0.5$, $q = 0$. At the right boundary, an outlet condition on a dry bed is simulated. The left boundary is a mirror state. The flow reaches a steady state with $h + z = .2$ and $q = 0$ to the left of the bump and $h = 0$ to its right.

In Figure 3.8 we show the evolution of the water surface as in [70]. Here we used the second order version of the 1J-2J scheme with 300 nodes. As noticed in [70], the non-preservation of discrete steady states leads to the wrong water height in this experiment. In our case, it is important to use the modification of the source term contribution specified in section 3.3.5. If $\bar{\beta}_{i,i+1} = g/2(z_{i+1} - z_i)(h_{i+1} + h_i)$ is used at wet/dry fronts, then the C property does not hold. In this case, the water level at the steady state in the numerical simulation (not shown) is below the exact value.

Oscillating lake

This test was proposed in [12]. The aim here is to asses the behaviour of the scheme in situations of wet/dry fronts over non-flat topographies. The topography simulates a

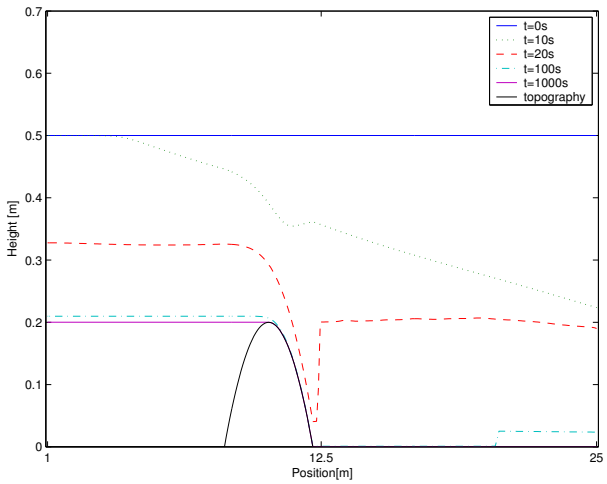


Figure 3.8: Drain on a non-flat bottom: water surface evolution (300 nodes, 2nd order 1J-2J scheme)

lake bed with non-flat bottom and non-vertical shores,

$$z(x) = 0.5(1 - 0.5(cos(\pi(x - 0.5)/0.5) + 1))$$

The water surface of a lake at rest is perturbed initially with a small sinusoidal wave,

$$h(0, x) = \max(0, 0.4 - z(x) + 0.04(sin((x - 0.5)/0.25) + 0.04\max(0, -0.4 + z(x))))$$

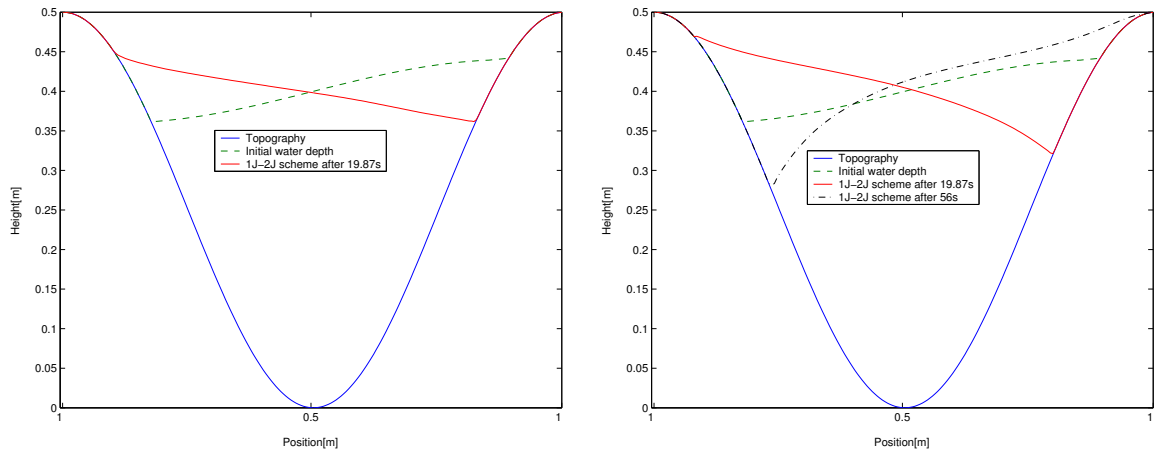
The flow oscillates in such a way that an interface between a wet cell and a dry cell has to be computed at each time step. The result after 19.87s with the 2nd order 1J-2J scheme can be seen in figure 3.9(a). The authors in [12] already pointed out the importance of using high order extensions of the scheme. In our case, the 1st order scheme does not damp the oscillations, instead we observe that the numerical solution leaves the domain in the course of time (see figure 3.9(b)). The 2nd and 3rd order versions of the scheme do maintain the periodic regime for all time.

Dry bed generation by rarefaction separation

This is an experiment over flat topography ($z = 0$) proposed by Toro in [134]. The initial conditions

$$U(x, 0) = \begin{cases} U_L = (h_L, q_L) = (0.1\ m, -0.3\ m^2/s) & \text{if} \quad x \leq 5\ m \\ U_R = (h_R, q_R) = (0.1\ m, 0.3\ m^2/s) & \text{if} \quad x > 5\ m \end{cases}$$

do not satisfy (2.29) at $x = 0$, hence a dry bed is formed instantaneously in the middle of a left going and right going rarefaction waves.



(a) 2nd order

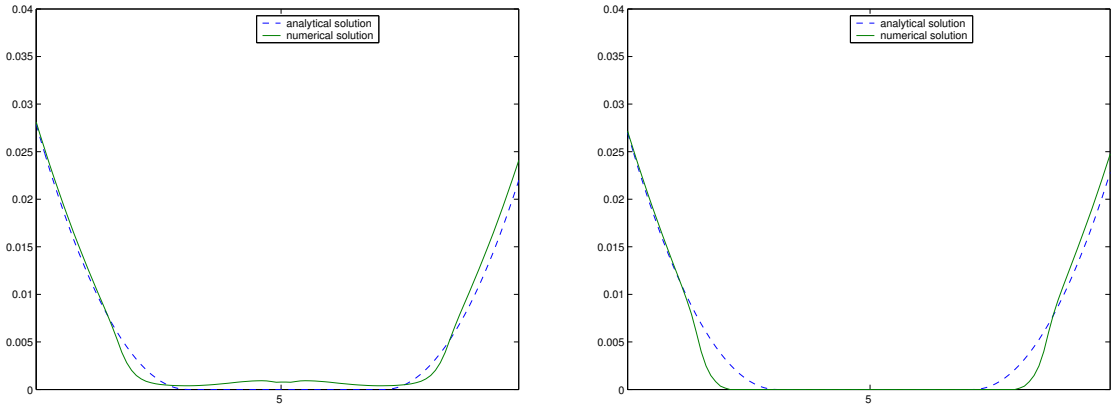
(b) 1st order

Figure 3.9: Oscillating lake (200 nodes, 2nd order 1J-2J scheme)

The generation of a dry state in between makes this test numerically difficult. In our context, it provides an example of a situation where the two-sided local characteristic decomposition does play a significant role. Our numerical experimentation shows that the 1J scheme cannot create a dry zone in the middle of the two rarefaction waves (with either of the two entropy fixes proposed, the regular LLF or H&H specified in (3.43))

As specified in section 3.3.5, two different modifications in the scheme are necessary in these situations: a) U^L and U^R should be computed avoiding any mixed information (considering an artificial dry state at the other side), b) compute the viscosity coefficient, $\alpha_{i+1/2}$, as in (3.43), i.e. using H&H entropy fix.

If we do not make any of these modifications in the 2J scheme (or the 1J-2J scheme), the dry zone is created but it is much larger than the analytical solution (simulation not shown). If we comply only with a) above and still use the viscosity coefficient in the regular LLF splitting (LLF-entropy fix), the dry bed is not created (see figure 3.10(a)). We can see in figure 3.10(b) that both modifications result in the generation of a dry zone (notice that it is still a little larger than the one in the analytical solution but not as large as the one obtained without any of the modifications). We also notice that when using the H&H entropy-fix (3.43) it is crucial to comply with a) above, otherwise instabilities such those in the next experiment (figure 3.11(a)) could be created.



(a) LLF entropy-fix only

(b) H&H entropy-fix and

Figure 3.10: Two rarefaction waves with dry bed generation (200 nodes, 2nd order, magnified view at center).

Dry bed generation with bottom topography

In [70], the basic test by Toro is modified by including a non-trivial topography, which is defined as

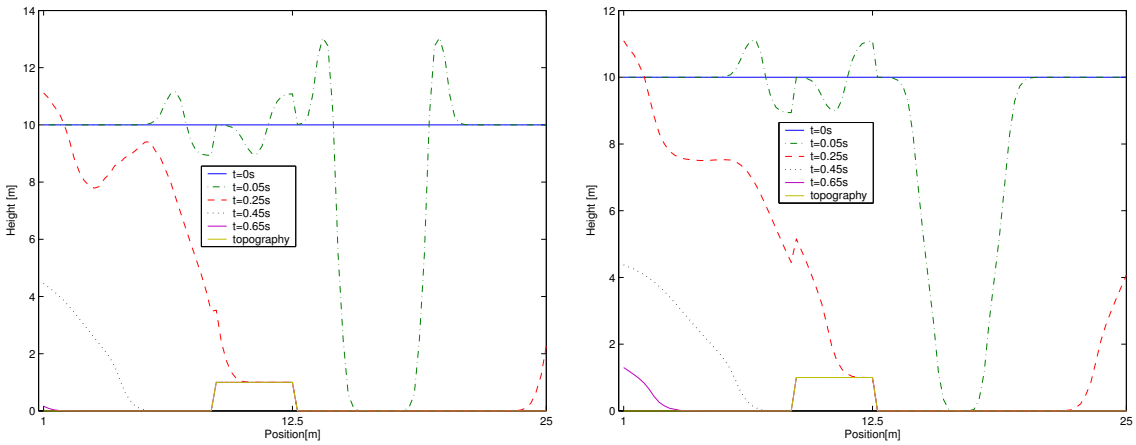
$$z(x) = \begin{cases} 1\,m & \text{if} \quad 25/3\,m < x < 12.5\,m \\ 0 & \text{otherwise} \end{cases}$$

with a length of 25 m. The initial conditions are $h + z = 10\,m$ and the discharge is -350 m^2/s if $x < 50/3$ m and 350 m^2/s otherwise.

As in the previous test, the initial data do not satisfy condition (2.29) so a dry bed, separating two rarefaction waves, is formed instantaneously over the flat portion to the right of the bump . The left-going wave interacts with the non-flat topography and as a consequence some waves are formed at the water surface.

The numerical results in figure 3.11 can be directly compared with those in [70]. Both results in figure 3.11 are computed switching to H&H entropy-fix when condition (2.29) is not satisfied. In Figure 3.11(a) we do not comply with condition a) i.e. allow mixing when computing U^L and U^R. In this case, instabilities are created when the left-going wave interacts with the bump in the topography. On the other hand, when complying with a), the result is more accurate (see figure 3.11(b)). In both simulations we have used the 1J-2J scheme (in this test, no noticeable differences with the 2J scheme

can be observed). The same type of instabilities also occur when using the original scheme (without any of the modifications in section 3.3.5). If we only avoid mixing information in computing U^L and U^R at cell walls not satisfying (2.29), but always use the LLF entropy-fix the instabilities disappear but the dry bed still has not been created at $t = 0.05$s.



(a) Modified Harten-Hyman entropy-fix (with original ENO interpolation of states at interfaces)

(b) Modified Harten-Hyman entropy-fix (with modif. ENO interpolation of states at interfaces)

Figure 3.11: Dry bed generation by a double rarefaction wave over a step (100 nodes, 2nd order 2J scheme)

3.4.4 Two-dimensional experiments

The numerical scheme can be easily extended to the two-dimensional case. It suffices to apply the one-dimensional ENO discretization in a standard dimension by dimension fashion as explained in sections 3.2.2 and 3.3 (see also [126]) and then the Runge-Kutta method is applied over the discretized spatial term (which includes the difference, in each direction, of modified fluxes).

In this section we show two experiments of a fluid running over a channel with the same topography but with different initial and boundary conditions in each one of them. The topography can be seen in figure 3.12. It represents a channel of 75 m length and 30 m width with three mounds and was proposed by Kawahara and Umetsu [97]. In order to discretize the topography we use 60x150 quadrilateral cells. Both experiments are done considering that the upper and lower boundaries are solid walls (the normal

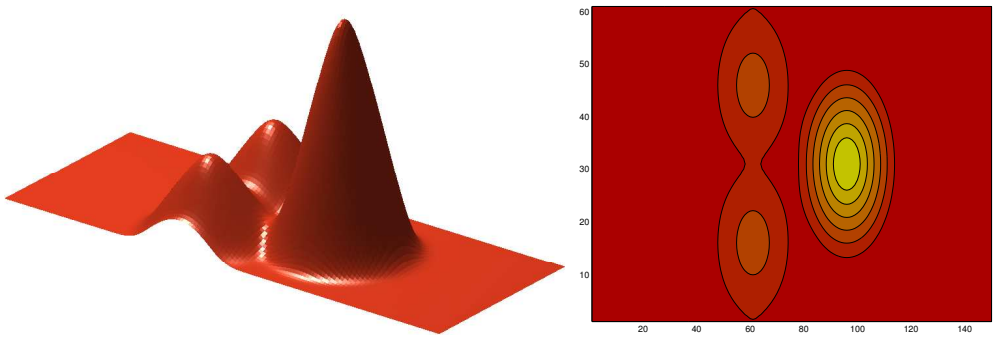


Figure 3.12: Topography with three mounds proposed by [97]. Left: 3D representation. Right: level lines (axis numbers indicate cells, which are quadrilateral cells)

velocity to the boundary is set to zero). They were proposed in [97] and in [31], both of them considered friction terms and the first one a turbulent viscosity term also. Since our aim is to assess the correct discretization of convective terms and source terms due to the topography we are not going to consider those additional source terms here.

Flooding over an open channel

This experiment was proposed by Kawahara and Umetsu [97]. The upper and lower boundaries simulate solid walls. The right boundary represents an open wall (we use first order extrapolation). On the other hand, the left boundary represents an inflow of the following characteristics:

- From $t = 0$ s to $t = 300$ s: $h = 0.5$ m and $u = 1.0$ m/s (horizontal velocity).

- From $t = 300$ s to $t = 900$ s: $h = 1.0$ m and $u = 1.0$ m/s (horizontal velocity).

It has to be noticed that as a difference with [97] we do not consider turbulent viscosity nor friction terms in the simulation. We present in figure 3.13 some results of the water elevation and vector field of velocities at different time instants. We can see in these results that the essential characteristics of the convective flux are well modeled and that the vector field of velocities is consistent with the topography. In particular, notice the difference between the last two frames in figure 3.13. The two smaller mounds in the frame corresponding to 300 s are not covered by the water because the inflow at the left boundary is 0.5 m. However, as the inflow is increased to 1.0 m from 300 s to 900 s, we can see in the last frame that the smaller mounds are then completely covered by the water.

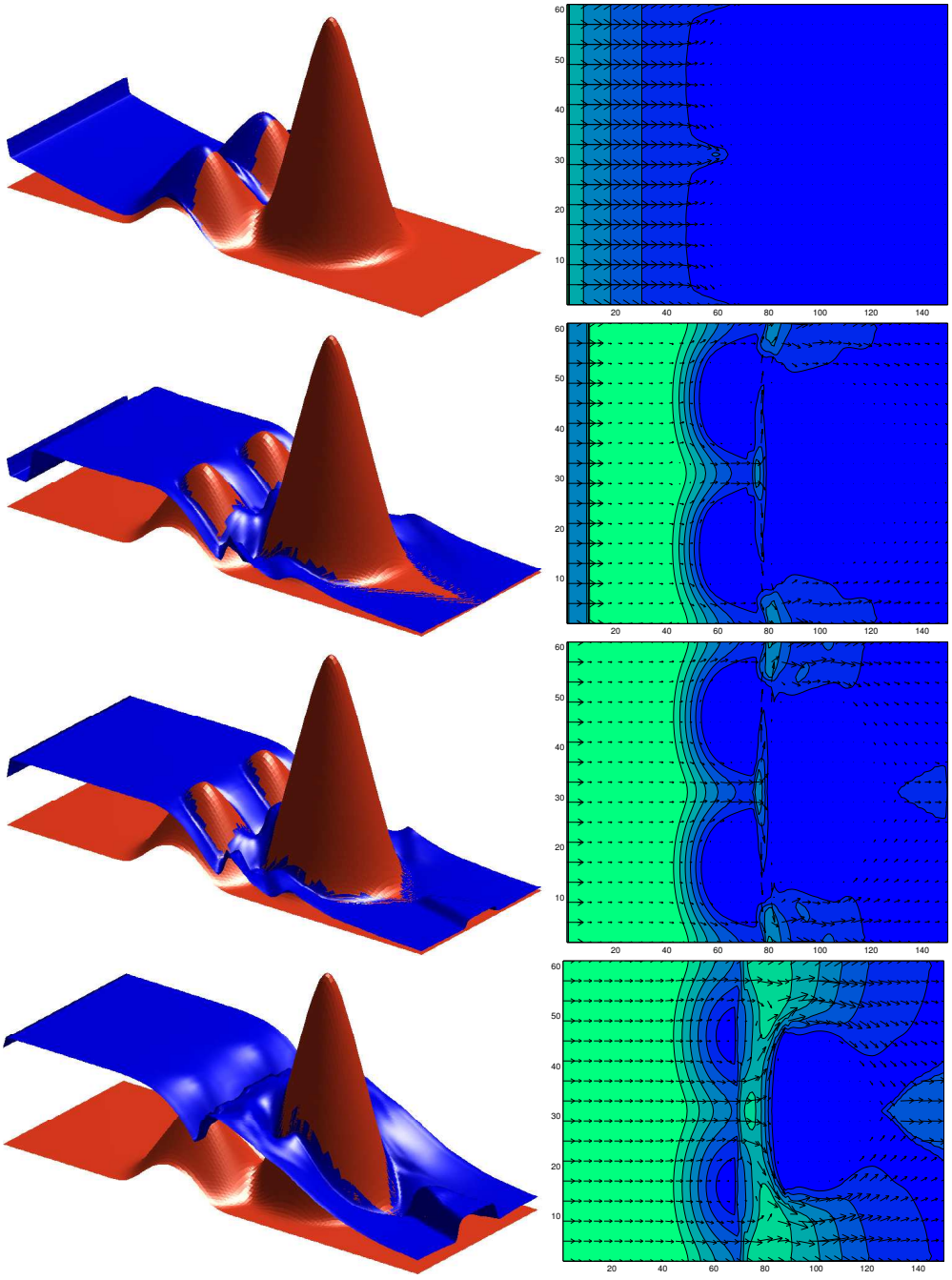


Figure 3.13: Open channel over three mounds, results after 7.5, 30, 300 and 900 s from top to bottom. Left: 3D representation. Right: level lines and vector field of velocities.

Dam-break over a closed channel

Brufau et al. [31] take the same topography as in [97] but they simulate a dam-break and all the boundaries are considered solid walls so water never leaves the rectangular domain. The dam is situated at $x = 16$ m and contains 900 m^3 of water. They also consider friction losses but we do not consider this term. Results after different time instants can be seen in figure 3.14. It seems that the simulated flux is physically consistent and that the vector fields of velocities are correct. Notice also how the result at 450 s is near the steady state.

3.5 Conclusion

We have proposed an extension of Marquina's flux formula [59, 66] to the shallow water system with source terms coming from a non flat topography. The source term is included in a direct discretization of the system following a technique introduced in [71] by Gascón and Corberán.

We call our extension the 2J scheme, since it complies with the basic design principle in Marquina's flux formula: Two Jacobian evaluations are used at each interface in order to determine the characteristic information and the way in which this information is used in the numerical scheme. Since our 2J combined numerical flux does not verify the exact C-property, while a closely related 1J combined numerical flux does, we propose to use a 1J-2J combined numerical flux which *essentially* satisfies the C-property.

The results in this paper show a variety of situations where the *combined* 1J-2J scheme performs well. These situations involve, in addition to standard tests on shallow water flow over a hump in steady and quasi-steady situations, the generation of dry beds and flows over adverse slopes. The results indicate that the 1J-2J numerical flux provides a sufficiently simple numerical scheme that can be easily upgraded to obtain second and third order accuracy together with high resolution.

We carried out a couple of 2D experiments over a non-trivial topography. More realistic simulations in 2D is the subject of ongoing work. The addition of source terms coming from friction losses poses no special difficulties, since it is agreed that these terms do not need any special upwinding procedure.

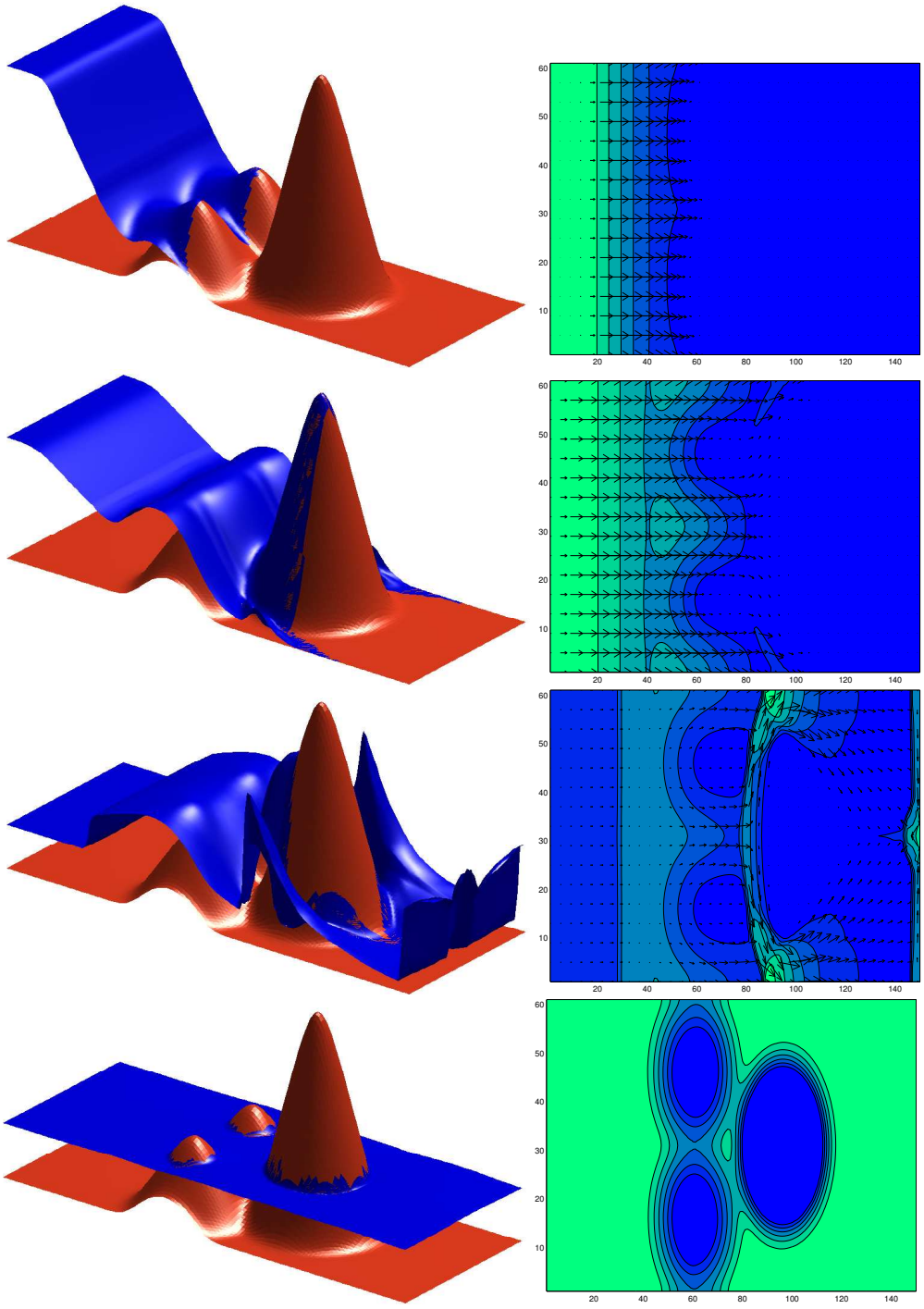


Figure 3.14: Dam-break over three mounds with solid walls, results after 2, 5, 12 and 450 s from top to bottom. Left: 3D representation. Right: level lines and vector field of velocities.

Appendix A

Some remarks on the conservativity

In section 3.3.3 we have seen that in the scalar case we have

$$G_{i+1/2} - G_{i-1/2} = G^{+}_{i+1/2} - G^{-}_{i-1/2} \tag{A.1}$$

so the scheme continues being conservative if we use $G^{\pm}_{i+1/2}$ instead of $G_{i+1/2}$. To make the extension to nonlinear systems we simply apply, in each characteristic field, the same rules that we have found in the scalar case in order to compute the fluxes $G^{+}_{i+1/2}$ and $G^{-}_{i+1/2}$ that collect the source term contribution when the wind comes from the right and from the left respectively. We recall that in the system case the fluxes $G^{\pm}_{i+1/2}$ are computed from the contribution of the fluxes $(\tilde{G}^{\pm})^{p,R}$ and $(\tilde{G}^{\pm})^{p,L}$ in each characteristic field (see (3.28)). Thus, the construction of the algorithm makes possible that in the same interface $i+1/2$ the source term $B_{i,i+1}$ is projected using the left-biased eigenvectors in one characteristic field and using the right-biased eigenvectors in other field. As a consequence, (A.1) is not always verified and the error $\epsilon = (G_{i+1/2} - G_{i-1/2}) - (G^{+}_{i+1/2} - G^{-}_{i-1/2})$ that we make is

$$\epsilon = \sum_{p} \left(L^{p,S(p,x_{i+1/2})}_{i+1/2} \otimes R^{p,S(p,x_{i+1/2})}_{i+1/2} - L^{p,S(p,x_{i-1/2})}_{i-1/2} \otimes R^{p,S(p,x_{i-1/2})}_{i-1/2} \right) B_i \tag{A.2}$$

where $\otimes$ is the tensor product between vectors and $S(p, x_{i+1/2})$ is a discrete function with two possible values $\{L, R\}$. The function $S(p, x_{i+1/2})$ indicates which sided decomposition we use at interface $x_{i+1/2}$ and at p-th characteristic field. Indeed, if the conserved variables are sufficiently smooth, after (A.2) the error is $O(\Delta x^r)$ which is of the same order that the error order of the scheme. Thus, in smooth solutions, the extension to systems is 'conservative' (except for an error $O(\Delta x^r)$). On the other hand, if we use only one Jacobian, the error (A.2) is zero, so the scheme is always conservative. Moreover, the error (A.2) is only due to the first order terms of the source term contribution.

Let us describe an extension of the first order terms to systems which is conservative. The technique that we follow is basically the same, except that we write $B_i = \sum_{j=0}^{i-1} B_{j,j+1}$ and we divide $B_{i,i+1}$ into two pieces each one projected with its corresponding local-sided eigenvectors, ie,

$$(\tilde{G}^p_{i+1/2})^L R^p(U^L) = \begin{cases} L^p(U^L)F_i R^p(U^L) + \{i+\frac{1}{2}\}_p & \text{if } \lambda^p(U^L), \lambda^p(U^R) > 0 \\ 0 & \text{if } \lambda^p(U^L), \lambda^p(U^R) < 0 \\ \frac{1}{2}L^p(U^L)(F_i + \alpha_{i+1/2}U_i)R^p(U^L) + \frac{1}{2}\{i+\frac{1}{2}\}_p & \text{else} \end{cases}$$

$$(\tilde{G}^p_{i+1/2})^R R^p(U^R) = \begin{cases} 0 & \text{if } \lambda^p(U^L), \lambda^p(U^R) > 0 \\ L^p(U^R)F_{i+1} R^p(U^R) + \{i+\frac{1}{2}\}_p + \{i+\frac{1}{2}, i+1\}_p & \text{if } \lambda^p(U^L), \lambda^p(U^R) < 0 \\ \frac{1}{2}L^p(U^R)(F_{i+1} - \alpha_{i+1/2}U_{i+1})R^p(U^R) + \frac{1}{2}\{i+\frac{1}{2}\}_p + \frac{1}{2}\{i+\frac{1}{2}, i+1\}_p & \text{else} \end{cases}$$

where
$$\begin{aligned} \{i+\tfrac{1}{2}\}_p &:= \textstyle\sum_{j=0}^{i} \left(L^p(U^R_{j-1/2})B_{j-1/2,j}R^p(U^R_{j-1/2}) + L^p(U^L_{j+1/2})B_{j,j+1/2}R^p(U^L_{j+1/2})\right) \\ \{i+\tfrac{1}{2}, i+1\}_p &:= L^p(U^R)B_{i+1/2,i+1}R^p(U^R) \end{aligned}$$

In practice, we take $B_{i,i+1/2} = B_{i+1/2,i+1} = \frac{1}{2}B_{i,i+1}$ in the experiments. We have then,

$$G_{i+1/2} = \sum_p (\tilde{G}^p_{i+1/2})^L R^p(U^L) + (\tilde{G}^p_{i+1/2})^R R^p(U^R) = \sum_p G^p_{i+1/2}$$

We define the fluxes that collect the source term contribution in the p-th characteristic field following the wind direction as,

$$G^{p,+}_{i+1/2} = G^p_{i+1/2} - \{i\}_p$$

$$G^{p,-}_{i+1/2} = G^p_{i+1/2} - \{i+1\}_p$$

where $\{i+1\}_p := \{i+\frac{1}{2}\}_p + \{i+\frac{1}{2}, i+1\}_p$. Thus, $G^{p,-}_{i-1/2} = G^p_{i-1/2} - \{i\}_p$ and we have the equality,

$$G^+_{i+1/2} - G^-_{i-1/2} = G_{i+1/2} - G_{i-1/2} \qquad \text{where } G^\pm_{i+1/2} = \sum_p G^{p,\pm}_{i+1/2}$$

We can see that this extension to systems is conservative even if we use two Jacobians. The final expression of fluxes $G^{p,+}_{i+1/2}$ and $G^{p,-}_{i+1/2}$ is

$$G^{p,+}_{i+1/2} = \begin{cases} L^p(U^L)F_i R^p(U^L) & \text{if } \lambda^p(U^L), \lambda^p(U^R) > 0 \\ L^p(U^R)F_{i+1}R^p(U^R) + L^p(U^L)B_{i,i+1/2}R^p(U^L) + L^p(U^R)B_{i+1/2,i+1}R^p(U^R) & \text{if } \lambda^p(U^L), \lambda^p(U^R) < 0 \\ \frac{1}{2}L^p(U^L)(F_i + \alpha_{i+1/2}U_i)R^p(U^L) + \frac{1}{2}L^p(U^R)(F_{i+1} - \alpha_{i+1/2}U_{i+1})R^p(U^R) & \\ \quad + \frac{1}{2}L^p(U^L)B_{i,i+1/2}R^p(U^L) + \frac{1}{2}L^p(U^R)B_{i+1/2,i+1}R^p(U^R) & \text{else} \end{cases}$$

$$G^{p,-}_{i+1/2} = \begin{cases} L^p(U^L)F_i R^p(U^L) - L^p(U^L)B_{i,i+1/2}R^p(U^L) - L^p(U^R)B_{i+1/2,i+1}R^p(U^R) & \text{if } \lambda^p(U^L), \lambda^p(U^R) > 0 \\ L^p(U^R)F_{i+1}R^p(U^R) & \text{if } \lambda^p(U^L), \lambda^p(U^R) < 0 \\ \frac{1}{2}L^p(U^L)(F_i + \alpha_{i+1/2}U_i)R^p(U^L) + \frac{1}{2}L^p(U^R)(F_{i+1} - \alpha_{i+1/2}U_{i+1})R^p(U^R) & \\ \quad - \frac{1}{2}L^p(U^L)B_{i,i+1/2}R^p(U^L) - \frac{1}{2}L^p(U^R)B_{i+1/2,i+1}R^p(U^R) & \text{else} \end{cases}$$

We note that this new scheme has the same construction that the one in section 3.3.4 with the difference that now we use

$$L^p(U^L)B_{i,i+1/2}R^p(U^L) + L^p(U^R)B_{i+1/2,i+1}R^p(U^R)$$

instead of $L^{p,S(p,x_{i+1/2})}_{i+1/2}B_{i,i+1}R^{p,S(p,x_{i+1/2})}_{i+1/2}$. As we have already said, these expressions ($G^{p,+}_{i+1/2}$ and $G^{p,-}_{i+1/2}$) correspond to the first order terms of the flux $G_{i+1/2}$ (thus making an abuse of notation). The HOT can be obtained as before or in the case of the source term, directly computing the high order extension of the new expression of the source term discretization.

Observe that interpreting $U^L = U^R$ the above notation encompasses the case of one Jacobian at each interface.

With this discretization of the source term and using (3.27) we have:

$$\sum_{j=0}^{N}(U_j)_t + \sum_{j=0}^{N}\frac{G^+_{j+1/2} - G^-_{j-1/2}}{\Delta x} = 0$$

Since

$$\sum_{j=0}^{N}\frac{G^+_{j+1/2} - G^-_{j-1/2}}{\Delta x} = \sum_{j=0}^{N-1}\frac{G^+_{j+1/2} - G^-_{j+1/2}}{\Delta x}$$

(where we assume that the boundary conditions are such that $G^-_{-1/2} = G^+_{N+1/2} = 0$) and

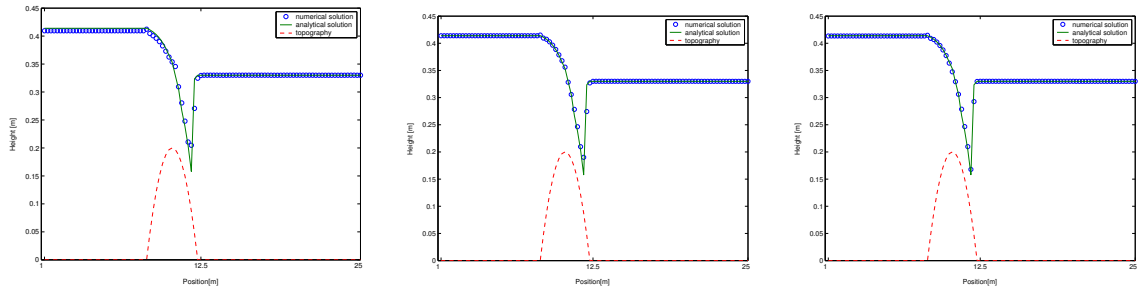
$$G^+_{j+1/2} - G^-_{j+1/2} = B_{j,j+1}$$

we obtain that

$$\Delta x\left(\sum_{j=0}^{N}U_j\right)_t + \sum_{j=0}^{N-1}B_{j,j+1} = 0$$

(we would add $G^+_{N+1/2} - G^-_{-1/2}$ in the right hand side of the above equation to include general boundary conditions).

We show in figure A.1 the results obtained with this new discretization of the source term in the steady case corresponding to a transcritical flow with shock over a hump. We can compare the results with those in figure 3.6 which correspond to the scheme explained in section 3.3.4. We can observe that these results are quite similar (do not improve nor worsen them) to those of figure 3.6 so we choose the discretization based on integrals over cells which is more easily computable.



(a) 1st order (b) 2nd order (c) 3rd order

Figure A.1: Transcritical flow with shock over a hump (T=200s, 100 nodes). New first order source term contribution based on integrals over semi-cells.

Part II

Day for night

Chapter 4

Digital Day for Night

ABSTRACT

In film production, it is sometimes not convenient or directly impossible to shoot some night scenes at night. The film budget, schedule or location may not allow it. In these cases, the scenes are shot at daytime, and the 'night look' is achieved by placing a blue filter in front of the lens and under-exposing the film. This technique, that the American film industry has used for many decades, is called 'Day for Night' (or 'American Night' in Europe). But the images thus obtained don't usually look realistic: they tend to be too bluish, and the objects' brightness seems unnatural for night-light. In this chapter we introduce a digital Day for Night algorithm that achieves very realistic results. We use a set of simple equations, based on real physical data and visual perception experimental data. To simulate the loss of visual acuity we introduce a novel diffusion Partial Differential Equation (PDE) which takes luminance into account and respects contrast, produces no ringing, is stable, very easy to implement and fast.

4.1 Introduction

In film production, it is sometimes not convenient or directly impossible to shoot some night scenes at night. The film budget, schedule or location may not allow it. In his book 'Making movies' [104], the great American director Sidney Lumet points out the difficulties of night shooting on location. Apart from the inherent lack of control that locations have (as opposed to studios), night shooting requires that *everything* be lit artificially. This may be a big problem if the location covers a wide area. In some locations the terrain may make it difficult to bring the truckloads of lamps and generators that the shooting requires. Because they make a lot of noise, generators are placed

far from the set so as not to interfere with the sound department. Very long cables have to be laid from the lights to the generators, and the rigging crew and electricians have much more extra work. If the night shooting continues for weeks, cast and crew get exhausted. And on the days when night shooting takes place, there is time for very little or no day shooting at all, which may complicate the schedule. If the geography of the location is very inconvenient or dangerous for shooting at night, or if the budget can't afford extra pay for night shooting, or if a schedule delay is out of the question, what is usually done is Day for Night: the night scene is shot at day, but with a technique that gives a 'night look' to the film.

This technique is exclusively optical, typically a blue filter is placed behind the lens, and the film is under-exposed. This technique, that the American film industry has used for many decades, is called 'Day for Night' (or 'American Night' in Europe). While this usually works, the results nonetheless lack realism. Blue turns out to be the predominant color, the other colors virtually indistinguishable. The objects in the scene have a very unnatural brightness. They are dimmer than with daylight, but somehow we still can see everything that is in the scene. Moreover, no detail is lost: if there is a sign with small fonts, we can read it as if it were day.

The problem is that with mere optical means (blue filters, under-exposure) we can not expect to reproduce all the modifications that account for how we see the world at night. Firstly, let us assume that no artificial light source is present in our scene, i.e., that all the light is supposed to be coming from the moon, the sky and the stars. While it is true that at night we perceive blue objects brighter, this is *not* due to the light at night being bluer. Actually, light at night has a power spectrum with a stronger red component than daylight, and with less blue [91]. It is the human visual system that works very differently under low light conditions. The colors are perceived as less vivid, brightness is modified according to wavelength, the contrast changes significantly, and visual acuity decreases. It is therefore usual for Day for Night footage to be heavily retouched in post-production, mostly manually, by a color artist.

In this part we propose an algorithm to automatically transform a 'day image' into a 'night' version of it. The only other input necessary is the desired level of darkness of the final result. Our algorithm uses a very simple set of equations to model the different factors involved in night vision. These equations are based on physical data and visual perception experimental data. We start by replacing daylight illumination with night-like illumination, a procedure which to the best of our knowledge is novel when addressing the Day for Night problem. For the simulation of the loss of visual

acuity, we introduce a novel diffusion PDE that models the spatial summation principle [51], takes luminance into account and respects contrast, produces no ringing, is stable, very easy to implement and fast.

4.2 The Day for Night algorithm

Our algorithm takes as input a color RGB image (coming from digital video or obtained from a scanned film) and a desired level of luminance for the final result. We transform the image in five steps. While the user would typically only care about the final image, we present here all the steps separately. This may help the user to modify some steps to obtain a more expressive result. The aim is to achieve realistic and visually pleasing results, even if they are not entirely compliant with the models of human visual perception.

We perform the following operations for each pixel in the image, one at a time. In the first step, we estimate the reflectance values for the object in the scene at that particular pixel. For this we assume that the day scene has been lit with daylight and we use the standard illuminant D_{65} to approximate it. Then we replace the D_{65} with an estimation of the spectrum of the night sky. This is, to the best of our knowledge, a novel procedure in the context of this problem. In the second step we modify the chromaticity of the estimated reflectance values, assuming that the eye is dark adapted. In the third step we modify the luminance values, since the luminous efficiency function depends on the illumination level of the scene. In the fourth step we modify the contrast, since threshold values for 'just noticeable differences' depend on illumination levels. These steps 2 to 4 are implementations of algorithms already introduced in the literature on tone reproduction and modeling of visual perception, like those of [135, 142, 69]. Finally, in the fifth step we perform diffusion on the image to account for the loss of visual acuity at nighttime illumination levels. For this last step we introduce a novel equation that models the spatial summation principle [51], but without producing ringing or other visual artifacts.

Let us comment on each of these steps now. For a very thorough coverage of Color Science, and an in-depth exposition of the concepts mentioned in this chapter, we refer the interested reader to the excellent treatise by Wyszecki and Stiles [145].

One note concerning the evaluation of the results. We will be producing images of rather low luminance, so they will look very differently on a computer screen in a well lit room than on a dimly lit room. Another crucial factor for the evaluation is the

'brightness' setting of the monitor. We recommend the reader to set the brightness level to 50%, the monitor being in a well lit room with no direct light on the screen. This would be an approximation only, for correct evaluation calibrated hardware is required.

4.2.1 Estimation of reflectance values

Let us say we have an object with reflectance $\beta(\lambda)$, where λ is the wavelength, and it is lit with an illuminant with spectral power $S(\lambda)$. In the XYZ color model of the CIE [145], the tristimulus values X, Y and Z of an object-color stimulus are obtained with these equations:

$$\begin{aligned} X &= k \int \beta(\lambda) S(\lambda) \overline{x}(\lambda) d\lambda \\ Y &= k \int \beta(\lambda) S(\lambda) \overline{y}(\lambda) d\lambda \\ Z &= k \int \beta(\lambda) S(\lambda) \overline{z}(\lambda) d\lambda \end{aligned} \tag{4.1}$$

where the factor k is defined so that the Y tristimulus for the *perfect reflection diffuser* ($\beta(\lambda) = 1.0$ at all wavelengths) is equal to 100. Functions $\overline{x}(\lambda)$, $\overline{y}(\lambda)$ and $\overline{z}(\lambda)$ denote the color-matching functions for a standard observer. These are experimental values tabulated by the CIE. The values X, Y and Z at each given pixel are known, we get them by converting the original R, G and B values (i.e. we go from the RGB to the XYZ color model):

$$r = \frac{R}{R+G+B} \quad g = \frac{G}{R+G+B} \quad b = \frac{B}{R+G+B}$$

$$L = L_R R + L_G G + L_B B \; ; \quad L_R = 1, L_G = 4.59, L_B = 0.06$$

$$x = \frac{0.49r+0.31g+0.20b}{0.67r+1.13g+1.20b}$$

$$y = \frac{0.18r+0.81g+0.01b}{0.67r+1.13g+1.20b}$$

$$z = \frac{0.00r+0.01g+0.99b}{0.67r+1.13g+1.20b}$$

$$X = \frac{x}{y} L \quad Y = L \quad Z = \frac{z}{y} L$$

Care must be taken in that the RGB image has been acquired non-linearly from real-world luminances [135]. Each type of photographic film has a characteristic curve that determines how luminances in the real world (L_{rw}) are encoded as transparencies (T) in the film: $T = a_1 {L_{rw}}^{\gamma}$, where the constant factor a_1 depends on the film speed and the choice of exposure time, lens and camera aperture. This function $T(L_{rw})$ is valid within a certain interval of luminances, outside which the value T is 'clipped'. See figure 4.1, as in [135]. Since we need to work with real-world luminances, we must undo the transformation: $T' = a_2 T^{\frac{1}{\gamma}}$. For this we need the contrast sensitivity specification of the film stock that was used to shoot the original image. If we don't have this data, we

can nonetheless obtain good results by assuming a value for γ and selecting a value for the 'night γ' that gives a visually convincing result, as is done in section 4.2.4.

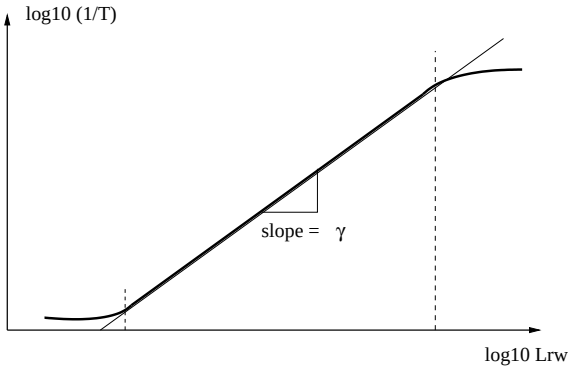


Figure 4.1: Characteristic curve for photographic film (after [135])

Returning to equation (4.1), the values for $S(\lambda)$ are not known precisely, unless they were measured when the original image was taken. But it is safe to assume for $S(\lambda)$ the properties of the CIE standard illuminant D_{65}, which corresponds to a phase of natural daylight. The only unknowns are then the reflectance values. Since we want to substitute the daylight illuminant $S(\lambda)$ with a nighttime illuminant $S'(\lambda)$, we must first compute $\beta(\lambda)$. If we take just three wavelengths λ_1, λ_2 and λ_3, we can obtain a very rough estimate of $\beta(\lambda_1)$, $\beta(\lambda_2)$ and $\beta(\lambda_3)$ by solving the 3x3 system of equations we get from (4.1) if we replace the integrals with discrete summations of three terms. This is a very crude and simple approximation, that nonetheless produces good enough results: in figure 4.6 we have tested three different CIE standard illuminants without producing great perceptual changes in the simulated night scene. For this step we could use much better reflectance models like those in [108] or [128].

Once we know the reflectance values, we substitute $S(\lambda_i)$ with $S'(\lambda_i)$, $i = 1, 2, 3$. We have taken λ_i near the values for monochromatic red, green and blue. In our experiments we have used for $S'(\lambda)$ the experimental values obtained in [111]. Surprisingly, if we only change the illuminant, the image colors get warmer, as we see in figure 4.5(b). This is due to the fact that the night light $S'(\lambda)$ has more power in the long wavelengths and less in the short ones (i.e. more 'red' and less 'blue'). Therefore, the perceived shift towards blue of colors under natural night light conditions is due only to the human night vision system.

4.2.2 Modification of chromaticity

The perceived chromaticity depends greatly on the illumination level. As we decrease the illumination level, the colors become less saturated. A color that is very vivid under daylight seems less and less vivid as the illumination decreases. This property is very difficult to emulate directly on film, with techniques ranging from the pre-exposing of film [104] to the 'bleach by-pass' at the developing stage [64]. The experimental data in [89], [145] and [131] show how monochromatic lights of different wavelengths are seen with evolving chromaticities as the surrounding luminances change. We use these data to modify accordingly the color matching functions $\overline{x}(\lambda)$, $\overline{y}(\lambda)$ and $\overline{z}(\lambda)$. This is nothing novel, see for instance the excellent works [69], [60], [142].

Figure 4.5(c) shows the result of applying this modification to the original image (figure 4.5(a)).

4.2.3 Modification of luminance

There are two kind of photoreceptors in the retina, the rods and the cones. Under low light (scotopic) conditions only the rods are active. Since there is only one type of rods, nocturnal vision is monochromatic. At daylight (photopic) light levels the rods become saturated and all the visual information comes from the cones. There are three types of cones, each one is tuned to a certain wavelength range. *Trichromatic generalization* explains how color stimuli can be expressed as additive mixtures of three fixed primary stimuli. At intermediate illumination levels (mesopic conditions) both rods and cones are active.

The spectral luminous efficiency functions $V(\lambda)$ and $V'(\lambda)$ are tabulated by the CIE and measure how brightness is perceived as a function of wavelength in the photopic and scotopic ranges respectively. In the mesopic range, the spectral luminous efficiency function depends on the illumination level, and we get it from tabulated experimental data in [145]. It can not be approximated by a linear combination of $V(\lambda)$ and $V'(\lambda)$.

By definition, $V(\lambda)$ is identical to $\overline{y}(\lambda)$ and setting the constant factor $k = K_m = 683\ \mathrm{lm} \cdot \mathrm{W}^{-1}$ in (4.1) the Y-tristimulus value becomes the luminance L of the color stimulus (measured in $\mathrm{lm} \cdot \mathrm{sr}^{-1} \cdot \mathrm{m}^{-2}$). In the same way, the luminance at scotopic levels is obtained with $V'(\lambda)$ and $k = K'_m = 1700\ \mathrm{lm} \cdot \mathrm{W}^{-1}$.

There are two values to be set concerning luminance at night, its mean value and variance. The mean value is set by the specification of desired ambient luminance by the user. While the variance is set by default, the user may modify it. The choice of variance

sets the maximum brightness, which is crucial if we have artificial light sources in the scene, as we will see in section 3.

We proceed in this manner and compute:

$$L'_0 = \int \beta(\lambda) S'(\lambda) V'(\lambda) d\lambda$$

Then we impose the selected mean and variance:

$$L' = \frac{L'_0 - \mu}{a} \frac{b}{\mu} + b$$

where b will be the desired mean at night, μ is the mean of L'_0 (the quotient $\frac{b}{\mu}$ allows the change of units), and a controls the variance ratio of L'_0 and L' (i.e. it allows us to set any given variance for L'). Unless otherwise stated, we use a value of $a = 1$. The user may modify a to increase the variance and make artificial lights brighter, for instance.

The results of applying the modification of luminance to figure 4.5(c) can be seen in figure 4.5(d).

4.2.4 Modification of contrast

Because the eye adjusts to its surroundings, human sensitivity to contrast depends on the adaptation luminance [135]. Contrast in our night image then must be different than in the original daylight scene. We can achieve this in two ways, either by approximating the eye's performance or by simulating the use of a given type of photographic film.

To approximate the eye's performance several models have been proposed [135, 142, 69]. We have implemented in our experiments a modification of the tone reproduction operator of Ward et al. [142], where they combine the rod and cone sensitivity functions and build a five-interval piecewise approximation for $\Delta L_t(L_a)$, see table 4.1. In their notation, ΔL_t is the 'just noticeable difference' at the given adaptation level L_a. We compute the resulting luminance L_n from the real-world luminance L_{rw} with $L_n = \frac{\Delta L_t(L'_a)}{\Delta L_t(L_{a_{rw}})} L_{rw}$, where L'_a and $L_{a_{rw}}$ are actually 8-neighbor local averages of L' and L_{rw} respectively.

If we choose to simulate the use of a given type of photographic film with a characteristic curve of γ_n, then our night luminance L_n will be approximated as $L_n = cL'(L^{\frac{1}{\gamma}})^{\gamma_n}$, where L' stands for the original L after modification in the previous steps. If γ is not known, we just assume $\gamma = 1$ and choose γ_n to achieve visually pleasing results. (These equations may be refined at will with more accurate descriptions of the characteristic curves of the film).

The modification of contrast following human vision is applied to figure 4.5(d) thus obtaining figure 4.5(e). Another example, in figure 4.7, compares both methods (in this figure we have also simulated the loss of acuity, see section 4.2.5.) Figure 4.7 (bottom right) shows the result of simulating a photographic film. To most observers, the result may be visually more pleasing than that obtained with an emulation of the eye's performance (figure 4.7 bottom left). The reason for this is that we are accustomed to a certain 'look' of night images on photographs and movies, which we usually prefer over 'real' looking night images, that may appear as too dark or without enough contrast. On the other hand, if the original day image has a sharp contrast, the 'night' result may be very convincing. In any case, the user may select in our algorithm the method that suits him/her best.

log10 of just noticeable difference	*applicable luminance range*
-2.86	$log10(L_a) < -3.94$
$(0.405log10(L_a) + 1.6)^{2.18} - 2.86$	$-3.94 \leq log10(L_a) < -1.44$
$log10(L_a) - 0.395$	$-1.44 \leq log10(L_a) < -0.0184$
$(0.249log10(L_a) + 0.65)^{2.7} - 0.72$	$-0.0184 \leq log10(L_a) < 1.9$
$log10(L_a) - 1.255$	$log10(L_a) \geq 1.9$

Table 4.1: Piecewise approximation for $log10$ of just noticeable difference for adaptation level L_a, $log10(\Delta L_t(L_a))$ (after [142]).

4.2.5 Loss of acuity: Diffusion

Visual acuity is the ability of the eye to see fine detail. The highest level of acuity is achieved at photopic levels and it decreases as the background luminance diminishes. But it also depends on contrast: increasing the level of contrast increases the resolution at a given luminance level. In previous work [69] this is modeled as isotropic diffusion. A 2D Gaussian filter is applied to the image, the radius of the Gaussian depending on the adaptation luminance. The idea is to cancel high spatial frequencies, following experimental data relating maximum visible spatial frequency (of a high contrast grating) and adaptation luminance. However, as neither contrast nor local luminance are taken into account, the resulting images seem unrealistic since they evoke the effects produced by an out of focus camera. In [60] there is a small correction to the computation of the Gaussian's size, but the procedure is basically the same. In [142] the authors approximate convolution with a Gaussian of explicitly varying radius: we shall prove that this approach causes ringing to appear in the resulting image. In [133] the authors choose a spatial filtering approach, performing low-pass filtering followed by sharpening. The

results are definitely better than with Gaussian blurring, but artifacts like ringing are present, and there are several parameters that are set subjectively.

Our contribution is the following: we will introduce a novel diffusion PDE that models the spatial summation principle, takes luminance into account and respects contrast, produces no ringing, is stable, very easy to implement and fast. We shall start by discussing a set of axioms modeling the loss of visual acuity, which will lead to a family of nonlinear diffusion equations which take into account local luminance and respect contrast. These PDE's are called Fast Diffusion Equations, and are related to Porous Medium Equations. The method we introduce here could also be used to simulate loss of acuity on images shot at night, since film and cameras can not emulate this human vision process.

Psychophysical and physiological experiments cited in [51] show that neighbouring photoreceptors in the retina interact accordingly to the level of illuminance at each point. That is, the light perceived at a single point in the retina not only creates an excitation at the photoreceptor at this site, it produces a lateral excitation as well and all of them are combined additively. This process is called spatial summation and the extent of the area of summation varies inversely with the local illuminance.

Indeed, in [51] the authors start from a set of axioms for intensity dependent spatial summation to determine the point spread function relating the input image I to the output image $O(I)$. The basic idea is that each input point (x, y) contributes with a non-negative point spread value to every output point (p, q), the size of this contribution depending on the intensity value $I(x, y)$ and the distance from (x, y) to (p, q). Thus the point spread function (PSF) has the form $S((x, y), (p, q), I)$ and this gives the contribution from (x, y) to (p, q) when the input intensity at (x, y) is I. Then they assumed

(i) S is nonnegative: each input point (x, y) contributes with a non-negative point spread value to every output point (p, q), the size of this contribution depending on the intensity value $I(x, y)$ and the distance from (x, y) to (p, q). Thus the point spread function has the form $S((x, y), (p, q), I)$ and this gives the contribution from (x, y) to (p, q) when the input intensity at (x, y) is I.

(ii) S is spatially homogeneous and circularly symmetric, hence $S = S(d^2, I)$, where $d^2 = (x - p)^2 + (y - q)^2$,

(iii) The effective area covered by the PSF around each input point varies inversely with the intensity at that point, which can be translated into the relation

$$S(d^2, I) = Q(I)S(Q(I)d^2, 1)$$

where $Q(I)$ is an increasing function of I. Indeed, in [51], the authors took $Q(I) = I$.

(iv) If we write $S(r^2) = S(r^2, 1)$ then we normalize the integral of $S(x^2 + y^2)$ over the (x, y) plane to be equal to 1, i.e.,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S(x^2 + y^2)\, dx\, dy = 1.$$

The output is given by

$$O(I)(p, q) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x, y)S(d^2, I(x, y))\, dx\, dy. \tag{4.2}$$

In [51], the output was interpreted as the response of a retinal cell and it was taken as $O(I)(p, q) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S(d^2, I(x, y))\, dx\, dy$. Notice its difference with (4.2) in which the output of the filter is an intensity image given by a convolution with a PSF with intensity depending variance. As in [51], our main example of point spread function will be the Gaussian, but we could also choose a radial function with zero first-order moments and finite second order moments. To relate $Q(I)$ with the variance of the Gaussian function, we use the notation $Q(I) = \frac{1}{\sigma^2(I)}$. Then we may write

$$S(d^2, I(x, y)) = \frac{1}{2\pi\sigma^2(I)} exp^{-\frac{[(x-p)^2+(y-q)^2]}{2\sigma^2(I)}} \tag{4.3}$$

Arguing as in [51], we observe that the spatial summation (4.2) does *not* satisfy the maximum principle.

Theorem 1 *We have $O(I)(p, q)$ as in eq.(4.2), and $I(x, y)$ is a step function valued I for $x < 0$ and $I+D$ for $x \geq 0$. Then there exists a positive real number Θ s.t. $O(I)(p, q) > I+D$ if $p > \Theta$.*

In other words, ringing is *guaranteed* to appear if we use directly this spatial summation (see proof in appendix B), or if, equivalently, we perform convolution with Gaussians of varying radius, as in [142].

We can try to restore the maximum principle (and its locality) if we know the infinitesimal action of the filter. For that following [4], we introduce an scale parameter $t > 0$, and write $t\sigma(I)$ instead of $\sigma(I)$. Next theorem gives the asymptotic behavior of $O(I, t)(p, q)$ as $t \to 0$.

Theorem 2 *Assume that we have a general PSF satisfying axioms (i)-(iv), then we have $O(I, t)(p, q) = I(p, q) + Ct^2\Delta(I\sigma^2(I))(p, q) + o(t^2)$.*

See appendix B for the proof of this theorem. If we write the above expansion in the form $\frac{O(I,t)(p,q)-I(p,q)}{Ct^2} = \Delta(I\sigma^2(I))$ and let $t \to 0^+$, we obtain the following nonlinear diffusion equation to model the loss of acuity

$$I_t = \Delta(I\sigma^2(I)). \tag{4.4}$$

From the mathematical point of view, existence and uniqueness results for initial data $I_0 \in L^1(\Omega) \cap L^\infty(\Omega)$, together with a comparison principle for (4.4) have been proved in [18] when the function $\varphi(r) = r\sigma^2(r)$ is continuous and increasing in $\mathbb{R}$ (we refer also to [47, 138] and references therein for the mathematical treatment of this kind of equations). We stress the fact that going from (4.2) to (4.4) permits us to guarantee the maximum principle and, therefore, no ringing behavior is exhibited by solutions of these equations. The usual rigorous derivation of this would be based on the computation of the iterates $O(I, \frac{t}{n})^n$ and its limit as $n \to \infty$, though the usual convergence Theorem [81] does not apply and the convergence of the iterates $O(I, \frac{t}{n})^n$ is an open question.

To write some particular instances of (4.4) we write it as $I_t = \operatorname{div}(\varphi'(I)\nabla I)$. Then we choose $\varphi'(I) = \frac{1}{(1+\alpha I)^\beta}$ with $\alpha, \beta > 0$ as generic models for φ which exhibit a power behaviour for large intensities and are differentiable at $I = 0$. Thus, we shall consider

$$I_t = \nabla \cdot \left(\frac{\nabla I}{(1+\alpha I)^\beta} \right) \tag{4.5}$$

The parameter α controls the level of diffusion. In equation (4.5), the anisotropy of the diffusion is controlled by the local luminance values. Pixels with high luminance values are diffused less than pixels with low luminance. Recall that usually the anisotropy is controlled by the magnitude of the *gradient* (see [136] and the seminal work by Perona and Malik [120], a model in which images are smoothed while preserving edges). We show in figure 4.2 some examples after applying the PDE (4.5) to a pair of square waves of different amplitude for different values of the parameter α ($\beta = 1$). Note that $\alpha = 0$ corresponds to an isotropic diffusion.

The level of diffusion is also controlled by parameter β, the diffusion is reduced as we increase the value of β, this effect being more pronounced for high luminances. These facts have been checked experimentally. In our algorithm we set $\beta = 1$. Figure 4.3(a) shows the effect of applying the PDE (4.5) to square grating signals of same contrast but different mean luminance (contrast is taken as the quotient $(I_{max} - I_{min})/I_{mean}$). The result proves that the square grating of higher luminance is less diffused than the lower one. On the other hand, figure 4.3(b) shows the diffusion effect on two square grating functions of different contrast but with the same mean luminance. The diffusion

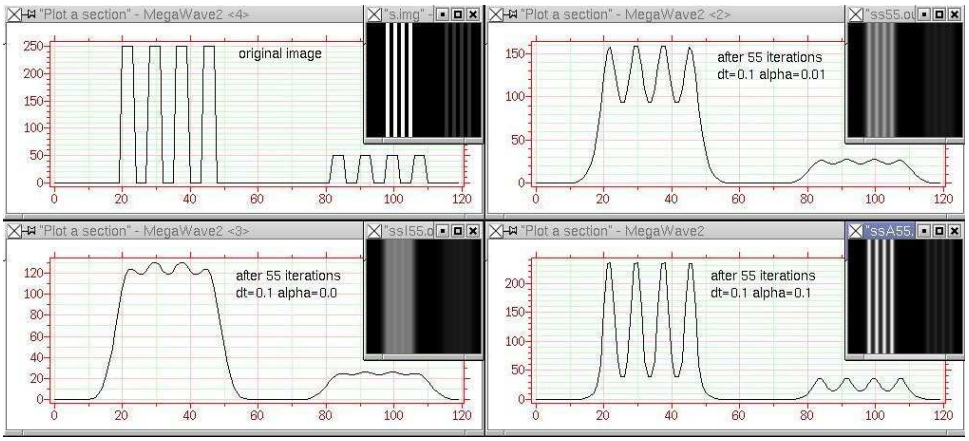


Figure 4.2: Loss of acuity simulation via PDE's: comparison for different values of α and two different levels of local luminance.

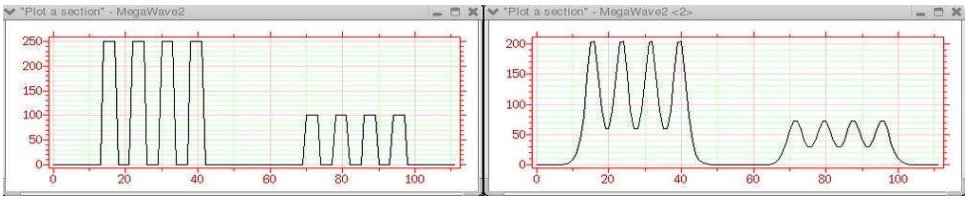
respects contrast: the left-hand square grating has higher contrast than the right-hand one, before and after diffusion. Therefore, the square grating on the left is perceived as sharper. In conclusion, we have experimentally checked that the behavior exhibited by solutions of equation (4.5) is consistent with experimental data on human vision acuity showing that spatial summation is inversely proportional to luminance and respects contrast. Furthermore, the results are free from ringing and other visual artifacts that other approaches bring. We compare in figure 4.4 our approach to different techniques to simulate the loss of acuity at night. Notice that both the convolution with gaussians of varying radius [142] and low pass filtering with sharpening [133] produce different types of artifacts. If we convolve with a Gaussian of fixed radius depending on the adaption luminance [60] artifacts do not appear but the result has the effect of an out of focus blur.

For simplicity we apply this equation to each of the three color components separately, though an equivalent vector-diffusion equation could be devised after [132].

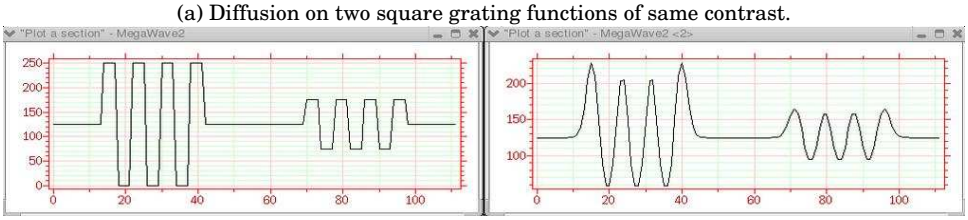
The numerical implementation of each scalar diffusion equation is done with a scheme based on finite differences. If we consider the representation of a color component value at each point of the image grid as $I_{i,j} = I(i,j)$, with $1 \leq i \leq N$ and $1 \leq j \leq M$ (N is the number of lines and M the number of columns) and the finite differences at both sides to represent the spatial derivatives, we get:

$$\nabla^+ I_{i,j} = (I_{i+1,j} - I_{i,j}, I_{i,j+1} - I_{i,j})$$

$$\nabla^- I_{i,j} = (I_{i,j} - I_{i-1,j}, I_{i,j} - I_{i,j-1})$$



(a) Diffusion on two square grating functions of same contrast.



(b) Diffusion on two square grating functions of different contrast.

Figure 4.3: Loss of acuity simulation via PDE's: diffusion depends on luminance. Observe that, after diffusion, the left grating is still more contrasted than the right one (25 iterations, dt=0.1, α=0.01, β=1).

The numerical scheme used is:

$$I^{n+1} = I^n + \Delta t \begin{cases} \nabla^- \cdot \left(\frac{\nabla^+ I}{1+\alpha I} \right) & \text{if } n = \dot{2} \\ \nabla^+ \cdot \left(\frac{\nabla^- I}{1+\alpha I} \right) & \text{if } n = \dot{2}+1 \end{cases} \tag{4.6}$$

where I^n is the image I at time n and Δt is the time step between two iterations.

We have obtained experimentally the time of diffusion T necessary to lose at each level of darkness the details whose frequency is above the Highest Resolvable Spatial Frequency in accordance with data from Shaler in [69]. Firstly, we have chosen three different luminances and we have created three images of a square wave grating with image dimensions corresponding to the width subtended by one degree of arc at a viewing distance of one meter. Each of these images contains a number of cycles just above the maximum number of cycles detectable at that level of luminance. The maximum value of the image is fixed to 255 for a log luminance of 3 and we decrease it proportionally to the decrease in log luminance. Then, we fix $\alpha = 0$ (isotropic diffusion) and $\Delta t = 0.1$ (below 0.25 which is the CFL stability condition for the Perona Malik equation [120]) and we find the necessary number of iterations to achieve a uniform image at a distance of one meter. Finally, we interpolate linearly the number of iterations between these points and we obtain the following expression for the number of steps:

$$steps = \begin{cases} 12 - 36log(L) & \text{if } log(L) < 0 \\ 12 - 6.4log(L) & \text{if } 0 \leq log(L) < 1.875 \\ 0 & \text{if } log \geq 1.875 \end{cases}$$

We have set α to 0.01, obtaining very good results for natural images in a wide range of ambient luminances. Please note that this value of α is fixed in the algorithm and thus it is not a parameter that the user has to change. The robustness of the equation

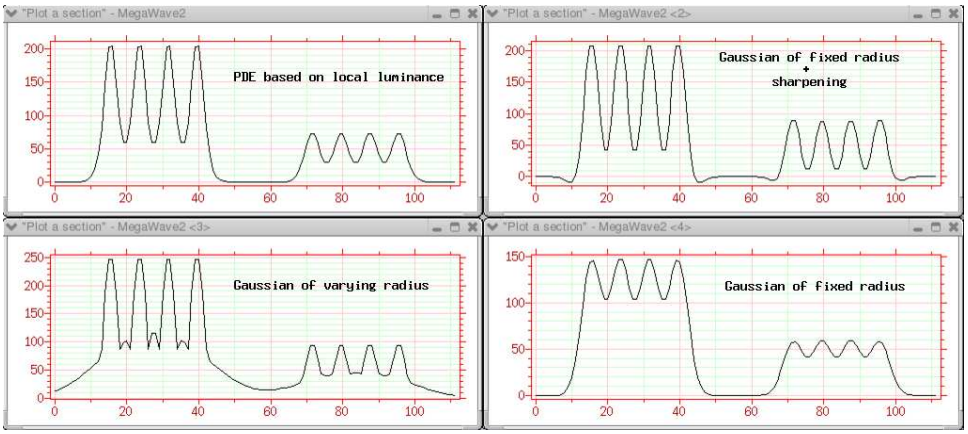


Figure 4.4: Comparison of different diffusion methods simulating loss of acuity at night. Notice that our method (top left) produces no artifacts, neither the Gaussian of fixed radius (bottom right) but this approach does not simulate the spatial summation principle depending on local luminance.

makes it suitable for video sequences, no temporal artifacts appear (see examples in http://www.tecn.upf.es/~mbertalmio/day4nite).

The results of applying this modification to figure 4.5(e) can be seen in figure 4.5(f). Figures 4.5(f) to 4.5(i) and different images in figure 4.9 show how fine details are lost as the luminance level decreases. Also, for any given image and luminance level, more detail is lost in darker regions than in light ones. Notice how the achieved effect of loss of acuity is very different from an out-of-focus blur. In particular, in figure 4.9, as the luminance decreases it becomes harder and harder to read the numbers on the wall, or the text in banners, books and cardboard boxes, just as it happens to our eyes when the light grows dim. But *pronounced* edges are preserved, as we can see in the dark bands on the wall, or the white sheets of paper hanging from the tables in this same figure.

4.3 Examples

As mentioned earlier, the results of our experiments necessarily look very different depending on the monitor and room light conditions. We recommend the reader to set the monitor's brightness level to 50%, the monitor being in a well lit room with no direct light on the screen. This would be an approximation only, for correct evaluation calibrated hardware is required.

Figures 4.5 to 4.9 show several results, for different images, night illumination levels and contrast-modification methods. Notice how these images look realistic. In partic-

ular, notice how colors have become less saturated but we may still tell them apart, they are not predominantly different shades of blue as we would get with conventional Day for Night. Brightness and contrast are what we would expect in a night scene, objects do not have an unreal illumination. Realism is enhanced by the controlled loss of resolution, which blurs small (and not too bright) details, as our eyes do at night. We show in figure 4.6 that the algorithm is robust to the natural illuminant that we assume: the results do not vary much using three different CIE standard illuminants: D_{55}, D_{65} and D_{75}; which correspond to natural daylight at different color temperatures.

Our algorithm has been developed with the assumption that all light in the scene is natural, i.e. that the illuminant is one for the whole image. We are currently working on how to circumvent this constraint, so we can introduce artificial light sources in our images. The problem is that it is very hard to approximate, at each pixel location, the interaction between different light sources, with different intensities and spectral power. Figure 4.8 shows a test for one image of this sort, where we have assumed that the highest luminances in the scene correspond to the artificial light source. We have increased the luminance there modifying the original variance by a factor greater than 1. The results are more realistic but this method can fail at points with high luminance but which do not correspond to light sources (see white line on the road in figure 4.8).

If the original image presents a cloudy sky, as in figure 4.9, we cannot achieve a dark sky in the night scene. If this is a problem we could choose to avoid showing the sky when shooting, as done in traditional Day for Night. We could also explore a way to segment the sky and treat differently the pixels in that region.

The whole process takes less than ten seconds in a regular PC for a 512x768 24 bits RGB image. Within that execution time, seven seconds are dedicated to the diffusion process in the three color components in an example where seven iterations of the equation are required. The use of a GPU (Graphics Processor Unit) may speed-up the anisotropic diffusion process by an order of magnitude [25], bringing the algorithm closer to real-time. We have also verified that the parallelization of the code does not create artifacts. By constructing a color LUT (Look-Up Table) the speed may be increased greatly from our current implementation, where we deal with each pixel separately. Also in moving pictures there is great space and time redundancy, another source for speed-ups.

4.4 Conclusion and future research

We have introduced a digital Day for Night algorithm that achieves realistic results. Our algorithm performs modification of the spectrum for the night illuminant, desaturation of the colors, brightness modification according to wavelength, contrast modification according to luminance adaptation levels, and non-uniform and non-linear loss of resolution. We use a set of simple equations, based on real physical data and visual perception experimental data. To simulate the loss of resolution we have introduced a novel diffusion equation, which is well-posed, has existence and uniqueness results, and is also monotonicity preserving, so no ringing may occur. The robustness of the equation makes it suitable for video sequences, no temporal artifacts appear. The user only has to provide the original day image and the desired level of darkness of the result. More accurate results are obtained if our algorithm is provided with the characteristic curve of the photographic film used. The whole process from original day image to final night image takes a few seconds, all the computations being local, but optimizations could easily speed up the process in an order of magnitude.

The main limitation of our algorithm is that it has been developed with the assumption that all light in the scene is natural, i.e. that the illuminant is one for the whole image. We are currently working on how to circumvent this constraint. The input images are in RGB format coming from digital video or obtained from a scanned film. Part of future research is to include emulations of the film developing process, and to reformulate our algorithm in terms and units that cinematographers use.

(a) original image

(b) after change of illuminant

(c) after change of chromaticity

(d) after modification of luminance at 1 log cd/m^2 (without change of contrast nor diffusion)

(e) after change of contrast following human vision (1 log cd/m^2)

(f) simulated night scene at 0.8 log cd/m^2

(g) simulated night scene at 0.5 log cd/m^2

(h) simulated night scene at 0.3 log cd/m^2

(i) simulated night scene at 0.1 log cd/m^2

Figure 4.5: Original image, results after some steps of the algorithm and simulated night scenes with different levels of ambient luminance.

Figure 4.6: Results for three simulated night scenes at 1 log cd/m^2 with different assumptions for the day illuminant: D_{55} (left) D_{65} (middle) and D_{75} (right). Original image in figure 4.5(a)

Figure 4.7: Original scene (top), emulating night vision (bottom left), emulating a given film stock (bottom right).

(a) original scene

(b) result of our algorithm without changes

(c) result after increasing the variance of the original luminance ($a = 0.1$, see section 4.2.3)

Figure 4.8: A scene with artificial light sources and two simulated night scenes at 0.1 log cd/m^2 with 11 iterations of diffusion.

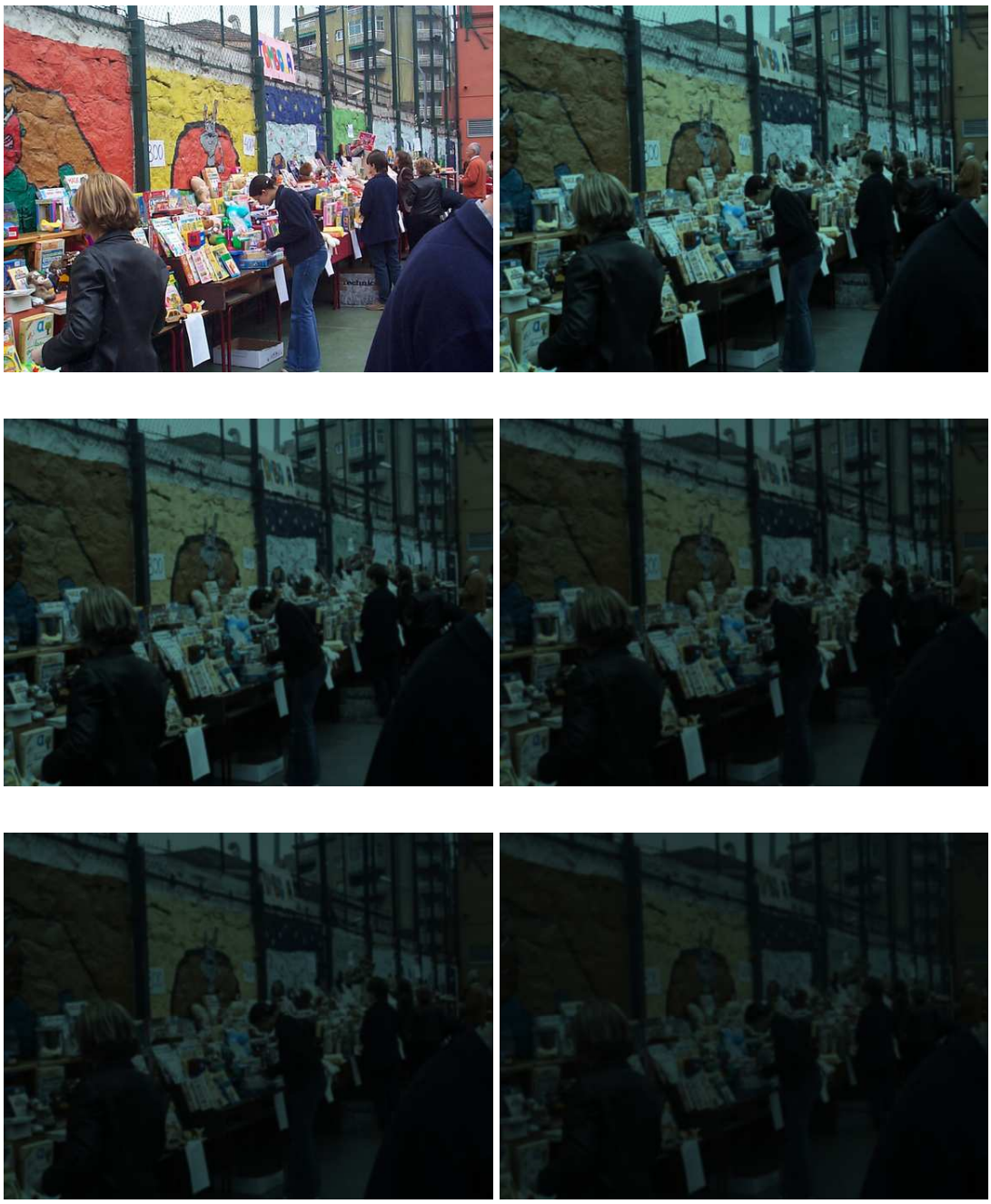


Figure 4.9: Some results with decreasing values of ambient luminance: 1, 0.6, 0.3, 0.1 and -0.1 log cd/m^2 , 5, 8, 10, 11 and 15 iterations of diffusion respectively from left to right and from top to bottom.

Appendix B

Proofs of Theorems

Proof of Theorem 1 We compute the integral $O(I)(p,q)$ resulting from the convolution of the PSF given by (4.3) and the step function valued I for $x<0$ and $I+D$ for $x\geq 0$,

$$O(p,q)=\int_{\mathbb{R}^2} S(d^2,I(x,y))I(x,y)dxdy=\int_{\mathbb{R}}\int_{-\infty}^{0} IS(d^2,I)dxdy+\int_{\mathbb{R}}\int_{0}^{\infty}(I+D)S(d^2,I+D)dxdy \tag{B.1}$$

We have, after assumption (iv), that

$$\int_{\mathbb{R}^2}(I+D)S(d^2,I+D)dxdy=I+D$$

so we will have $O(p,q)>I+D$ in (B.1) for those p such that

$$\int_{\mathbb{R}}\int_{-\infty}^{0} IS(d^2,I)dxdy>\int_{\mathbb{R}}\int_{-\infty}^{0}(I+D)S(d^2,I+D)dxdy \tag{B.2}$$

Specifying for the PSF of the form (4.3) this relation is implied by the following inequality

$$e^{\frac{Q(I+D)-Q(I)}{2}|x-p|^2}>\frac{(I+D)\sqrt{Q(I+D)}}{I\sqrt{Q(I)}}\quad \forall x<0$$

which may be written in an equivalent form as

$$|x-p|^2>\frac{2}{Q(I+D)-Q(I)}log\left(\frac{(I+D)\sqrt{Q(I+D)}}{I\sqrt{Q(I)}}\right)\forall x<0 \tag{B.3}$$

Since $Q(I)$ is an increasing function of I and $I+D>I$, we have that the right hand side of (B.3) is positive. Observe that if we choose $p>\Theta$, where Θ is the square root of the right hand side of (B.3), then (B.3) holds, and therefore condition (B.2) holds also. □

Proof of Theorem 2 Let $\mathbf{x} = (x, y)$, $\mathbf{p} = (p_1, p_2)$ and $\mathbf{z} = (z_1, z_2) \in \mathbb{R}^2$.

Definition B.0.1 *Consider the real function $g(\mathbf{x}) \in L^1(\mathbb{R}^2)$ which represents a smoothing kernel to be convolved with images. We say that $g(\mathbf{x})$ is radial if $g(\mathbf{x}) = g(|\mathbf{x}|)$ only depends on the norm of $\mathbf{x}$. We say that it is pseudo-radial if it satisfies the following relations:*

$$\int_{\mathbb{R}^2} g(\mathbf{x})d\mathbf{x} = 1 \tag{B.4}$$

$$\int_{\mathbb{R}^2} \mathbf{x}_i g(\mathbf{x})d\mathbf{x} = \int_{\mathbb{R}^2} \mathbf{x}_i\mathbf{x}_j g(\mathbf{x})d\mathbf{x} = 0 \quad \forall\, i,j=1,2 \mid i\neq j \tag{B.5}$$

$$\int_{\mathbb{R}^2} \mathbf{x}_i^2 g(\mathbf{x})d\mathbf{x} = \nu \quad \forall\, i=1,2 \tag{B.6}$$

Then, the rescaling of g, $g_h(\mathbf{x}) = \frac{1}{h} g\left(\frac{\mathbf{x}}{h^{1/2}}\right)$ maintains (B.4) and (B.5). Integral (B.6) gives now $h^{1/2}\nu$.

We call $O(I)(\mathbf{p})$ the result of convolving the image I with the kernel g_h:

$$O(I)(\mathbf{p}) = \int I(\mathbf{x}) \frac{1}{h} g\left(\frac{\mathbf{p}-\mathbf{x}}{h^{1/2}}\right) d\mathbf{x}$$

where $h(t, I)^{1/2} = t\sigma(I)$ and $\sigma(I) = I^{-\alpha}$ with $\alpha > 0$. In order to compute the integral we make the change of variables $\mathbf{z} = (\mathbf{p} - \mathbf{x})/h^{1/2}$. The change of variables gives $\mathbf{x} = \mathbf{p} - t\sigma(I)\mathbf{z}$. Differentiating these expressions and writing the result in matrix form yields

$$(\mathrm{Id} + t\mathbf{z} \otimes \nabla\sigma)d\mathbf{x} = -t\sigma d\mathbf{z} \tag{B.7}$$

where Id is the identity matrix and $\otimes$ represents the tensorial product.

Since $\det(\mathrm{Id} + tM) = 1 + t\,\mathrm{trace}(M) + t^2 \det(M)$ and $\det(M) = \det(\mathbf{z} \otimes \nabla\sigma) = 0$, the product of forms in (B.7) gives

$$dx\, dy = t^2\sigma^2 dz_1 dz_2 - t^3\sigma^2 <\mathbf{z}, \nabla\sigma> dz_1 dz_2 + t^4\sigma^2 <\mathbf{z}, \nabla\sigma>^2 dz_1 dz_2$$

We can now compute the difference between the filtered and the original image:

$$\begin{aligned} O(I)(\mathbf{p}) - I(\mathbf{p}) &= \int \frac{I(\mathbf{p} - t\sigma\mathbf{z})}{t^2\sigma^2} g(\mathbf{z})dxdy - \int I(\mathbf{p})g(\mathbf{z})d\mathbf{z} \\ &= \int (I(\mathbf{p} - t\sigma\mathbf{z}) - I(\mathbf{p}))g(\mathbf{z})d\mathbf{z} - t\int I(\mathbf{p} - t\sigma\mathbf{z}) <\mathbf{z}, \nabla\sigma> g(\mathbf{z})d\mathbf{z} \\ &\quad + t^2 \int I(\mathbf{p} - t\sigma\mathbf{z}) <\mathbf{z}, \nabla\sigma>^2 g(\mathbf{z})d\mathbf{z} = A + B + C \end{aligned} \tag{B.8}$$

The Taylor development of $I(\mathbf{p} - t\sigma\mathbf{z})$ up to second order:

$$I(\mathbf{p} - t\sigma\mathbf{z}) = I(\mathbf{p}) - t\sigma(\mathbf{p} - t\sigma\mathbf{z}) <\mathbf{z}, \nabla I> + \frac{t^2}{2}\sigma^2(\mathbf{p} - t\sigma\mathbf{z})I_{ij}z_i z_j \tag{B.9}$$

here, the convention of implicit summation over repeated indices is used and I_{ij} denotes the second derivate of I with respect to components i and j for $i = 1, 2$. In the same manner we develop the term $\sigma(\mathbf{p} - t\sigma\mathbf{z})$ and substitute it in (B.9) rejecting all terms of $O(t^3)$

$$I(\mathbf{p} - t\sigma\mathbf{z}) = I(\mathbf{p}) - t\sigma(\mathbf{p}) < \mathbf{z}, \nabla I > + t^2\sigma(\mathbf{p}) < \mathbf{z}, \nabla\sigma >< \mathbf{z}, \nabla I > + \frac{t^2}{2}\sigma^2(\mathbf{p})I_{ij}z_iz_j \quad \text{(B.10)}$$

We can now compute the term A in (B.8) making use of (B.10) and (B.5)

$$A = t^2\sigma\sigma_i I_j a_{ij} + \frac{t^2}{2}\sigma^2 I_{ij}a_{ij} \quad \text{(B.11)}$$

where $a_{ij} = \int z_i z_j g(\mathbf{z})d\mathbf{z} = \delta_{ij}$ and subindices i and j in σ and I represent first derivate with respect to component i and j respectively.

As regards the term B in (B.8) we substitute (B.10) and reject terms in $O(t^3)$

$$B = -t\int I(\mathbf{p}) < \mathbf{z}, \nabla\sigma > g(\mathbf{z})d\mathbf{z} + t^2\int \sigma(\mathbf{p}) < \mathbf{z}, \nabla I >< \mathbf{z}, \nabla\sigma > g(\mathbf{z})d\mathbf{z}$$

Since the first moments of g are null, and developing $\nabla\sigma(\mathbf{p} - t\sigma\mathbf{z})$ in B, terms up to $O(t^3)$ give

$$B = t^2 I\sigma\sigma_{ii} + t^2\sigma I_i\sigma_i \quad \text{(B.12)}$$

Finally, we also substitute in the term C of (B.8) the Taylor development of $I(\mathbf{p} - t\sigma\mathbf{z})$ and $\nabla\sigma(\mathbf{p} - t\sigma\mathbf{z})$ and retain terms up to $O(t^3)$

$$C = t^2 I\sigma_i\sigma_i \quad \text{(B.13)}$$

Replacing expressions (B.11), (B.12) and (B.13) in (B.8) we obtain

$$O(I)(\mathbf{p}) - I(\mathbf{p}) = t^2\nu\left(2\sigma < \nabla\sigma, \nabla I > + \frac{1}{2}\sigma^2\Delta I + I\sigma\Delta\sigma + I|\nabla\sigma|^2\right) \quad \text{(B.14)}$$

Indeed, the right hand side of the equality (B.14) corresponds to $\Delta(I\sigma^2)$

$$\Delta(I\sigma^2) = \nabla\cdot(\nabla(I\sigma^2)) = \nabla\cdot(\sigma^2\nabla I + I2\sigma\nabla\sigma) = 2\left(2\sigma < \nabla\sigma, \nabla I > + \frac{\sigma^2}{2}\Delta I + I\sigma\Delta\sigma + I|\nabla\sigma|^2\right).$$

So, (B.14) reduces to

$$O(I)(\mathbf{p}) - I(\mathbf{p}) = \frac{\nu}{2}t^2\Delta(I\sigma^2(I)) + O(t^3).$$

□

Part III

Inpainting Surface Holes

Chapter 5

Geometric approaches to surface reconstruction

ABSTRACT

In this chapter we study geometric approaches for filling-in surface holes. First, we review the importance of the elastica in image inpainting and its relation with the Willmore functional. We also review some surface inpainting techniques in the literature. The basic idea of our approach is to represent the surface of interest in implicit form, and fill-in the holes with a scalar, or system of, geometric partial differential equations, often derived from optimization principles. These equations include a system for the joint interpolation of scalar and vector fields, a mean curvature diffusion flow, the minimization of the Laplacian of the distance function and two other interpolation methods based on the solution of the Laplace equation and an absolutely minimizing Lipschitz extension. The theoretical and computational framework, as well as examples with synthetic and real data, are presented.

5.1 Introduction

The retouching process of damaged regions in art restoration is often called *inpainting*. Paint losses are filled up to the level of the surrounding paint, and then colored to match. This topic has closed applications in image processing: image and video restoration (e.g., removal of scratches or dust spots). However, image inpainting techniques are also used for removing objects to produce special effects or even for recovering lost blocks in image coding and transmission. The basic idea behind the computer algorithms that have been proposed in the literature is to fill-in these regions with available information from their surroundings. This information can be automatically detected as in [26, 62], or hinted by the user as in more classical texture filling techniques

[56, 87, 127]. Several names have been used for this filling-in operation, including *dis-occlusion* in [16, 110], or *inpainting* in [22, 26, 27]. The interested reader is also referred to the works [15, 42, 116, 122].

As regards the 3D inpainting it also has practical applications. For example, surfaces obtained from range scanners often have holes: regions where the 3D model is incomplete. The main cause of holes are occlusions, but these can also be due to low reflectance, constraints in the scanner placement, or simply lack of sufficient coverage of the object by the scanner. This is frequently observed in the scanning of art pieces [103], and is in part due to the fact that complicated geometry has a lot of self-occlusions and details. Art pieces also impose significant restrictions on the scanner placement. Other times, 3D inpainting is useful to restore damaged regions of sculptures or museum objects scannings [49]. Holes are also observed in common scenarios where LADAR data is collected (e.g., a house behind an occluding tree), and in all the major areas where range scanners are used. With the increasing popularity of range scanners as 3D shape acquisition devices, with applications in geoscience, art, medicine, manufacturing and defense, it is very important to be able to inpaint this missing information. This is often needed for post-processing as well as for presentation.

Our work is inspired by the one reported in [55], and it is presented as an alternative to this method. This pioneering work addressed the problem of hole filling via isotropic diffusion of volumetric data (that is, iterative Gaussian convolution of some distance function to the known data). The approach proposed by these authors addresses holes with complicated topology, a task very difficult with mesh representations. The reader is directed to this paper for an excellent and detailed description of the nature of holes in scanning statues and for a literature review in the subject. We should only note that most algorithms on reconstructing surfaces from range data are point-cloud reconstruction based and treat holes as regions with low sampling density, thereby interpolating across them [7, 14, 21, 50, 61, 88]. Of course, these algorithms often do not distinguish between a real hole in the data and one due to the lack of sampling, and equally fill or fail to fill both cases in the same fashion. Other point-cloud methods evolve a surface over time until it approximates the data [48, 144, 148], or fit a set of 3D radial basis functions to the data, compute a weighted sum of them and use a level set of this last function as reconstructed surface [58, 34]. Mesh based methods for surface reconstruction [137, 53, 143] can perform hole filling as a post-process or integrate hole filling into surface reconstruction [53]. One of our proposed models is closed related to the one presented in [49] where the authors use the Willmore flow with a finite

element implementation. In contrast with their work, our model works on implicit surfaces, thereby allowing for more complicated hole topologies, and also naturally leads to systems of low order differential equations.

The first algorithm here proposed is an extension of a previous work on image inpainting [15, 16, 26] (see also [22, 27, 42, 110, 116, 122]). In particular, we show how to adapt the variational formulation presented in [15, 16] to the problem of surface hole filling. As in [55], the use of volumetric data (that is, the surface is represented as the zero level-set of a function) brings us topological freedom. In contrast with [55], we use a system of coupled anisotropic (geometric) partial differential equations designed to smoothly continue the isophotes of the embedding function, and therefore the surface of interest (as the zero level isophote). These equations are based on the geometric characteristics of the known surface (e.g., the curvatures), and as [55], are applied only at the holes and a neighborhood of them (being these equation anisotropic and geometry based, they lead to a slightly slower algorithm than the one reported in [55], as expected with geometric flows). A preliminary version of this (first) model was presented in [139]. We formalize this and improve it here with an automatic initialization method. This initialization is based on the computation of a conical neighborhood $\mathcal{F}$ of the known part of the surface, call it $\mathcal{S}$, where the distance function is uniquely attained. Thereby we can define the signed distance function d_s and ∇d_s is the extension of the unit normal to $\mathcal{S}$ to a neighborhood of it. This construction also helps us to label both parts of the surface as interior and exterior, and this is useful in this first method.

We also develop other curvature based hole surface reconstruction methods. The first of them is based on a variational model which integrates the Laplacian of a distance function (i.e., a function which satisfies $|\nabla D| = 1$, and $D = d_s$ in $\mathcal{F}$) in a open set containing the hole. Recall that the Laplacian of the distance function gives the mean curvature of its level sets. The other method is more heuristic and is based on the diffusion of a function which represents mean curvature of level sets of an underlying implicit function.

Finally, we have also present simpler methods based on Laplace equation and the so-called AMLE model which permit to reconstruct a function which is distance-like near the known part of the surface and whose zero level set can be interpreted as the reconstructed surface. If our interest is just to find a smooth reconstruction, this approach may be sufficient. If one wants a reconstruction which is based on minimizing mean curvature, it can serve as an initialization.

These algorithms, except the one based on curvature diffusion which is less reliable,

exhibit a similar behavior in reconstructing surface holes for synthetic and real data. As mentioned above, the reconstructions based on the Laplace or AMLE equation can be used as initializations for the curvature based approaches.

The remainder of this chapter is organized as follows. Section 5.2 describes the use of the elastica in the context of image inpainting, its relation to the Willmore functional on surfaces and it describes other related surface inpainting techniques present in the literature. In Section 5.3 we explain how we set the reliable information (in a neighborhood around the hole) to be used in the inpainting process. This is a common step in all the approaches here studied. In section 5.4 we present an automatic initialization method to improve the surface inpainting model introduced in [139]. Section 5.5 presents the two curvature based approaches: a variational one minimizing the absolute value of curvature, and an heuristic one, based on the diffusion of curvature. Section 5.6 describes two simple methods for surface reconstruction based on the Laplace equation and the so-called AMLE model. Section 5.7 describes the numerical algorithms used in our computations. In Section 5.8 we present some numerical experiments on hole filling obtained with the algorithms previously introduced. Finally, in Section 5.9, we summarize the main conclusions of this work.

5.2 Related work

5.2.1 The elastica and its role in image disocclusion

One of the perceptual organization principles of the Gestalt theory in visual perception [96] is the *principle of good continuation*: our visual system is able to complete partially occluded shapes in such a way that the visible boundaries of the occluded object are continued smoothly. Contours based on smooth continuity are preferred to abrupt changes of direction. The occluded and occluding boundaries form a particular configuration called *T-junction*. The visual system then smoothly continues the occluded boundary between T-junctions (see figure 5.1).

One of the first works that introduced the smooth continuation principle in image processing using powers of the curvature as smoothness measure is due to Nitzberg, Mumford and Shiota [115] in the context of image segmentation. They segmented the objects according to their depth in the scene and reconstructed the occluded boundaries between T-junctions using curves minimizing the Euler's elastica. Thus, their proposal

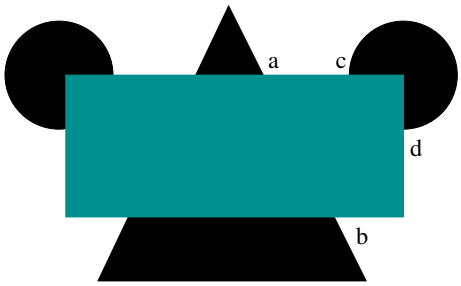


Figure 5.1: The visual system tends to connect the pairs of T-junctions a-b and c-d (rather than a-c and d-b).

of completion curve is the curve C that minimizes the energy

$$\int_C (\alpha + \beta k^2) ds \tag{5.1}$$

where k denotes the mean curvature of C, ds its arclength and α, β are positive constants. Indeed, the elastica has been widely used in computer vision due to its interpolation capability and the Bayesian principles underlying on it [112].

Masnou and Morel [110] proposed a variational approach for image disocclusion based on the elastica (5.1) and which minimizes also the angles formed by the tangents of completion curves and the tangents of the visible parts of occluded boundaries at T-junctions. The energy term based on the elastica used a power p of the curvature: $\int_C (\alpha + \beta |k|^p) ds$. The optimal disocclusion minimizes the addition of curve functionals over all the gray levels of the image and over the family of completion curves associated to each level set. The authors in [110] proved that there is an optimal disocclusion for each $p \geq 1$. Then, Ambrosio and Masnou [6] observed that the criterion based on the elastica could be computed not only in the completion curve but also in a small piece of the partially occluded level line outside the occluded part. Then, with this reinterpretation there was not necessary to include the restriction on the tangents and the functional was only based on the elastica term.

An analysis of the inpainting problem in [42] by Chan and Shen shows the necessity of including a curvature term in order to have a connectivity principle. Actually, most of the inpainting techniques in the literature are based on minimizing the curvature [110, 15, 16, 45]. In [45] is shown that previous techniques like [26, 43] originally not based on the elastica actually include both (in [26]) or one (in [43]) of the basic mechanisms arising in the elastica: transport and diffusion.

Relation with the Willmore energy

Clarenz et al. [49] address the problem of surface reconstruction of damaged regions in a surface using the Willmore flow. Let us consider a two-dimensional surface $\mathcal{S}$ embedded in $\mathbb{R}^3$ and $\mathcal{M}_0$ be a damaged or destroyed region of the surface. Then the authors in [49] find a surface patch $\mathcal{M}$ which minimizes the Willmore energy over all C^1 surfaces $\mathcal{M} \cup \mathcal{S}^{ext}$ with exterior surface $\mathcal{S}^{ext} = \mathcal{S} \setminus \mathcal{M}_0$. The Willmore energy is

$$\int_{\mathcal{M}} h^2 dx$$

where h denotes the mean curvature of the surface, i.e. the sum of the principle curvatures of $\mathcal{M}$. They work with a parametrized surface and use finite elements for the discretization. Moreover, they recall that the Willmore energy is closely related to the elastica. Nitsche [114] shows that functionals with integrands depending on the principal curvatures and which are symmetric, positive definite and of polynomial order $n \geq 2$ can be written in the form $\alpha + \beta(h - h_0)^2 - \gamma K$, where K is the Gauss curvature and α, β, γ and h_0 being appropriate constants. With this expression of the integrand, the functional is an extension of the elastica in the case of surfaces (remember that the integral of the Gauss curvature over closed surfaces is a topological invariant). Then, the Willmore functional is the particular case of $\alpha = h_0 = 0$ and $\beta = 1$.

5.2.2 Joint interpolation of vector fields and gray levels

Let us go through the general ideas of the variational approach to filling-in by joint interpolation of vector fields and gray values which was introduced in [15, 16] and adapted to the problem of hole filling on surfaces in [139].

Let Q be a hyper-rectangle in $\mathbb{R}^N$ and Ω be an open bounded subset of Q with smooth boundary. Consider a function $u_0 : Q \setminus \overline{\Omega} \to \mathbb{R}$, where $\overline{\Omega}$ denotes the closure of Ω. The function u_0 is an implicit representation of the known surface. In [15, 16, 139] the authors proposed to fill-in the hole Ω using both the gray level u and the vector field of normals θ to the level sets of the image outside the hole. This permitted to design energy functionals which minimize a power of (mean) curvature and to write them in terms of the pair of variables (u, θ).

We denote by $L^p(Q)$, $1 \leq p < \infty$, the space of (measurable) functions $f : Q \to \mathbb{R}$ such that $\int_Q |f(x)|^p \, dx < \infty$. By $L^\infty(Q)$ we denote the space of bounded functions $f : Q \to \mathbb{R}$.

Let $\tilde{\Omega}$ be an open subset of Q with smooth boundary such that $\overline{\Omega} \subset\subset \tilde{\Omega}$. The band around Ω will be the set $B = \tilde{\Omega} \setminus \overline{\Omega}$. To fill-in the hole Ω they use the information of u_0 contained in B, mainly the gray level u_0 and the vector field of normals (i.e. the gradient

directions) to the level sets of u_0 in B, which are denoted by θ_0. They assume that θ_0 is a vector field with values in $\mathbb{R}^N$ satisfying $\theta_0(x) \cdot \nabla u_0(x) = |\nabla u_0(x)|$ and $|\theta_0(x)| \leq 1$. The basic goal then is to extend in a smooth way the pair (u_0, θ_0) from the band $B = \tilde{\Omega} \setminus \overline{\Omega}$ to a pair of functions (u, θ) inside Ω. For that they attempt to continue the isosurfaces of u_0 (i.e the hypersurfaces $[u_0 = \lambda]$ or, more generally, the boundaries of the level sets $[u_0 \geq \lambda]$, $\lambda \in \mathbb{R}$) in B inside Ω by taking into account the principle of good (smooth) continuation. The energy functional proposed in [15, 16, 139] was based on the following principles:

a) Constrain the solution (u, θ) to coincide with the data on the band B. The vector field θ should also satisfy $|\theta| \leq 1$ on Ω and should be related to u by $\theta \cdot \nabla u = |\nabla u|$, i.e., impose that θ is the vector field of directions of the gradient of u.

b) Impose that the vector field θ_0 in the band B is smoothly continued by θ inside Ω. Note that if θ are the directions of the normals to the level hypersurfaces of u, then $div(\theta)$ (a possible measure of smoothness of the vector field) is the mean curvature. The smooth continuation of the levels sets of u_0 inside Ω is imposed by requiring that $div\,\theta \in L^p(\tilde{\Omega})$. Indeed, if θ are the directions of the normals to the level hypersurfaces of u, then $div(\theta)$ is the mean curvature, which is a possible measure of smoothness of the vector field.

Based on these basic principles, the energy functional proposed in [15, 16, 139] to interpolate the pair (θ, u) in Ω is

$$\begin{aligned} &\text{Minimize} \int_{\tilde{\Omega}} |div(\theta)|^p (\gamma + \beta |\nabla k * u|) dx \\ &|\theta| \leq 1, \ \| u \|_\infty \leq M, \\ &|\nabla u| - \theta \cdot \nabla u = 0 \text{ in } \tilde{\Omega}, \\ &u = u_0, \ \theta = \theta_0 \text{ in } B, \end{aligned} \tag{5.2}$$

where $p > 1$, $\gamma > 0$, $\beta \geq 0$, k denotes a regularizing kernel of class C^1 such that $k(x) > 0$ a.e., and $M = \|u_0\|_{L^\infty(B)} := \sup_{x \in B} |u_0(x)|$. Note that if u is the characteristic function of a set $A \subseteq \mathbb{R}^3$ (i.e. $u(x) = \chi_A(x) = 1$ if $x \in A$ and $= 0$ otherwise) with smooth boundary and θ is an smooth extension of the unit normal to ∂A, then $\int_{\tilde{\Omega}} |div(\theta)|^p |\nabla u| dx = \int_{\partial A} |H|^p \, dS$ where $H(x)$ is the mean curvature of ∂A and dS denotes the surface area. The convolution of ∇u with the kernel k is done for technical reasons: it permits to prove the existence of a minimum for (5.2) [15, 16]. Finally, the constant γ has to be > 0, it implies an L^p bound on $div\,\theta$, and this is useful to prove that the limits

of minimizing sequences satisfy the second constraint in (5.2) [15, 16]. We refer to [16] for a detailed theoretical analysis of this formulation and its approximation by smoother functionals.

In section 5.4 we will return to this formulation and improve it with an automatic initialization method.

5.2.3 Volumetric diffusion

The algorithm proposed in [55] initially converts the surface to a volumetric representation in terms of a signed distance function d_s. They also define a weight function w_s, ranging from 0 to 1, which measures the confidence in the values of d_s: it is taken $w_s = 1$ in the known part of the surface and it decreases near the boundaries and inside the hole.

The hole filling is achieved with a process that alternates steps of blurring and compositing. The diffusion process works with two volumetric data: d^n, the signed distance at step n, and $v^n \in \{0, 1\}$, an indicator of voxels where the values d^n are valid (only valid voxels are used at each diffusion step). The initialization is:

$$(d^0, v^0) = (d_s, \chi_{w_s>0})$$

Then each time step n alternates diffusion with a low-pass filter h

$$(d^{n*}, v^n) = h * (d^{n-1}, v^{n-1})$$

and composition

$$d^n = w_s d_s + (1 - w_s) d^{n*}$$

It can be shown that this process is equivalent to heat diffusion with a source term that is added after each diffusion step.

5.3 Fixing the scenario

Assume, to fix ideas, that $\mathcal{S}$ is a smooth compact connected surface (which we assume to be embedded in $\mathbb{R}^3$), and $\mathcal{M}$ is a part of $\mathcal{S}$ which is unknown or could not be obtained during scanning (or is damaged and needs to be replaced). Let us identify $\mathcal{S}$ with its known part. Let us choose a bounding box Q in $\mathbb{R}^3$ strictly containing the surface gap $\mathcal{M}$ and part of $\mathcal{S}$ (see Figure 5.7). Let $\partial\mathcal{M}$ be the boundary of the gap. Even if

$\mathcal{M}$ is unknown, its relative boundary in $\mathcal{S}$ is known. Let $\mathcal{F}$ be a neighborhood of $\mathcal{S} \cap Q$ such that

$$\mathcal{F} = \{x \in Q : d(x, \mathcal{S} \cap Q) < \alpha d(x, \partial\mathcal{M})\}, \quad 0 < \alpha < 1.$$

We assume that $\mathcal{F} \setminus (\mathcal{S} \cap Q)$ consists of two connected components, which can be identified as the two sides of the surface $\mathcal{S}$ (see Figure 5.2). The information derived from the region $\mathcal{F}$ is considered reliable and we impose it as a constraint in our reconstruction. Let $d^{\mathcal{F}}(x)$ be the distance of a point $x \in \mathcal{F}$ to $\mathcal{S} \cap Q$. By changing the sign of $d^{\mathcal{F}}$ in one of the sides of the surface we may define the signed distance function to $\mathcal{S} \cap Q$ in $\mathcal{F}$ (take it positive inside and negative outside). We denote it by $d_s^{\mathcal{F}}(x)$, or simply, by d_s.

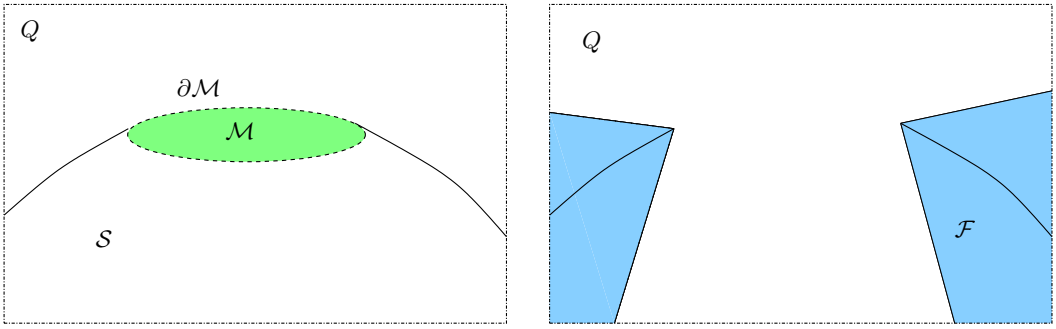


Figure 5.2: Left: part of the surface $\mathcal{S}$ (contained in Q) with the *hole* $\mathcal{M}$. Right: section of the surface and the neighborhood $\mathcal{F}$.

5.4 Joint interpolation of vector fields and gray levels and its application to surface inpainting

Let us now describe how to adapt the formulation in section 5.2.2 to inpaint (fill-in) holes (or gaps) on surfaces $\mathcal{S}$, which we assume to be embedded in $\mathbb{R}^3$.

As explained in the previous section, we can compute $d_s^{\mathcal{F}}(x)$, the signed distance of a point $x \in \mathcal{F}$ to $\mathcal{S} \cap Q$. The vector field in $\mathcal{F}$,

$$N(x) = \nabla d_s(x),$$

is an extension of the unit normal vector field on $\mathcal{S} \cap Q$ to its neighborhood $\mathcal{F}$. Again, we consider this information as reliable and it will be used as a constraint.

To adapt functional (5.2) to surface hole reconstruction we must make explicit the hole Ω and the functions (u_0, θ_0) which are the known data on a neighborhood of Ω. To define the hole we take a ball $\mathcal{B}$ (or any open set homeomorphic to a ball) such that

$\overline{\mathcal{B}} \subset\subset Q$ and containing the boundary of the gap ∂M in its interior. We define the hole Ω by removing from $\mathcal{B}$ the points of $\mathcal{F}$. We take the band $B = Q \setminus \overline{\Omega}$.

We then consider $u_0 : Q \setminus \overline{\Omega} \to \mathbb{R}$ a characteristic function, that is, a binary function taking values 0 and 1. The values $u_0(x) = 1$ and $u_0(x) = 0$ represent the points which are interior, respectively, exterior, to $\mathcal{S}$. Recall that we are assuming that $\mathcal{F} \setminus (\mathcal{S} \cap Q)$ consists of two connected components which represent the two sides of the surfaces. We then label these two sides with the values $u_0 = 1$, representing the inner part of the surface, and $u_0 = 0$, representing its outer side. By propagation, we extend this labeling to the rest of $Q \setminus \Omega$, knowing it already in $\mathcal{F}$. Notice that this can be done in a consistent way, we cannot connect two points with different labels without crossing $\mathcal{S}$ (see Figure 5.3). We call A the set of points x in $Q \setminus \Omega$ such that $u_0(x) = 1$, hence $u_0(x) = \chi_A(x)$. In this case, by minimizing (5.2), we want to reconstruct the set A inside the hole Ω knowing the set outside Ω.

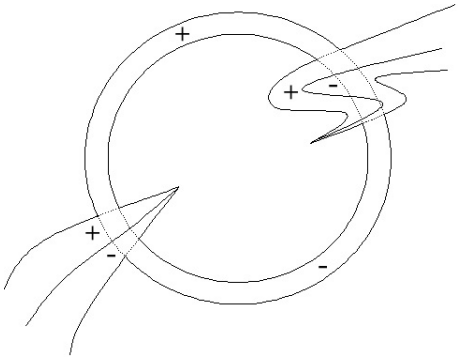


Figure 5.3: Sign assignment to the two faces of $\mathcal{S}$.

To construct θ_0 we proceed as follows. We define u in $\Omega \cup B$ as the extension of u_0 inside Ω by a geodesic propagation, and we define D_s as the signed distance to $\partial[u = 0]$ (negative in $[u = 0]$ and positive in $[u = 1]$) so that, by construction, D_s is an extension of d_s where d_s is defined in $\mathcal{F}$. We take the vector field $\theta = \nabla D_s$ in $Q := \Omega \cup B$ and $\theta_0 = \theta$ in B. Observe that $\nabla u_0 = \nu \delta_{\mathcal{S} \cap Q}$, where ν denotes the inner unit normal to $\mathcal{S}$ and $\delta_{\mathcal{S} \cap Q}$ is the Hausdorff measure on $\mathcal{S} \cap Q$. We have $\theta_0 \cdot \nabla u_0 = |\nabla u_0|$.

We constrain $u = u_0$ and $\theta = \theta_0$ in the band B. Then we minimize (5.2) by solving the gradient descent equations (5.12), (5.13) presented below, using the numerical approach described in Section 5.7.1, where the initial conditions for u and θ are defined above, so that $\theta \cdot \nabla u = |\nabla u|$ in Q.

5.5 Alternative curvature based approaches

Following the description in the introduction, we now present alternative filling-in approaches for the problem of holes in three dimensions.

5.5.1 Energy in terms of distance functions

Recall that if $\mathcal{S}'$ is a smooth manifold of class C^2, then the signed distance function D to $\mathcal{S}'$ is also of class C^2 in a neighborhood of $\mathcal{S}'$. The vector field ∇D is an extension of the unit normal to $\mathcal{S}'$ and satisfies $|\nabla D| = 1$. The operator $\Delta D(x) = \operatorname{div} \nabla D(x)$ represents the sum of the principal curvatures of the isosurface $[D = D(x)] := \{y \in Q : D(y) = D(x)\}$. When we look at this as a function in $\mathbb{R}^3$, the distance function is Lipschitz and it satisfies $|\nabla D| = 1$ in the viscosity sense. The isosurfaces may develop singularities and the only thing we can expect is that the mean curvature is a Radon measure. Indeed, recall that the mean curvature of a polyhedral surface is a Dirac's measure concentrated at the edges, and the signed distance function may have such type of singularities. We shall assume that the signed distance D to the surface $\mathcal{S}$ is such that $\Delta D \in \mathcal{M}(Q)$ where $\mathcal{M}(Q)$ denotes the space of Radon measures in Q [5]. We define

$$W(Q) = \{u \in L^1(Q) : \nabla u \in L^1(Q),\, \Delta u \in \mathcal{M}(Q)\}.$$

We propose to fill-in the three dimensional holes via the minimization of the functional

$$\operatorname{Min}\big\{D \in W(Q),\, |\nabla D| = 1,\, D = d_s \text{ in } \mathcal{F}\big\} \int_Q |\Delta D(x)|\, dx. \tag{5.3}$$

This energy integrates the mean curvature on the isosurfaces of D. Due to the singularities of the isosurfaces of D, the integral of a power of the mean curvature with an exponent $p > 1$ may be infinite. Let us observe that problem (5.3) has a minimizer as soon as the admissible set is nonempty, and we assume that this is the case.

5.5.2 Curvature diffusion and distance reconstruction

We also present studies based on diffusion of the mean curvature (see also [146] for related work based on the linear Poisson equation). For convenience, let us write $Q_{\mathcal{F}} := Q \setminus \mathcal{F}$. We propose to diffuse the mean curvature of $\mathcal{S}$ and then reconstruct the surface with the prescribed curvature, that is, we propose to solve the system of PDEs

$$\begin{aligned} \omega_t &= \Delta\omega && \text{in } [0,\infty) \times (Q \setminus \mathcal{F}) \\ u_t &= |\nabla u| \left(\operatorname{div}\left(\tfrac{\nabla u}{|\nabla u|}\right) - \omega\right) && \text{in } [0,\infty) \times (Q \setminus \mathcal{F}) \end{aligned} \tag{5.4}$$

with the following boundary conditions on $\partial\mathcal{F} \cap Q$:

$$\begin{aligned} \omega &= \Delta d_s && \text{on } \partial\mathcal{F} \cap Q, \\ \nabla u \cdot \nu &= \nabla d_s \cdot \nu && \text{on } \partial\mathcal{F} \cap Q, \end{aligned} \tag{5.5}$$

where d_s denotes the signed distance to $\mathcal{S}$. Observe that we did not write the Dirichlet boundary condition $u = d_s$ because it is not possible, in general, to impose it to the equation for u. Let us comment on the boundary conditions used on $\partial Q_{\mathcal{F}} \setminus \partial\mathcal{F}$. First of all, we observe that the ideal scenario would be to consider $Q = \mathbb{R}^N$ and solve the system of PDEs (5.4) in $[0, \infty) \times (\mathbb{R}^N \setminus \mathcal{F})$ with the boundary conditions (5.5), but this is impossible at the numerical level. For that we modify the boundary conditions on $\partial Q_{\mathcal{F}} \setminus \partial\mathcal{F}$, we do linear extrapolation of u and ω along the normal to the level sets of u.

5.6 Some interpolation operators

In [37], Caselles, Morel and Sbert studied and classified interpolation algorithms which satisfy a reasonable series of axioms in terms of the solution of a partial differential equation. Two particular examples are the Absolutely Minimizing Lipschitz Extension, denoted as AMLE in the sequel, and the Laplacian interpolation. In this section we discuss the applicability of AMLE and the Laplace equation to the problem of surface reconstruction. First of all, we recall the axiomatic approach introduced in [37].

5.6.1 Axiomatic description

Let us consider a function $u : \partial\Omega \to \mathbb{R}$ defined on the boundary of a region $\Omega \in \mathbb{R}^N$ (e.g. an image for $N = 2$ or a N-dimensional distance function). We call $\mathcal{I}$ the interpolation operator that permits to extend the function u in the region Ω, i.e. $\mathcal{I}(u, \Omega) : \Omega \to \mathbb{R}$. The authors in [37] establish a series of axioms that the interpolation operator is desired to have.

1. *Monotonicity:* Given two functions u and v, defined on $\partial\Omega$, such that $u \leq v$, then

$$\mathcal{I}(u, \Omega) \leq \mathcal{I}(v, \Omega)$$

2. *Stability:* This principle requires that the interpolation on a previously interpolated region does not alter the first result. In other words, for any $\Omega' \subset \Omega$ we have

$$\mathcal{I}(\mathcal{I}(u, \Omega)|_{\partial\Omega'}, \Omega') = \mathcal{I}(u, \Omega)|_{\Omega'}$$

3. *Regularity:* Consider that u is a second order polynomial of the form: $u(y) = \frac{1}{2}\langle A(y-x), y-x\rangle + \langle p, y-x\rangle + c$, where A is a real symmetric matrix, p is a real N-dimensional vector (different from zero) and c is a real scalar. Consider also a region $D(x,r) = \{y \in \mathbb{R}^N : ||y-x|| \leq r\}$. Then, the interpolation operator satisfies the regularity principle if there exists a continuous function F such that

$$\lim_{r\to 0} \frac{\mathcal{I}(u|_{\partial D(x,r)}, D(x,r))(x) - u(x)}{r^2/2} \to F(A,p,c,x)$$

Notice that this applies to all C^2 functions since they can be expressed under the form of a second order polynomial.

4. *Translation Invariance:* Let $x, h \in \mathbb{R}^N$, the translation operator is defined as: $\mathcal{T}_h u(x) = u(x+h)$. Then, this property is expressed as

$$\mathcal{I}(\mathcal{T}_h u, \Omega - h) = \mathcal{T}_h \mathcal{I}(u, \Omega)$$

5. *Rotation Invariance:* The rotation operator is: $\mathcal{R}u(x) = u(R^t x)$, where R is an ortogonal map in $\mathbb{R}^N$. A rotation invariant interpolator should verify:

$$\mathcal{I}(\mathcal{R}u, R\Omega) = \mathcal{R}\mathcal{I}(u, \Omega)$$

6. *Linear Contrast Change Invariance:* The interpolant should conmute with linear contrast change, i.e. changes on the values u of the type: $\lambda u + c$ where $\lambda, c \in \mathbb{R}$ (scaling and offset of the values)

$$\mathcal{I}(\lambda u + c, \Omega) = \lambda \mathcal{I}(u, \Omega) + c$$

7. *Scale (or Zoom) Invariance:* Let $u_a(x) = u(ax)$, where $a > 0$, then

$$\mathcal{I}(u_a, a^{-1}\Omega) = \mathcal{I}_a(u, \Omega)$$

Caselles, Morel and Sbert [37] show that if $\mathcal{I}$ is an interpolator operator satisfying axioms 1-7 and $u = \mathcal{I}(u_0, \Omega)$, then u, the interpolated function, is a viscosity solution of

$$\begin{array}{rcll} G\left(D^2u\left(\frac{\nabla u}{|\nabla u|}, \frac{\nabla u}{|\nabla u|}\right), D^2u\left(\frac{\nabla u}{|\nabla u|}, \frac{\nabla u^\perp}{|\nabla u|}\right), D^2u\left(\frac{\nabla u^\perp}{|\nabla u|}, \frac{\nabla u^\perp}{|\nabla u|}\right)\right) & = & 0 & \text{in } \Omega \\ u & = & u_0 & \text{in } \partial\Omega \end{array} \tag{5.6}$$

where $G(a,b,c)$ is a nondecreasing function of its variables satisfying $G(\lambda a, \lambda b, \lambda c) = \lambda G(a,b,c)\ \forall \lambda \in \mathbb{R}$. Let us clarify the notation $D^2u(v,w) = \langle D^2u(v), w\rangle = \sum_{i,j=1}^N D^2u_{ij} v_i w_j$.

Moreover, if G is differentiable at 0 then it is linear with respect to its variables and we can write the first equation in (5.6) as

$$c_1 D^2 u \left(\frac{\nabla u}{|\nabla u|}, \frac{\nabla u}{|\nabla u|} \right) + 2c_2 D^2 u \left(\frac{\nabla u}{|\nabla u|}, \frac{\nabla u^\perp}{|\nabla u|} \right) + c_3 D^2 u \left(\frac{\nabla u^\perp}{|\nabla u|}, \frac{\nabla u^\perp}{|\nabla u|} \right) = 0$$

where c_1, $c_3 \geq 0$ and $c_1 c_3 - c_2^2 \geq 0$.

Then the authors in [37] study the combination of values c_1, c_2 and c_3 that result in possible interpolation operators. More concretely, they focus on particular cases of the p-Laplacian

$$\begin{aligned} \operatorname{div}(|\nabla u|^{p-2} \nabla u) &= 0 \qquad \text{in } \Omega \\ u &= u_0 \qquad \text{in } \partial\Omega \end{aligned} \tag{5.7}$$

where $p \geq 1$. The first equation in (5.7) can be written in an equivalent form after dividing by $|\nabla u|^{p-2}$

$$(p-1) D^2 u \left(\frac{\nabla u}{|\nabla u|}, \frac{\nabla u}{|\nabla u|} \right) + D^2 u \left(\frac{\nabla u^\perp}{|\nabla u|}, \frac{\nabla u^\perp}{|\nabla u|} \right) = 0$$

We classify the different operators according to the value of p:

- $p = 1$: The resulting interpolation equation is $\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) = 0$. Indeed, this means that the curvature of the interpolated function has to be zero, which is not always desirable. In that case there is not a unique solution and the boundary conditions cannot be satisfied. Hence, this model is not a good interpolator.

- $p = 2$: This case corresponds to the Laplace equation. This equation is standard in interpolation problems but it cannot interpolate isolated points since it does not permit to respect the boundary conditions at isolated points.

- $2 < p < \infty$: In this case there exists a solution but the gradient may be unbounded, even for smooth data at the boundary.

- $p = \infty$: It yields the Absolutely Minimizing Lipschitz Extension Model (or AMLE). This equation has been studied in [10, 11, 92] and presents good properties: existence and uniqueness of viscosity solutions with boundary Lipschitz data and the gradients become bounded (more details of this model are given in subsection 5.6.3).

5.6.2 The Laplace interpolation

The Laplacian interpolation is based on solving the PDE

$$-\Delta u = 0 \qquad \text{in } Q_{\mathcal{F}}, \tag{5.8}$$

with specified boundary data on $\partial Q_{\mathcal{F}}$. Indeed, boundary data is only known in $\partial\mathcal{F} \cap Q$ where we should impose that $u = d_s$. Thus, a reasonable assumption would be to assume that

$$\frac{\partial u}{\partial \nu} = 0 \qquad \text{in } \partial Q_{\mathcal{F}} \setminus \partial\mathcal{F} \tag{5.9}$$

where ν denotes the outer unit normal to $\partial Q_{\mathcal{F}} \setminus \partial\mathcal{F}$. In some sense, from the theoretical point of view, the lack of knowledge of boundary conditions for u in $\partial Q_{\mathcal{F}} \setminus \partial\mathcal{F}$ excludes the possibility of using (5.8) to reconstruct the surface $\mathcal{S} \cap Q$ (which is defined as $\partial[u > 0]$). In spite of this, we tested using (5.8) with boundary condition (5.9) and the results are also presented in Section 5.8. We should of course mention that this approach is closely related to the work in [55], based on linear diffusion.

5.6.3 The Absolute Minimizing Lipschitz Extension interpolation

The AMLE interpolation ([10, 11]) is based on solving the PDE

$$\langle D^2 u\,(\nabla u)\,, \nabla u\rangle = 0 \quad \text{in } Q_{\mathcal{F}}. \tag{5.10}$$

with boundary data on $\partial Q_{\mathcal{F}}$. Here ∇u and $D^2 u$ denote the gradient and the Hessian matrix of u, respectively, so that in Cartesian coordinates,

$$\langle D^2 u\,(\nabla u)\,, \nabla u\rangle = \sum_{i,j=1}^{N} \frac{\partial^2 u}{\partial x_i \partial x_j}\frac{\partial u}{\partial x_i}\frac{\partial u}{\partial x_j}.$$

This equation can be solved with general domains and boundary data, in particular the data can be given in a finite number of surfaces, curves and/or points. Indeed, we may assume that the boundary data $\varphi \in Lip_{\partial}(Q_{\mathcal{F}})$ where

$$Lip_{\partial}(Q_{\mathcal{F}}) = \left\{ g \in C(\partial Q_{\mathcal{F}}) : |||g||| = \sup_{x,y\in\partial Q_{\mathcal{F}}} \frac{|g(x) - g(y)|}{d_{\partial Q}(x,y)} < \infty \right\},$$

and $d_{Q_{\mathcal{F}}}(x, y)$ is the geodesic distance between x and y in $Q_{\mathcal{F}}$, i.e., the minimal length of all possible paths joining x and y and contained in $Q_{\mathcal{F}}$ [92].

Let us recall that, if X is an open set or a smooth manifold in $\mathbb{R}^N$, $W^{1,\infty}(X)$ (resp. $W^{1,p}(X)$) denotes the space of functions $u \in L^\infty(X)$ (resp. $u \in L^p(X)$) such that $\nabla u \in L^\infty(X)$ (resp. $\nabla u \in L^p(X)$). By $W_0^{1,\infty}(X)$, resp. $W_0^{1,p}(X)$, we denote the closure in $W^{1,\infty}(X)$, resp. in $W^{1,p}(X)$, of the smooth functions with compact support in X.

Existence and uniqueness of viscosity solutions for the AMLE model (5.10) with boundary data $\varphi \in Lip_{\partial}(Q_{\mathcal{F}})$ was proved by Jensen [92]. Moreover, he proved that the

viscosity solution of (5.10) is an absolutely minimizing Lipschitz extension of φ, i.e., $u \in W^{1,\infty}(Q_{\mathcal{F}}) \cap C(\overline{Q_{\mathcal{F}}})$ and satisfies

$$\|\nabla u\|_{L^\infty(Q';R^N)} \leq \|\nabla w\|_{L^\infty(Q';R^N)} \tag{5.11}$$

for all $Q' \subseteq Q_{\mathcal{F}}$ and w such that $u - w \in W_0^{1,\infty}(Q')$. Let us add that the AMLE model was introduced by Aronsson in [10, 11] as the Euler-Lagrange equation of the variational problem (5.11) (which can be interpreted as the limit as $p \to \infty$ of the variational problems $\|\nabla u\|_{L^p(Q';R^N)} \leq \|\nabla w\|_{L^p(Q';R^N)}$ for all $Q' \subseteq Q_{\mathcal{F}}$ and w such that $u - w \in W_0^{1,p}(Q')$ [17, 92]). The above results were extended in [92] to the case of continuous boundary data $\varphi \in C(\partial Q_{\mathcal{F}})$, and Jensen proved that in that case, the $AMLE$ is locally Lipschitz continuous in $Q_{\mathcal{F}}$ [92].

The same remarks we made for the Laplace equation (5.8) can be done here, that is, boundary data is only known in $\partial\mathcal{F} \cap Q$ where we should impose that $u = d_s$ (by the results in [94] there exist absolutely minimizing Lipschitz extensions of $d_s|_{\partial\mathcal{F}\cap Q}$ and satisfy (5.10), but there is no uniqueness result for them). From the theoretical point of view, the lack of knowledge of boundary conditions excludes the possibility of using (5.10) to reconstruct the surface $\mathcal{S} \cap Q$ (which is defined as $\partial[u > 0]$). In spite of this, we experiment with it in Section 5.8. Another numerical possibility would be to linearly extrapolate the values of u along the direction ∇u.

5.7 Numerical considerations

We now present some basic concepts related to the numerical implementation of the different filling-in models described above.

5.7.1 Joint interpolation of vector fields and gray levels

To minimize the functional (5.2) we use the steepest descent method. If we denote the energy by $\tilde{E}(\theta, u)$, the steepest descent equations are

$$\theta_t = -\nabla_\theta \tilde{E}(\theta, u) \tag{5.12}$$

and

$$u_t = -\nabla_u \tilde{E}(\theta, u) \tag{5.13}$$

in $(0,\infty)\times\tilde{\Omega}$, supplemented with the corresponding boundary data and initial conditions. The constraints on (θ, u) can be incorporated either by penalization or by projecting onto

them after each time step. We tested both methods in an implicit (also in an explicit) in time discretization of (5.12), (5.13). Let us explain in some detail the implicit in time implementation of (5.12), (5.13) with the constraint $\theta \cdot \nabla u = |\nabla u|$ incorporated by penalization. Thus we consider

$$\tilde{E}(u,\theta) = \int_{\tilde{\Omega}} |div(\theta)|^p(\gamma + \beta|\nabla k * u|)dx + \eta \int_{\tilde{\Omega}} (|\nabla u| - \theta \cdot \nabla u) \tag{5.14}$$

which corresponds to the energy (5.2) plus a penalization term for the constraint that $\theta \cdot \nabla u = |\nabla u|$, with $\eta > 0$. To simplify our notation, let us write $g(\theta) := \beta|div(\theta)|^p$, $h(u) := \gamma + \beta|\nabla k * u|$. Then

$$\nabla_\theta \tilde{E}(\theta, u) = -p\nabla\big[h(u)|div(\theta)|^{p-2}div(\theta)\big] - \eta\nabla u = 0 \tag{5.15}$$

and

$$\nabla_u \tilde{E}(\theta, u) = -div\left(k * \left(g(\theta)\frac{\nabla k * u}{|\nabla k * u|}\right)\right) - \eta\, div\left(\frac{\nabla u}{|\nabla u|}\right) + \eta\, div\,\theta = 0, \tag{5.16}$$

To solve equations (5.12) and (5.13), we use an implicit discretization in time. To be precise, we write

$$\nabla_\theta \tilde{E}(\theta, \theta', u, v) = -p\nabla\big[h(u)(\epsilon + |div(\theta')|^{p-2})div(\theta)\big] - \eta\nabla u = 0 \tag{5.17}$$

and

$$\nabla_u \tilde{E}(\theta, \theta', u, v) = -div\left(k * \left(g(\theta)\frac{\nabla k * u}{\sqrt{\epsilon + |\nabla k * v|^2}}\right)\right) - \eta\, div\left(\frac{\nabla u}{\sqrt{\epsilon + |\nabla v|^2}}\right) + \eta\, div(\theta) = 0. \tag{5.18}$$

Then, we use the discretization in time given by

$$\theta^{n+1} - \theta^n = -\Delta t \nabla_\theta \tilde{E}(\theta^{n+1}, \theta^n, u^n, u^n), \tag{5.19}$$

and

$$u^{n+1} - u^n = -\Delta t \nabla_u \tilde{E}(\theta^{n+1}, \theta^{n+1}, u^{n+1}, u^n). \tag{5.20}$$

Finally, we make the change of variables $\xi^{n+1} := \theta^{n+1} - \theta^n$, $v^{n+1} := u^{n+1} - u^n$ and we have

$$\xi^{n+1} = -\Delta t \nabla_\theta \tilde{E}(\xi^{n+1} + \theta^n, \theta^n, u^n, u^n), \tag{5.21}$$

$$v^{n+1} = -\Delta t \nabla_u \tilde{E}(\theta^{n+1}, \theta^{n+1}, v^{n+1} + u^n, u^n). \tag{5.22}$$

In practice we solve equations (5.21),(5.22) in $\tilde{\Omega}$ with the boundary conditions

$$u^{n+1} = u_0 \quad \text{and} \quad \theta^{n+1} \cdot \nu^{\tilde{\Omega}} = \theta_0 \cdot \nu^{\tilde{\Omega}} \quad \text{on } \partial\tilde{\Omega}$$

and with $u^n = u_0$ and $\theta^n = \theta_0$ on the band B. Then we redefine $u^{n+1} = u_0$ and $\theta^{n+1} = \theta_0$ in B.

Now, since $\theta^n \cdot \nu^{\tilde{\Omega}}|_{\partial\tilde{\Omega}} = \theta^{n+1} \cdot \nu^{\tilde{\Omega}}|_{\partial\tilde{\Omega}}$ and $u^n|_{\partial\tilde{\Omega}} = u^{n+1}|_{\partial\tilde{\Omega}}$, the normal component of ξ^{n+1} and the value of v^{n+1} are zero at the boundary, and we may use a conjugate gradient method to solve (5.21) and (5.22). The constraint $|\theta| \leq 1$ is incorporated by renormalizing θ^n (when $|\theta^n| > 1$) after each time step. The constraints on $\|u\|_\infty$ can be also introduced after each time step. In spite of the penalization term, the relationship $\theta \cdot \nabla u = |\nabla u|$ is (numerically) lost and we reinforce it after a certain number of time steps.

We can also set $\eta = 0$ and incorporate the constraint that $|\nabla u| = \theta \cdot \nabla u$ by projecting onto it after each time step. We also tested this both in a time implicit and explicit discretization of equations (5.12), (5.13). After each time step of θ and u we redefine

$$\theta(i,j) = \frac{\theta(i,j) + \alpha \nabla u(i,j)}{max(1, |\theta(i,j) + \alpha \nabla u(i,j)|)}$$

for some $\alpha > 0$. As it has been shown in [90] this is a good way of imposing that $|\theta| \leq 1$ and $\theta \cdot \nabla u = |\nabla u|$. We have found quite similar results using both described methods for imposing the constraint.

In our experiments, we take k a Gaussian kernel with small variance, say one or two pixels. In practice, one can also dismiss the kernel k. The initial conditions for u and θ are taken as we explained at the end of Section 5.4 so that $\theta \cdot \nabla u = |\nabla u|$ in $Q := \Omega \cup B$. We could also construct an initialization (u, θ) using the solution Laplace equation, or AMLE (see Section 5.6).

5.7.2 Curvature based approaches

We use the steepest descent method to minimize the functional (5.3), then we must solve:

$$D_t = -\Delta \left(\frac{\Delta D}{|\Delta D|} \right) \tag{5.23}$$

In order to satisfy the constraint $|\nabla D| = 1$ we make use of the PDE that computes the signed distance function [117]:

$$D_t = -sign(D)(|\nabla D| - 1) \tag{5.24}$$

Then, as a numerical approach to minimize (5.3) we combine (5.23) and (5.24) at each time step. To go from D^n to D^{n+1} we first solve

$$D^* = D^n - \Delta t \Delta \left(\frac{\Delta D^n}{|\Delta D^n| + \epsilon} \right)$$

using centered differences and then solve

$$D^{n+1} = D^* - \Delta t sign(D^*)(|\nabla D^*| - 1)$$

using an upwind scheme for the gradient magnitude [117, 118]. As boundary conditions, we use $D^n = D^* = d_s$ on $\partial\mathcal{F} \cap Q$ and linear extrapolation of D^n and D^* along its gradient direction on $\partial Q_{\mathcal{F}} \setminus \partial\mathcal{F}$.

Even if not fully theoretically justified, unless we work in a small neighborhood of the surface $D = 0$, a similar scheme may be used to minimize the functional

$$\text{Min}_{\{D:\, |\nabla D| = 1,\, \Delta D \in L^2(Q),\, D = d_s \text{ in } \mathcal{F}\}} \int_Q |\Delta D(x)|^2 \, dx. \tag{5.25}$$

In order to solve the system (5.4), firstly we solve $\omega_t = \Delta\omega$ by an explicit Euler scheme and centered differences in space. Then, we reconstruct the distance function solving $u_t = |\nabla u| \left(\text{div}\left(\frac{\nabla u}{|\nabla u|}\right) - \omega\right)$. We also use an explicit Euler scheme in time. As regards the space discretization we use forward differences for the gradient, backward differences for the divergence and an upwind scheme for the modulus of the gradient multiplying $\text{div}\left(\frac{\nabla u}{|\nabla u|}\right) - \omega$ [118].

5.7.3 The Laplace equation and the AMLE

The Laplace equation (5.8) is solved by computing the steady state of equation $u_t = \Delta u$. This PDE is discretized with an implicit (or explicit) Euler scheme in time and centered differences in space. In case of the implicit Euler scheme, the corresponding linear system is solved using the conjugate gradient method. As boundary conditions we use $u = d_s$ on $\partial\mathcal{F} \cap Q$ and linear extrapolation of u along its gradient direction on $\partial Q_{\mathcal{F}} \setminus \partial\mathcal{F}$ (one can also use the Neumann boundary conditions (5.9)). The same boundary conditions will be used for the AMLE equation whose numerical scheme we describe now.

The AMLE equation (5.10) is also solved computing the steady state of its associated evolution problem. We use an implicit Euler scheme and centered differences. Then we use the Newton algorithm with a relaxation parameter. Thus, at each time step we compute:

$$u^{n+1} = u^n - w\frac{(1+2\Delta t)|\nabla^c u^*|^2 u^* - u^n|\nabla^c u^*|^2 - \Delta t \sum_{i,j=1}^3 \bar{u}^*_{x_i x_j} \delta^c_{x_i} u^* \delta^c_{x_j} u^*}{(1+2\Delta t)|\nabla^c u^*|^2 + \epsilon}$$

where the relaxation parameter is $0 < w < 2$ (we use, in practice, $w = 1.5$). We use the superscript c to denote centered differences in spatial derivatives, we define thus:

$\nabla^c u = (\delta^c_{x_1} u, \delta^c_{x_2} u, \delta^c_{x_3} u)$, $\bar{u}_{x_i x_i} = \delta^c_{x_i x_i} u - 2u$ and $\bar{u}_{x_i x_j} = \delta^c_{x_i x_j} u$ for $i \neq j$. Finally, if we denote $<<$ as the lexicographical order for indexes on R^3 then

$$u^*(p,q,r) = \begin{cases} u^{n+1}(p,q,r) & \text{if } (i,j,k) << (p,q,r) \\ u^n(p,q,r) & \text{else} \end{cases}$$

when computing the value at location (i,j,k).

5.8 Experimental results

We now show experimental results illustrating the filling-in techniques here proposed.

5.8.1 Simple geometric objects

First, we display several experiments of geometric objects done with the different methods discussed above: the joint interpolation of vector fields and gray levels (abbreviated JIVFGL), minimization of the absolute value of the Laplacian of the distance function (and also the case of power 2), curvature diffusion, the AMLE, and Laplace equations. The images in our experiments have been rendered using the AMIRA Visualization and Modeling System [8].

Figure 5.4 shows a pyramid with a hole of size 61×61 in the two horizontal dimensions and 6 in the vertical dimension, and its corresponding reconstructions. Figure 5.4.a shows the gap in the pyramid. Figure 5.4.b shows its reconstruction by joint interpolation of vector fields and gray levels (JIVFGL), i.e., functional (5.2) with $N = 3$. Figure 5.4.c shows the result obtained minimizing the absolute value of the Laplacian of the distance function, i.e., (5.3). Figure 5.4.d shows the result obtained minimizing the square of the Laplacian of the distance function, i.e., (5.25). Figure 5.4.e shows the result obtained with curvature diffusion (5.4). Finally, Figures 5.4.f and g display the results obtained solving the AMLE and Laplace equations in 3D, respectively. Note how the reconstructions obtained with the model JIVFGL and the square of the Laplacian for example manage to fill-in a relatively large hole. The others do a decent work that can certainly be used as a very good initial condition for the best models to refine.

Figure 5.5 displays a torus with a hole (it has 6 voxels in the inner circle arc, 15 in the outer circle arc, its width is 20, and its height is 18 voxels) and its corresponding reconstructions. Figure 5.5.a shows the gap in the torus. Figure 5.5.b shows its reconstruction by joint interpolation of vector fields and gray levels (JIVFGL). Figure

5.5.c displays the result obtained minimizing the absolute value of the Laplacian of the distance function, i.e., (5.3). Figure 5.5.d displays the result obtained minimizing the square of the Laplacian of the distance function, i.e., (5.25). Figures 5.5.e and f display the results obtained solving the AMLE and Laplace equations in 3D. Note once again the very good reconstruction with the JIVFGL model, for a significant hole, and the good initial conditions (at least) of the others.

5.8.2 Experiments with Michelangelo's David

For this real data coming from the Stanford Michelangelo's project [103], with the purpose of adapting it to our algorithm, the data, originally given as a triangulated surface, was converted to an implicit representation in a regularly spaced 3D grid. The result is visualized again as a triangulated surface. Figure 5.6 shows a rendering of a scanned version of Michelangelo's David which has several holes.

Figures 5.7.a, 5.7.b, 5.7.c show some particular holes with a bounding box isolating them. Figures 5.7.d, 5.7.e, 5.7.f show the triangulated surface (the data) around the hole. The reconstructed surface using JIVFGL is shown in Figures 5.7.g, 5.7.h, 5.7.i. The reconstructed surfaces look very natural.

Figure 5.8.a shows the hole in David's left hand. Figure 5.8.b,c,d,e,f show the corresponding results obtained minimizing the absolute value of the Laplacian, the square of the Laplacian, diffusion of curvature, the AMLE, and Laplace equations, respectively. All reconstructions look again very natural, while we can observe that for example the curvature diffusion is less smooth.

The hole in David's left hand is shown in figure 5.9.a. Inpainted surfaces obtained minimizing the absolute value of the Laplacian, the square of the Laplacian, diffusion of curvature, the AMLE, and Laplace equations are shown in figures 5.9.b,c,d,e,f respectively.

5.9 Conclusions

In this chapter we have introduced geometrical approaches to fill-in surface holes. The idea, inspired by [55] and [15, 16, 139], is to represent the surface of interest by means of a function u, as an upper level set $[u > 0]$, and minimize an energy functional which integrates a power of the mean curvature of the level sets of u. Then we use a gradient descent method and, thus, we run a system of coupled geometric partial differential equations that permit to geometrically continue the surface into the hole.

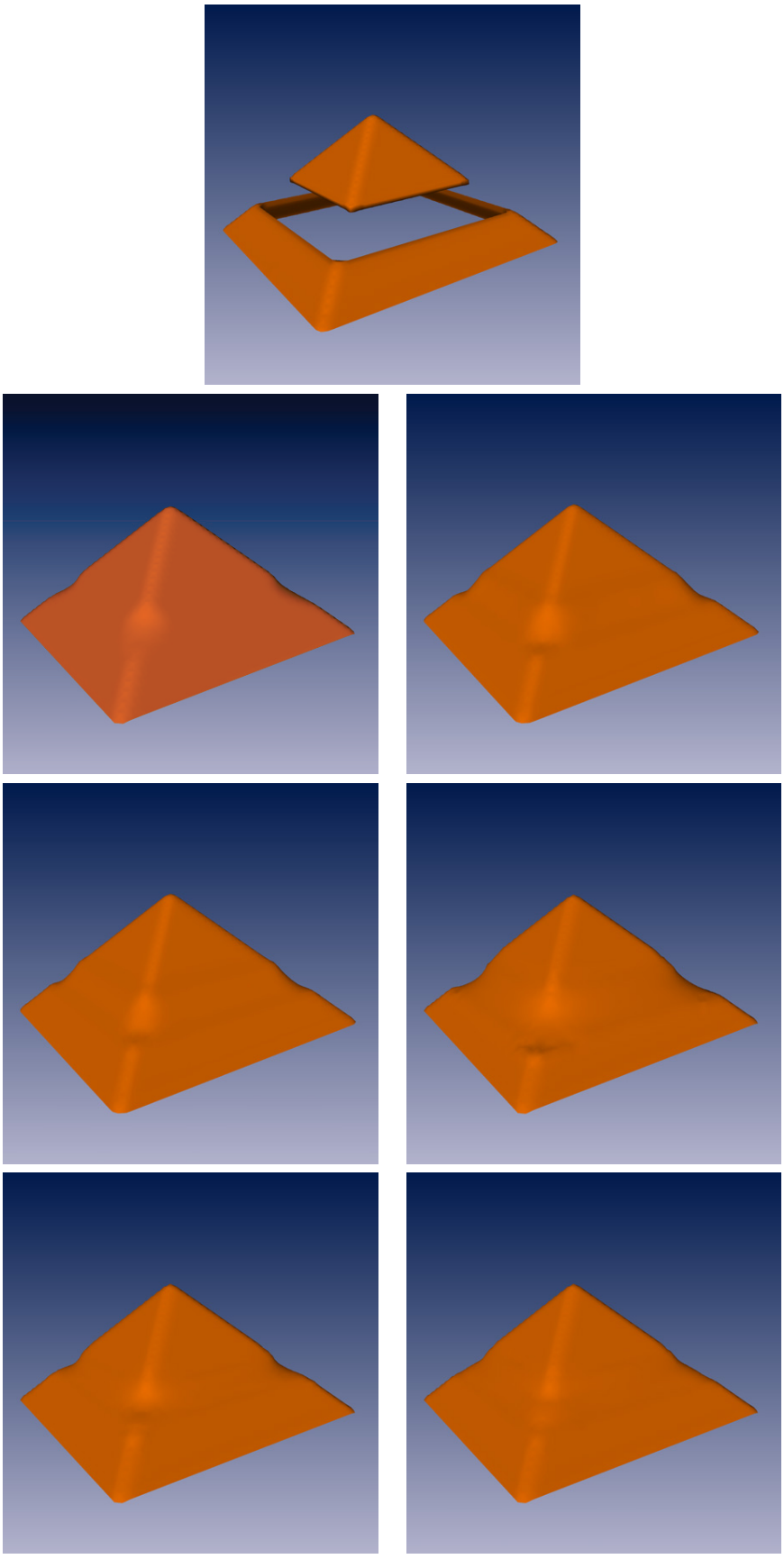


Figure 5.4: From top to bottom and left to right: a) Pyramid with a gap, b) Reconstruction using JIVFGL, c) Reconstruction obtained minimizing the absolute value of the Laplacian, d) Reconstruction obtained minimizing the square of the Laplacian e) Using curvature diffusion, f) Using the AMLE, g) Using Laplace equation.

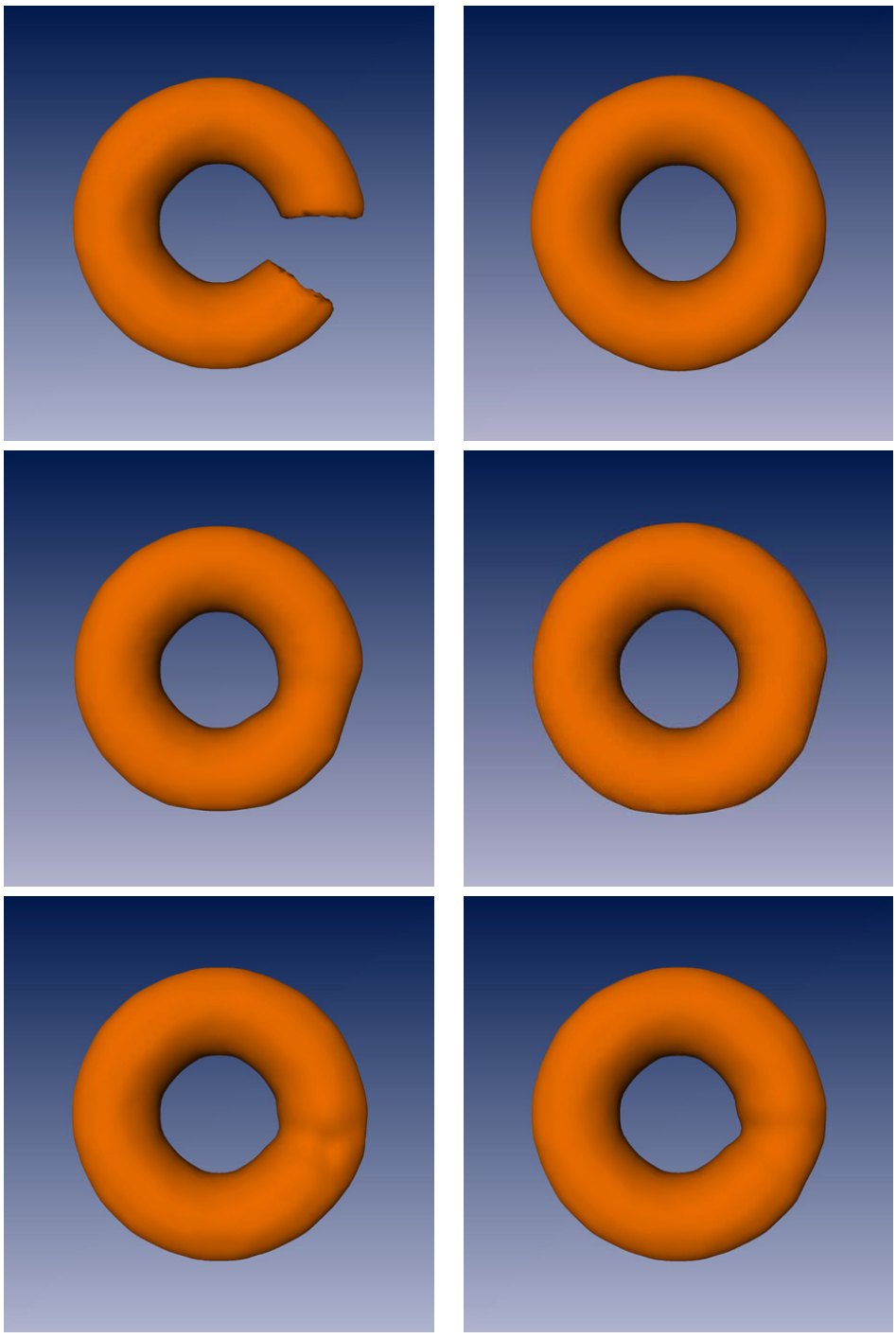


Figure 5.5: From top to bottom and left to right: a) Torus with a gap, b) Reconstruction using JIVFGL, c) Reconstruction obtained minimizing the absolute value of the Laplacian, d) Reconstruction obtained minimizing the square of the Laplacian, e) Using the AMLE, f) Using Laplace equation.

Figure 5.6: Scanned version of Michelangelo's David (data from the Stanford Michelangelo's project [103]).

We have also discussed other curvature based hole surface reconstruction models, one of them based on a variational model which integrates the Laplacian of a distance function, the other is heuristic and is based on the diffusion of a function which represents the mean curvature of level sets of an underlying implicit function. In all these cases, we have showed reconstruction of surface holes both for synthetic and real data.

Finally, we have also shown some other simpler methods based on the Laplace equation and the so-called AMLE model which reconstructs a function which is distance-like near the known part of the surface and whose zero level set can be interpreted as the reconstructed surface. If our interest is just to find a smooth reconstruction, this approach may be sufficient. If one wants a reconstruction which is based on minimizing mean curvature, it can just serve as an initialization stage.

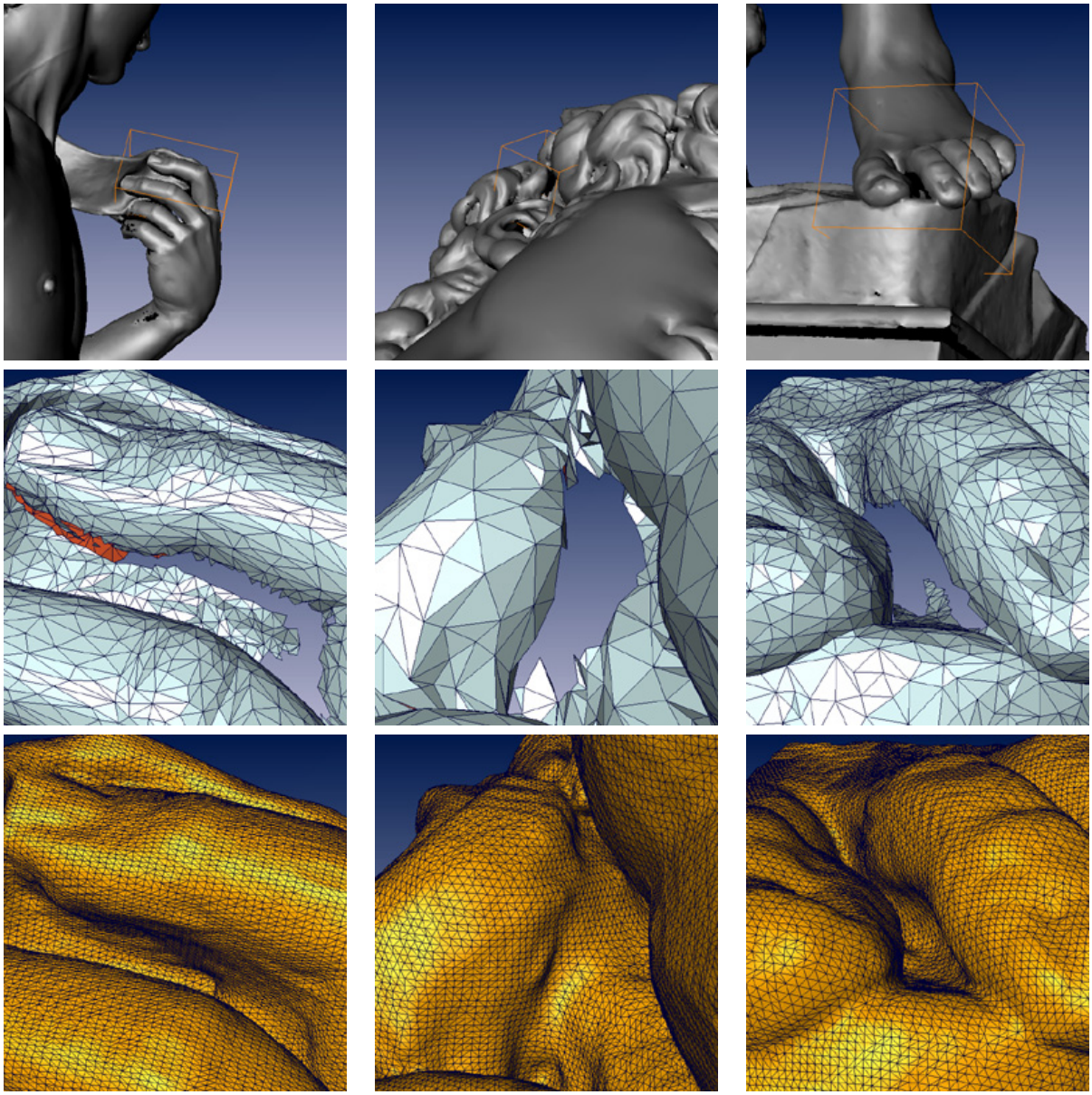


Figure 5.7: The results in this Figure have been obtained using JIVFGL. From top to bottom and left to right: a) David's left hand, b) A detail of its hair, c) David's left foot, d) A zoomed detail of a) showing the triangulated surface with the hole, e) A zoomed detail of b) showing the triangulated surface with the hole, f) A detail of the fingers with a hole, g) The reconstruction of the hole in d) displayed as a triangulated surface, h) The reconstruction of the hole in e) displayed as a triangulated surface, i) The reconstruction of f) displayed as a triangulated surface.

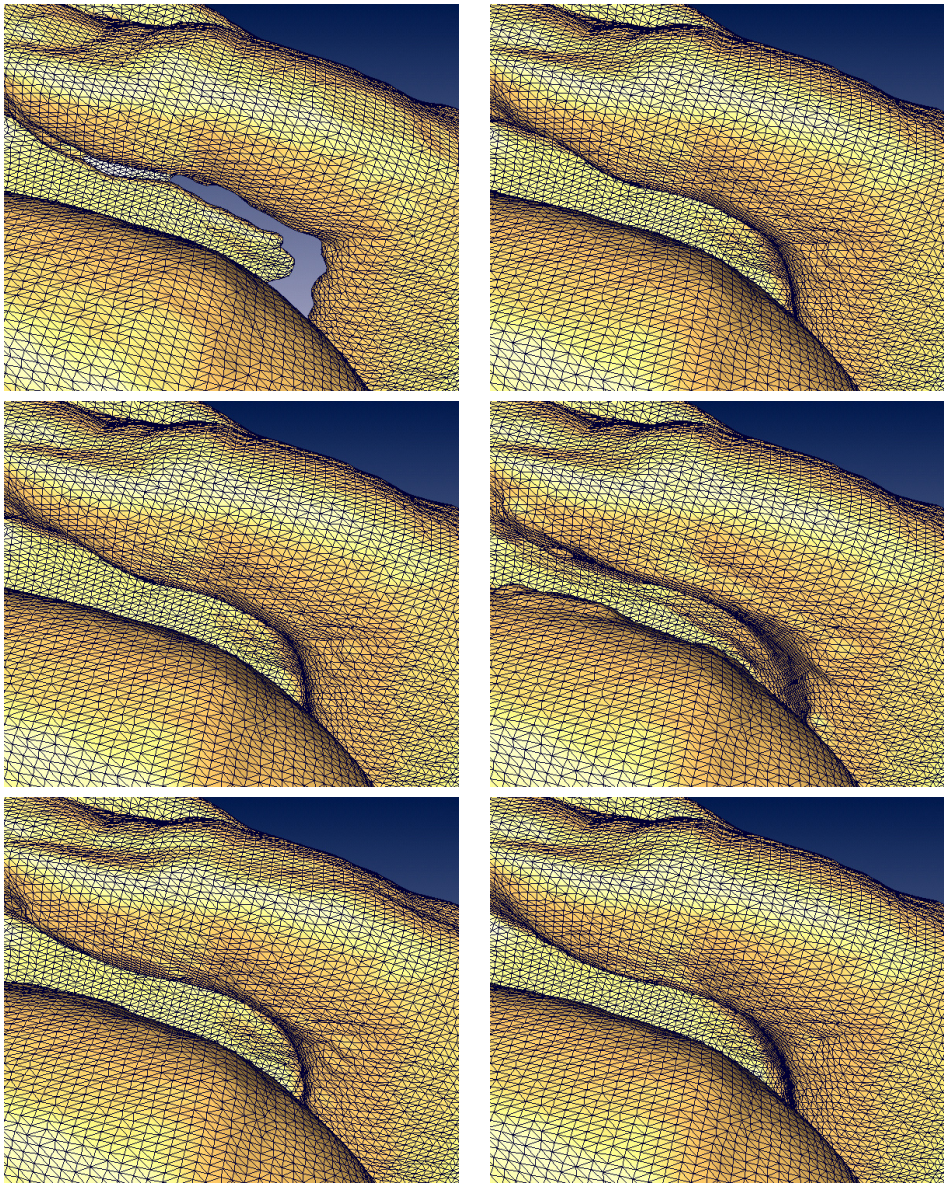


Figure 5.8: From top to bottom and left to right: a) The hole in David's left hand, b) Reconstruction obtained minimizing the absolute value of the Laplacian, c) Reconstruction obtained minimizing the square of the Laplacian, d) Reconstruction using curvature diffusion, e) Using the AMLE, f) Using Laplace equation.

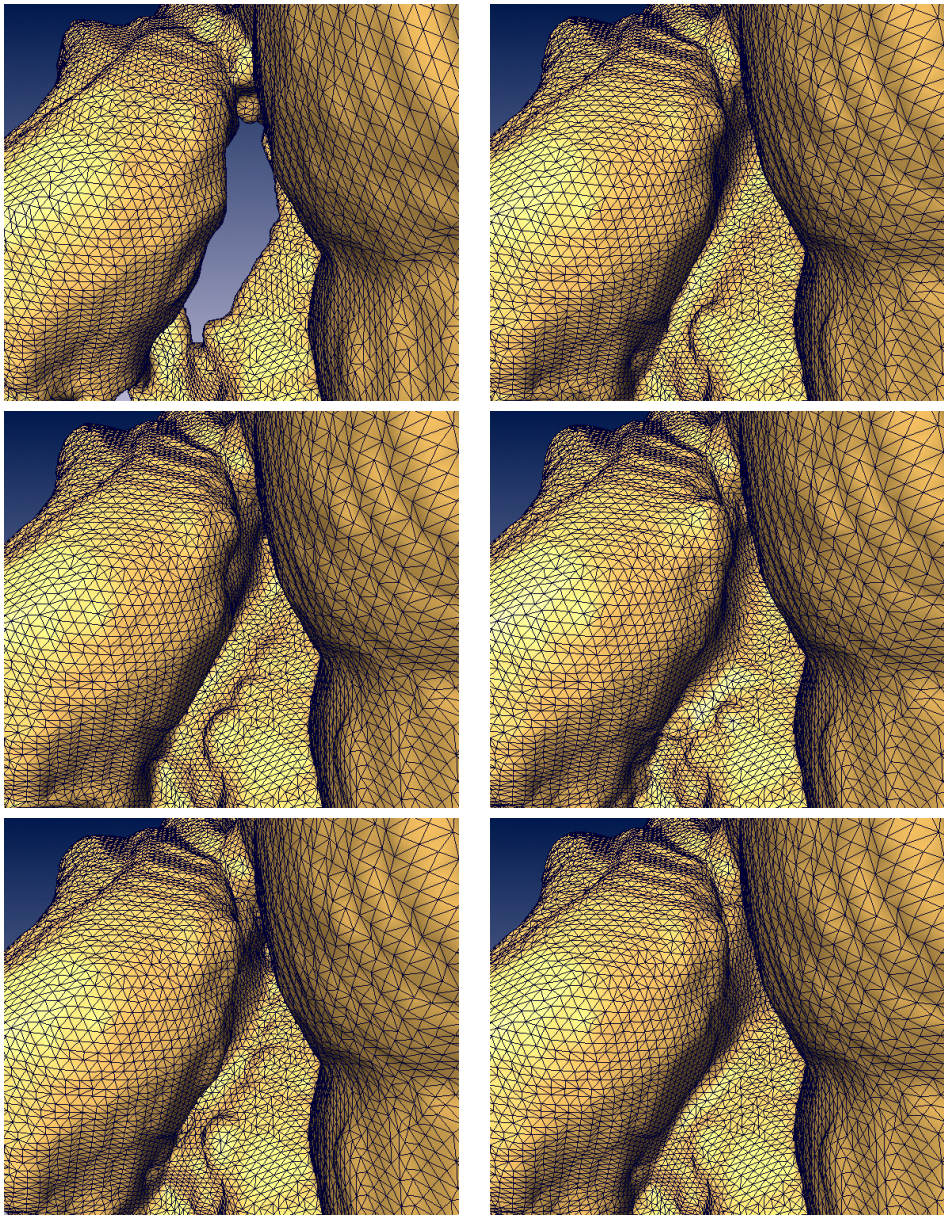


Figure 5.9: From top to bottom and left to right: a) The hole in David's hair. b) Reconstruction obtained minimizing the absolute value of the Laplacian c) Reconstruction obtained minimizing the square of the Laplacian, d) Reconstruction using curvature diffusion, e) Using the AMLE, f) Using Laplace equation.

Part IV

Restoration of irregularly sampled images

Chapter 6

Total Variation based restoration

ABSTRACT

This chapter reviews some important aspects of the irregular sampling literature. We also propose a general algorithm to solve a problem of image restoration which considers several different aspects of it, namely: irregular sampling, denoising, deconvolution, and also antialiasing and zoom. Our algorithm is based on an extension of an algorithm proposed by A. Chambolle [40] in the case of image denoising using total variation, combined with irregular to regular sampling algorithms proposed by K. Gröchenig and his coauthors [67, 80]. Finally we present some experimental results and we compare them with those obtained with the algorithm proposed by K. Gröchenig et al.

6.1 Introduction

If we want to model a general digital image acquisition system we have to take into account the following aspects:

- **Noise:** it is undesired information that contaminates the image and it comes from a variety of sources. The Gaussian distribution is most often used to model natural noise processes, such as those occurring from electronic devices in the acquisition system. We call n a Gaussian noise with standard deviation σ.

- **Modulation Transfer Function of the system (MTF):** it describes how frequency information is degraded by the image acquisition system. Sensors and optics may not be perfect and they blur the acquired image (see appendix C for more details of the MTF in satellite imaging). We denote h a blurring kernel whose Fourier spectrum, or Modulated Transfer Function, is essentially supported in $[-1/2, 1/2]^2$.

- Sampling process: when producing a digital image (directly obtained from a digital acquisition system or after converting an analogic image) several samples are taken at specific locations. Let us call $\Lambda = \{\lambda_k\}_k$ the set of sampling locations, it may be regular (equidistant samples) or irregular (see section 6.2 for a general overview of irregular sampling).

Then, a suitable digital image formation model would be:

$$g = \Delta_\Lambda \cdot (u * h) + n \tag{6.1}$$

where u is the ideal undistorted image and g is the acquired image. Sampling may be written as a multiplication by the Dirac comb

$$\Delta_\Lambda = \sum_{\lambda_k \in \Lambda} \delta(\cdot - \lambda_k).$$

We shall concentrate in the particular case of perturbed sampling and we may assume that the set Λ takes the particular form

$$\Lambda = \mathbb{Z}^2 + \varepsilon(\mathbb{Z}^2) \tag{6.2}$$

where $\varepsilon : \mathbb{R}^2 \to \mathbb{R}^2$ is a "smooth and small" perturbation function in the sense that $\mathrm{supp}(\hat{\varepsilon}) \subseteq [-\frac{1}{2T_\varepsilon}, \frac{1}{2T_\varepsilon}]^2$ for some period T_ε of the maximum vibration frequency such that $T_\varepsilon > 1$ and that the amplitude A which can be measured as the max or the standard deviation of each component of the vector $\varepsilon(x)$ is small with respect to 1 pixel (see section 6.2 for a modelization (6.5) of this perturbation).

In the more general setting where sampling is not necessarily perturbed, the well-posedness of the sampling set is measured by its density. Let us recall that a sampling set $\Lambda \subseteq \Omega$ is said to be γ-dense if

$$\bigcup_{\lambda_k \in \Lambda} D_\gamma(\lambda_k) = \Omega \tag{6.3}$$

i.e. if the image domain Ω can be covered by disks $D_\gamma(\lambda_k)$ of radius γ that are centered on the sampling points λ_k. We shall also call the value $\upsilon = 2\gamma$, *"maximal gap"* of the sampling set Λ whenever γ is the minimal value such that Λ is γ-dense (almost equivalently the maximal gap υ is the maximal diameter of the Voronoï cells associated to Λ).

Our problem consists of recovering as much as possible of u, from our knowledge of the sampling geometry Λ, the blurring kernel h, the statistics of the noise n, and the irregular samples g.

In [2] Almansa studied two subproblems of this general problem. First, sampling was assumed to be regular, but the MTF $\hat{h}$ was not restricted to $[-1/2, 1/2]^2$, and the problem was to study how to best circumvent the aliasing artifacts. As the author concluded (see also [3]), an optimal spectral region R similar to $[-1/2, 1/2]^2$ could be found (depending on h, and a statistical model of the image u and the noise n) where sampled coefficients contain a minimal amount of noise and aliasing. Then, following [106], Almansa proposed to extrapolate the restored information on R to a region R' containing $[-1/2, 1/2]^2$ by minimizing the total variation.

As a second case, sampling was assumed to be perturbed or irregular, but the MTF was assumed to be an ideal window: $\hat{h} = \chi_{[-1/2,1/2]^2}$. Most of the numerical algorithms that were analyzed worked relatively well only within a low-frequency spectral region $R \subseteq [-\alpha/2, \alpha/2]^2$ where $\alpha \approx 1 - 1/T_\varepsilon$ for an irregular sampling set such that $T_\varepsilon > 1$ when the sampling set is perturbed. When attempting to recover $\hat{u}$ on the high frequency band $[-1/2, 1/2]^2 \backslash R$ serious theoretical and numerical problems were faced, and, actually, restoration errors where most important in that band.

Motivated by the previous discussion, we consider the following variational model for restoring u based on the full image formation model in equation (6.1):

$$\min_u \frac{1}{2}\|w \cdot \Delta_\Lambda \cdot (p * h * u) - g\|^2 + \lambda J(u) \tag{6.4}$$

where $J(u) = \int |\nabla u|$ is the total variation of u, w is a weighting function acting as a preconditionner (typically is the Voronoï area of the sampling set Λ, see Section 6.2.2), and p is a spectral projector (e.g. $\hat{p} = \chi_R$ or a prolate function, see section 6.6) on a low-band region R which depends both on the MTF and the irregularity of the sampling set. The main idea behind the use of the projector is to use linear restoration methods within the well-posed region R and extrapolate this result -via total variation minimization- to a region R' containing $[-1/2, 1/2]^2$. In this way we deal, at the same time, with the problem of antialiasing and zoom.

The use of total variation for image restoration problems was originally proposed by Rudin, Osher and Fatemi in [99]. Functions with finite total variation, usually called bounded variation functions [98, 65, 149], are a reasonable functional model for many problems in image processing, in particular, for image restoration problems [99]. Typically, functions of bounded variation in the plane have discontinuities along rectifiable curves, being continuous in some sense (in the measure theoretic sense) away from discontinuities. These discontinuities could be identified with edges. The ability of total variation regularization to recover edges is one of the main features which advocates for

the use of this model (its ability to describe textures is less clear, some textures can be recovered, but up to a certain scale of oscillation). See appendix D for the definition of functions of bounded variation and some references.

Many numerical algorithms have been proposed to minimize the total variation subject to constraints (we refer to [9] for a survey). The particular case of total variation denoising has been widely studied numerically [140, 41, 44, 46, 40]. Recently, a very interesting algorithm for total variation denoising which is guaranteed to converge to the exact solution was proposed by Chambolle [40] (in this case the image formation model (6.1) simplifies to $g = u + n$). In this work we shall extend his result in such a way that it can also be applied within a very general image restoration framework.

The main focus of the present work is to propose a full image restoration model. For that, we study the case of irregular to regular bandlimited sampling, and observe that the ideas of the ACT algorithm by Feichtinger, Gröchenig, Strohmer and Rauth [67, 80, 123] fit perfectly within this framework, and can be easily combined with Chambolle's ideas to write a TV regularized irregular to regular sampling algorithm. The variational model (6.4) that we propose includes not only irregular sampling but also deconvolution, denoising, antialiasing and zoom. In this work we consider the inclusion of all these features except for the antialiasing. We shall prove the feasibility of a numerical algorithm for solving equation (6.4), by combining the Gröchenig and Chambolle techniques to be described below. This is, to the best of our knowledge, the first attempt in this direction, since in most of the irregular sampling literature, the only regularizers considered are linear, akin to Tikhonov regularization.

Let us finally explain the plan of the chapter. In section 6.2 we introduce the problem of irregular to regular sampling and review the ACT algorithm of Gröchenig and Strohmer [80]. Section 6.3 explains how to extend the Chambolle's algorithm [40] for total variation based image denoising to the more general case of irregular to regular sampling, deconvolution, zoom and denoising. The variational problem (6.4) proposed for the general restoration is studied in section 6.4. In section 6.5 we explain how to use local constraints in the proposed model. The inclusion of zoom in the general algorithm is studied in more detail in section 6.6. Finally in section 6.7 we show some experimental results and in section 6.8 we give the conclusions of this work.

6.2 Irregular to regular sampling

Digital data is usually obtained after sampling a continuous signal on a uniform cartesian grid. Then, the sampling set is said to be *uniform* or *regular*. However, it turns out that in many applications we only have access to a non-uniform sampling set. Aldroubi and Gröchenig [1] refer some examples of this kind of applications. In astronomy, weather conditions make rather it impossible to have regular measurements collections. In communication theory, signals that have originally been regularly sampled, often result in non-uniform samples due to sample losses in transmission or to damages in storage devices. In medical imaging spiral sampling is used for fast magnetic resonance imaging (MRI) and polar sampling is used in computerized tomography (CT). In other disciplines such as seismology, geophysics and spectroscopy data is only available on a non-uniform grid.

The micro-vibrations of satellites and the irregularities in the sensor position result in irregular sampling sets in satellite imaging. In most cases, the knowledge of certain vibration modes and the analysis of acquired images help to estimate with high accuracy the perturbations in the sampling grid which can be modeled [2] by:

$$\varepsilon(x) = \sum_{k=1}^{M} a_k(x) cos(\langle \omega_k, x \rangle + \phi_k) \tag{6.5}$$

where $a_k(x)$ are smooth modulation functions and the vibration frequencies ω_k are an order of magnitude (or even more) below the Nyquist frequency of the sampling rate. We call T_ε the period of the maximum vibration frequency and A the standard deviation of each component of the vector $\varepsilon(x)$. The bound on the modulation functions is inversely proportional to ω_k and the number of vibration modes is small. This results in smooth and small perturbations not higher than $0.1c$ where c is the distance between sensors. As a consequence, these perturbations are not visible and we should talk of *perturbed sampling* rather than irregular sampling in those cases. Even if the image distortion is not evident from a geometrical point of view it is very important to correct the perturbations in image registration applications where a sub-pixel accuracy is necessary.

There exist many works on the literature dealing with the irregular sampling problem. However, many of them are iterative algorithms which are adapted to well-conditioned problems and small data sets but are inadequate to more realistic problems with large sampling sets and their generalization to the two dimensional case is not always evident. A comparison between several iterative methods can be found in [13, 68]. In [2] the author makes a survey of other techniques [67, 80, 123, 1] which are well-suited for

the two dimensional case and also proposes a pseudo-inverse algorithm.

6.2.1 When is it possible to recover a regular sampling set

The problem of recovering a function f from its samples $\{f(x_i) : i \in \mathbb{Z}\}$ is ill-posed since there exist infinite solutions. In real applications one deals with band-limited functions, i.e., functions with a finite support in frequency domain. This assumption permits to redefine the problem and to study the necessary conditions to recover exactly the function f. The classical result of Shannon-Whittaker provides a reconstruction formula from uniform samples of finite energy functions [107].

Let $B_{[-1/2,1/2]^d}$ denote the space of band-limited functions f whose Fourier transform $\hat{f}$ is such that $\hat{f}(\xi) = 0$ for all $\xi \notin [-1/2, 1/2]^d$. The fact that a function $f \in L^2(\mathbb{R}^d) \cap B_{[-1/2,1/2]^d}$ could be recovered from its samples by the Shannon interpolation formula is equivalent to say that the set $\{e^{i2\pi\langle k,\xi\rangle}, k \in \mathbb{Z}^d\}$ forms an orthonormal basis of $L^2([-1/2, 1/2]^d)$. This basis is known as the *harmonic Fourier basis*. These results on uniform sampling have been extended to the non-uniform case and non-harmonic Fourier basis. For dimension $d = 1$, Kadec's theorem [95] states that for a perturbed sampling set of the form $\Lambda = \{\lambda_k \in \mathbb{R} : |\lambda_k - k| \leq c < 0.25\}_{k\in\mathbb{Z}}$, the set $\{e^{i2\pi\lambda_k\xi}, k \in \mathbb{Z}\}$ is a *Riesz basis* of $L^2([-1/2, 1/2])$. Hence, using Fourier transform methods, there exists a stable reconstruction of any function $f \in L^2(\mathbb{R}) \cap B_{[-1/2,1/2]}$ from its irregular samples $f(\lambda_k)$. A *stable reconstruction* implies an upper bound on the energy for any band-limited functions $f \in L^2(\mathbb{R}) \cap B_{[-1/2,1/2]}$,

$$||f||_2 \leq C \left(\sum_{\lambda_k \in \Lambda} |f(\lambda_k)|^2 \right)^{1/2}$$

A sampling set for which the reconstruction is stable is called a *(stable) set of sampling*. Notice that this notion is stronger than the notion of *set of uniqueness* which only requires that for all band-limited functions f, if $f|_\Lambda = 0$ then $f = 0$.

In the two dimensional case the Kadec's theorem requires a more restrictive bound on the maximal perturbation. If the perturbation is $\Lambda = \{\lambda_k \in \mathbb{R}^2 : ||\lambda_k - k|| \leq 0.11\}_{k\in\mathbb{Z}^2}$, then the set $\{e^{i2\pi\langle\lambda_k,\xi\rangle}, k \in \mathbb{Z}^2\}$ is a *Riesz basis* of $L^2([-1/2, 1/2]^2)$ and thus, there exists a stable reconstruction formula for any band-limited and finite energy function from its irregular samples. Moreover, if the perturbation is separable [2] then the problem is reduced to the one dimensional case in each dimension and the constant $c < 0.25$ is the maximal perturbation allowed in each dimension in order to have a Riesz basis and a stable reconstruction.

The sampling set in Kadec's theorem is just a perturbation of the uniform sampling grid. For general irregular sampling sets, the work of Beurling [28] and Landau [100] establish the necessary and sufficient conditions for stable reconstructions. A set Λ is a stable set of sampling if and only if the *Beurling density*

$$D(\Lambda) = \lim_{r\to\infty} \inf_{x\in\mathbb{R}^d} \frac{\#(\Lambda \cap (x + [0,r]^d))}{r^d}$$

satisfies $D(\Lambda) > 1$. Results about arbitrary sets of sampling are connected to the general notion of frames [28, 79]. The concept of frames generalizes the notion of orthogonal bases and Riesz bases in Hilbert spaces as well as of unconditional bases in some Banach spaces (see [1] and references therein).

6.2.2 ACT algorithm

One of the best performing reconstruction methods available for irregular to regular sampling is the ACT algorithm developed initially by Feichtinger et al. [67] and further analyzed, refined and generalized by Gröchenig and Strohmer [80], and Rauth [123]. The method intelligently combines an accelerated version of the *frame iteration* derived from the proof of Kadec's theorem, with *adaptive weights* in order to improve the condition number of the problem and to provide explicit estimates for the rate of convergence, a *conjugate gradient* iteration which accelerates convergence, and the formulation of the problem as a *Toeplitz system* in order to gain structure and thus numerical efficiency. Furthermore the preparation steps before the conjugate gradient iteration can start, can benefit from the USFFT (for unequally spaced fast Fourier transform) algorithm by Beylkin [29, 30].

More precisely, the algorithm is based on a representation of an $N \times N$ periodic band-limited function u as a trigonometric polynomial of order N^2

$$u(x) = \sum_{n\in[1,N]^2} a_n e^{\frac{2\pi i}{N}\langle n,x\rangle} \tag{6.6}$$

so that the interpolation conditions become

$$g_k = u(\lambda_k) = \sum_{n\in[1,N]^2} a_n e^{\frac{2\pi i}{N}\langle n,\lambda_k\rangle} \quad k \in \{1,\dots,M\}, \tag{6.7}$$

or equivalently in matrix form

$$g = Sa, \quad \text{where} \quad S = ((s_{kn})),\ s_{kn} = e^{\frac{2\pi i}{N}\langle n,\lambda_k\rangle} \tag{6.8}$$

i.e. S is the van der Monde matrix associated to the trigonometric polynomial. Now the problem is reduced to solving for a the system of linear equations (6.8). But if Λ contains

some regions with extremely dense sampling or large gaps the system will not be well balanced. In order to improve the condition number the k-th equation is multiplied by a weight

$$w_k = \text{area}\left(\{x : |x - \lambda_k| < |x - \lambda_j|, \forall j \neq k\}\right) \tag{6.9}$$

which is inversely proportional to the sampling density at λ_k. Thus, the use of weights w_k compensates the local variations in the sampling density. Moreover, the adaptative weights method provides explicit estimates for the rate of convergence and therefore gives useful stopping criteria. In addition, instead of solving the linear system (6.8) directly, it will be more convenient to solve the normal equations as an optimization problem

$$\min_a \|S^*WSa - S^*Wg\|^2 \tag{6.10}$$

because the $N^2 \times N^2$ matrix $T = S^*WS$ (where $W = \text{diag}(\{w_k\}_{k=1..N})$) is always a square matrix and can be shown to have Toeplitz structure, so that the multiplication Ta can be efficiently computed in $(2N)^2(\log_2((2N)^2) + 1)$ time using Fourier methods. Defining $T = S^*WS$ and $b = S^*Wg$ in equation (6.10), the non-harmonic series

$$t_n = \sum_{k\in[1,M]^2} e^{-\frac{2\pi i}{N}\langle n,\lambda_k\rangle} w_k \tag{6.11}$$

$$b_n = \sum_{k\in[1,M]^2} e^{-\frac{2\pi i}{N}\langle n,\lambda_k\rangle} w_k g_k \tag{6.12}$$

can be approximated using the USFFT [29] in $CM^2\log_2(M^2)$ time each, where C is a constant, which is inversely proportional to the required precision.

The overall procedure can be summarized as shown in the following algorithm.

Algorithm 1: ACT algorithm

REQUIRE: M^2 irregular samples in vector g, and degree $N^2 \leq M^2$ of trigonometric polynomial.

ENSURE: N^2 regular samples in vector u.

1. Compute $T = S^*WS$ and $b = S^*Wg$ using the USFFT.
2. Minimize $\|Ta - b\|^2$ using conjugate gradients.
3. Compute the regular samples $u_k = u(k)$ for $k \in [1, N]^2$ by applying the inverse FFT to a.

The convergence rate of the CG algorithm is determined by the condition number $\kappa = \text{cond}(T)$, or the ratio of the largest to the smallest eigenvalue of T. More precisely

at each iteration the approximation error is decreased by a factor $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ [73]. Gröchenig [78], Gröchenig and Strohmer [80] provided a useful characterization of the condition number of T in the 2-dimensional case,

Proposition 1 (ACT convergence rate) *If the sampling set is γ-dense with*

$$\gamma < \frac{\log 2}{4\pi}$$

then the matrix T is invertible and its condition number is

$$\kappa \leq \frac{4}{(2 - e^{4\pi\gamma})^2} \tag{6.13}$$

and Algorithm 1 converges to the exact solution.

Remarks:

1. The invertibility of T does not depend on the choice of the weights w_k. Indeed, S^*WS and S^*S have the same kernel and the matrix T will be invertible if and only if its kernel is $\{0\}$. Proposition 1 states the maximal gap $\upsilon = 2\gamma$ for a sampling set so that T is invertible.

2. The estimate (6.13) of κ in proposition 1 is only valid if we are using adaptative weights.

3. As a consequence of using this kind of weights, the rate of convergence is independent of possible clustering shapes in the set of samples and depends only on the maximal gap between samples.

4. Increasing the number of samples (oversampling) improves the condition number (6.13).

5. Note that the sampling has to be much more dense than the critical sampling rate, for the algorithm to ensure convergence to the exact result, a condition which does not hold in the case of satellite imaging.[1] Nevertheless, even when T is not invertible (and has an infinite condition number), the CG iteration chooses among the minimizers of $\|Ta - b\|$ the one of minimal norm, i.e. $a = T^+b$, where T^+ is the pseudo-inverse of T.

6. In our particular case we are working with perturbed sampling sets. The perturbation in satellite imaging is of the form (6.5), producing smooth and small

[1] The bound on the density γ can be significantly relaxed in the one-dimensional case and in the case of a "separable" two-dimensional perturbation [80].

perturbations over the uniform grid. Thus, the density of sampling points is very uniform and the use of weights w_k is not relevant in that case (in practice we fix $w_k = 1$ for all k).

On the other hand, if we know the a-priori spectral decay rate of the image

$$|\hat{u}(\omega)| \leq C\phi(\omega),$$

and the known transfer function, we can regularize the solution by imposing this decay rate (typically $\phi(\omega) = (1+\omega)^{-r}$ for $r \in [1,2]$ or a combination of this natural decay rate). As proposed in [80], we minimize the modified system

$$\min \|DTa - Db\| \tag{6.14}$$

where $D = \operatorname{diag}\left(\{\phi(\frac{2\pi}{N}n)\}_n\right)$ is a diagonal matrix containing the corresponding values of $\phi(\omega)$. In this way we shall obtain the solution $a = (DT)^+ Db$, where $(DT)^+$ is the pseudo-inverse of DT. The action of the matrix D is to penalize more the errors in high frequencies. Note that Algorithm 1 can be easily adapted to solve the problem (6.14), it suffices to replace the solution of $Ta = b$ by conjugate gradients in step 2 by the solution of $DTa = Db$ by preconditioned conjugate gradients. In contrast to the original purpose of the preconditioned conjugate gradients, here the matrix D is not acting as a preconditioner (actually it is possible that we decrease the rate of convergence using D) but to obtain an improved and regularized solution. In [80] they also propose an alternative way to impose a specified spectral decay, it is based on solving $TDc = b$ where $a = Dc$ whose the solution is $a = D(TD)^+ b$ (although we have only implemented the first approach).

This discussion explains why the ACT algorithm of Gröchenig and Strohmer [80] provides good approximations to the exact solution, even when the convergence conditions in Proposition 1 are not satisfied (as in the case of satellite imaging), while at the same time the convergence rate is not that good.

6.3 Generalization of Chambolle's algorithm

We shall denote by Ω the image domain, which is assumed to be a rectangle in $\mathbb{R}^2$. For simplicity, we shall denote

$$J(u) := \begin{cases} \displaystyle\int_\Omega |Du| & \text{if} \quad u \in BV(\Omega) \\ \\ +\infty & \text{if} \quad u \in L^2(\Omega) \setminus BV(\Omega). \end{cases} \tag{6.15}$$

where $BV(\Omega)$ denotes the space of bounded variation functions in Ω.

In [40], Chambolle proposed an algorithm to solve the total variation approach to image denoising, which amounts to solve the following constrained minimization problem

$$\begin{aligned} &\min \int_\Omega |Du|\, dx \\ &\text{with} \quad \textstyle\int_\Omega |u-g|^2\, dx = \sigma^2 |\Omega|, \end{aligned} \tag{6.16}$$

The constraint corresponds to the assumption that the standard deviation of the noise is σ. The constraint is a way to incorporate the simplified image acquisition model given by $g = u + n$, reflecting the fact that the ideal image u has been distorted by a Gaussian noise n. In practice, problem (6.16) is solved via the following unconstrained minimization problem

$$\min_u \quad J(u) + \frac{1}{2\lambda} \int_\Omega (u-g)^2\, dx \tag{6.17}$$

where $\lambda > 0$ is some Lagrange multiplier. Moreover, for any $\lambda > 0$, the solution u of (6.17) satisfies the Euler-Lagrange equation

$$u + \lambda \partial J(u) \ni g, \tag{6.18}$$

where $\partial J(u)$ denotes the subdifferential of J at u, i.e.,

$$\partial J(u) = \{w \in L^2(\Omega) : J(v) - J(u) \geq \langle w, v-u \rangle, \forall v \in L^2(\Omega)\}$$

We recall that $u \rightarrow \partial J(u)$ is a multivoque operator, this explains the $\ni$ sign in (6.18) instead of the more classical equality sign.

In [40], the author proposed an algorithm to solve (6.18) and he could prove it to be convergent. This algorithm was also proposed in [20] in a different context. Our purpose is to extend this algorithm to be able to apply it in the more general case where the image formation model is given by (6.1) (see Section 6.4).

Let us first develop the basic arguments in a continuous framework. For that, let Q be a bounded linear self-adjoint operator in $L^2(\Omega)$. We assume that Q is invertible. We want to solve the equation

$$Q(u) + \beta \partial J(u) \ni b \tag{6.19}$$

where $b \in L^2(\Omega)$, $\beta > 0$. In a more classical way we would write this equation as

$$\begin{aligned} &Q(u) - \beta \mathrm{div}\Big(\frac{Du}{|Du|}\Big) = b, \\ &[z, \nu^\Omega] = 0 \end{aligned} \tag{6.20}$$

where z represents the vector field $\frac{Du}{|Du|}$, since $\partial J(u)$ is nothing else than $-\mathrm{div}\left(\frac{Du}{|Du|}\right)$ plus Neumann boundary conditions. The case analyzed in [40] corresponds to the case where Q is the identity operator. Following the arguments in [40], let us write (6.19) in the form

$$\frac{1}{\beta}(b - Q(u)) \in \partial J(u), \tag{6.21}$$

and this implies that

$$u \in \partial J^*(\frac{1}{\beta}(b - Q(u))), \tag{6.22}$$

where J^* denotes the Legendre-Fenchel transform of J given by

$$J^*(v) = \sup_u\{< v, u > -J(u)\},$$

since the relation $v \in \partial J(w)$ is equivalent to $w \in \partial J^*(v)$. Let

$$\tilde{b} = Q^{-1}b, \qquad w = \tilde{b} - u$$

so that

$$Q(u) + Q(w) = b. \tag{6.23}$$

We may write (6.22) as

$$u \in \partial J^*(\frac{1}{\beta}Q(w)), \tag{6.24}$$

which, after multiplication by $\beta^{-1}Q$ becomes

$$\beta^{-1}Q(u) \in \beta^{-1}Q \circ \partial J^*(\frac{1}{\beta}Q(w)), \tag{6.25}$$

Since Q is a bounded self-adjoint operator in $L^2(\Omega)$, we have $\partial(J^* \circ (\beta^{-1}Q)) = (\beta^{-1}Q) \circ \partial J^* \circ (\beta^{-1}Q)$, hence we may write (6.25) as

$$\beta^{-1}b - \beta^{-1}Qw = \beta^{-1}Q(u) \in \partial(J^* \circ (\beta^{-1}Q)))(w). \tag{6.26}$$

Now, we observe that (6.26) corresponds to the Euler-Lagrange equations of the variational problem

$$\min_{\overline{w}} \quad \frac{1}{2} < Q\overline{w}, \overline{w} > - < b, \overline{w} > +\beta(J^* \circ (\beta^{-1}Q))(\overline{w}). \tag{6.27}$$

Observe that, since J is homogeneous of degree one (i.e. $J(\lambda u) = \lambda J(u)$ for every u and any $\lambda > 0$), then J^* is the indicator function of a convex set K ([63]):

$$J^*(v) = \chi_K(v) =:= \begin{cases} 0 & \text{if} \quad v \in K \\ \\ +\infty & \text{if} \quad v \notin K. \end{cases} \tag{6.28}$$

In our case

$$K := \{\mathrm{div}\,\xi : \xi \in L^\infty(\Omega, \mathbb{R}^N), \|\xi\|_\infty \leq 1,\ [\xi \cdot \nu^\Omega]|_{\partial\Omega} = 0\}$$

where $\nu^\Omega(x)$ denotes the outer unit normal at the point $x \in \partial\Omega$ (see, for instance, [54], vol. 5). Let us write

$$\mathcal{V} = \{\xi \in L^\infty(\Omega, \mathbb{R}^N) : \|\xi\|_\infty \leq 1,\ [\xi \cdot \nu^\Omega]|_{\partial\Omega} = 0\}.$$

Then we may write (6.27) as

$$\min_{\{\overline{w}:\beta^{-1}Q(\overline{w})\in K\}} \quad \frac{1}{2} < Q\overline{w}, \overline{w} > - < b, \overline{w} > \tag{6.29}$$

Writing $\beta^{-1}Q(\overline{w}) \in K$ as $\beta^{-1}Q(\overline{w}) = -\mathrm{div}\,\xi$ for some $\xi \in \mathcal{V}$, we may write (6.29) as

$$\min_{\{\xi\in\mathcal{V}\}} \quad \frac{\beta}{2} < Q^{-1}\mathrm{div}\,\xi, \mathrm{div}\,\xi > + < \mathrm{div}\,\xi, Q^{-1}b > \tag{6.30}$$

after dividing by β the energy functional. Now, observing that

$$\frac{\beta}{2}\|Q^{-1/2}\mathrm{div}\,\xi + \beta^{-1}Q^{-1/2}b\|^2 = \frac{\beta}{2} < Q^{-1}\mathrm{div}\,\xi, \mathrm{div}\,\xi > + < \mathrm{div}\,\xi, Q^{-1}b > + \frac{\beta}{2}\|\beta^{-1}Q^{-1/2}b\|^2$$

and defining the new norm

$$\|\overline{w}\|_Q = \|Q^{-1/2}\overline{w}\|$$

we may write the minimization problem (6.30) as

$$\min_{\{\xi\in\mathcal{V}\}} \|\mathrm{div}\,\xi + \beta^{-1}b\|_Q \tag{6.31}$$

In other words, $\mathrm{div}\,\xi$ is the projection of $-\beta^{-1}b$ onto the convex set K in the Hilbert space norm $\|\cdot\|_Q$.

Introducing the constraint $|\xi(x)| \leq 1$, $x \in \Omega$ a.e. by means of Lagrange multipliers $\alpha(x) \geq 0$, we minimize

$$\min_{\{\xi\in\mathcal{V}\}} \quad \frac{1}{2} < Q^{-1}\mathrm{div}\,\xi, \mathrm{div}\,\xi > + < \mathrm{div}\,\xi, Q^{-1}(\beta^{-1}b) > + \int_\Omega \alpha(x)|\xi(x)|^2\,dx. \tag{6.32}$$

The corresponding Euler-Lagrange equations are

$$\nabla[Q^{-1}\mathrm{div}\,\xi + Q^{-1}(\beta^{-1}b)](x) - \alpha(x)\xi(x) = 0, \tag{6.33}$$

supplemented with the boundary condition

$$[\xi \cdot \nu^\Omega] = 0 \quad \text{on } \partial\Omega.$$

6.3.1 The discrete case formulation

We denote by X the Euclidean space $\mathbb{R}^{N\times N}$. The Euclidean scalar product and the norm in X will be denoted by $\langle\cdot,\cdot\rangle_X$ and $\|\cdot\|_X$ respectively. Then the image $u \in X$ is $u = (u(i,j))$, and the vector field is $\xi : \{1,\dots,N\}\times\{1,\dots,N\} \to \mathbb{R}^N$. If $u \in X$, the discrete gradient is a vector in $Y = X \times X$ given by

$$\nabla^{+,+}u := (\nabla_x^+ u, \nabla_y^+ u)$$

where

$$\nabla_x^+ u(i,j) = \begin{cases} u(i+1,j) - u(i,j) & \text{if} \quad i < N \\ 0 & \text{if} \quad i = N. \end{cases} \tag{6.34}$$

$$\nabla_y^+ u(i,j) = \begin{cases} u(i,j+1) - u(i,j) & \text{if} \quad j < N \\ 0 & \text{if} \quad j = N. \end{cases} \tag{6.35}$$

for $i,j = 1,\dots,N$. Other choices of the gradient are possible, this one will be convenient for the developments below.

The Euclidean scalar product in Y is defined in the standard way by

$$\langle p, q\rangle_Y = \sum_{1\le i,j\le N} (p_{i,j}^1 q_{i,j}^1 + p_{i,j}^2 q_{i,j}^2)$$

for every $p = (p^1, p^2)$, $q = (q^1, q^2) \in Y$. The norm of $p = (p^1, p^2) \in Y$ is, as usual, $|p| = \langle p, p\rangle_Y^{1/2}$. Then the discrete total variation is

$$J_d(u) = |\nabla^{+,+}u|_Y = \sum_{1\le i,j\le N} |\nabla^{+,+}u(i,j)|. \tag{6.36}$$

We have

$$J_d(u) = \sup_{p\in Y, |p_{i,j}|_Y \le 1\, \forall (i,j)} \langle p, \nabla^{+,+}u\rangle_Y \tag{6.37}$$

By analogy with the continuous setting, we introduce a discrete divergence $\text{div}^{-,-}$ as the dual operator of $\nabla^{+,+}$, i.e., for every $p \in Y$ and $u \in X$ we have

$$\langle -\text{div}^{-,-} p, u\rangle_X = \langle p, \nabla^{+,+}u\rangle_Y.$$

One can easily check that $\text{div}^{-,-}$ is given by

$$\begin{aligned} \text{div}^{-,-}p(i,j) \;=\; & \begin{cases} p^1(i,j) - p^1(i-1,j) & \text{if} \quad 1 < i < N \\ p^1(i,j) & \text{if} \quad i = 1 \\ -p^1(i-1,j) & \text{if} \quad i = N. \end{cases} \\ + \; & \begin{cases} p^2(i,j) - p^2(i,j-1) & \text{if} \quad 1 < j < N \\ p^2(i,j) & \text{if} \quad j = 1 \\ -p^2(i,j-1) & \text{if} \quad j = N. \end{cases} \end{aligned} \tag{6.38}$$

for every $p = (p^1, p^2) \in Y$. In this setting

$$J_d^*(v) = \chi_{K_d}(v) =:= \begin{cases} 0 & \text{if} \quad v \in K_d \\ \\ +\infty & \text{if} \quad v \notin K_d \end{cases} \tag{6.39}$$

with

$$K_d := \{\mathbf{div}^{-,-}\, p : |p_{i,j}|_Y \leq 1 \, \forall (i,j) \in \{1, \ldots, N\}\}.$$

Let us also denote by Q a self-adjoint positive definite matrix acting on X, let $b \in X$. Our purpose is to solve the equation

$$Q(u) + \beta \partial J_d(u) \ni b \tag{6.40}$$

for some $u \in X$. Proceeding as in the continuous framework we have to solve the problem

$$\min_{\{\xi \in \mathcal{V}_d\}} \quad \frac{\beta}{2} < Q^{-1}\mathbf{div}^{-,-}\, \xi, \mathbf{div}^{-,-}\, \xi > + < \mathbf{div}^{-,-}\, \xi, Q^{-1} b > \tag{6.41}$$

which may be written as

$$\min_{\{\xi \in \mathcal{V}_d\}} \|\mathbf{div}^{-,-}\, \xi + \beta^{-1} b\|_Q \tag{6.42}$$

where

$$\mathcal{V}_d = \{\xi \in Y : \|\xi(i,j)\|_Y \leq 1, \ \forall i,j = 1, \ldots, N\}.$$

and

$$\|\overline{w}\|_Q = \|Q^{-1/2}\overline{w}\|_X, \quad \overline{w} \in X.$$

The solution of (6.42) amounts to the unique projection of $-\beta^{-1}b$ onto the convex set K_d in X endowed with the norm $\|\cdot\|_Q$. We shall denote this projection by $\Pi_{K_d}^Q(-\beta^{-1}b)$.

Introducing the constraint $\xi \in \mathcal{V}_d$ by means of Lagrange multipliers, and computing the corresponding Euler-Lagrange equations we arrive at the analogous of equation (6.33), namely,

$$\nabla^{+,+}[Q^{-1}\mathbf{div}^{-,-}\, \xi + Q^{-1}(\beta^{-1}b)](i,j) - \alpha(i,j)\xi(i,j) = 0, \tag{6.43}$$

with either $\alpha(i,j) > 0$ and $|\xi(i,j)| = 1$, or $|\xi(i,j)| < 1$, and $\alpha(i,j) = 0$. In the later case, we have $\nabla^{+,+}[Q^{-1}\mathbf{div}^{-,-}\, \xi + Q^{-1}(\beta^{-1}b)](i,j) = 0$. In any case we have

$$\alpha(i,j) = |\nabla^{+,+}[Q^{-1}\mathbf{div}^{-,-}\, \xi + Q^{-1}(\beta^{-1}b)](i,j)|. \tag{6.44}$$

Let $\nu > 0$, $\xi^0 = 0$, $n \geq 0$. We solve (6.43) using the following gradient descent (or fixed point) algorithm

$$\begin{aligned}\xi^{n+1}(i,j) = \xi^n(i,j) + \nu\nabla^{+,+}[Q^{-1}\mathrm{div}^{-,-}\,\xi^n + Q^{-1}(\beta^{-1}b)](i,j)\\ -\nu|\nabla^{+,+}[Q^{-1}\mathrm{div}^{-,-}\,\xi^n + Q^{-1}(\beta^{-1}b)](i,j)|\xi^{n+1}(i,j),\end{aligned} \tag{6.45}$$

hence

$$\xi^{n+1}(i,j) = \frac{\xi^n(i,j) + \nu\nabla^{+,+}[Q^{-1}\mathrm{div}^{-,-}\,\xi^n + Q^{-1}(\beta^{-1}b)](i,j)}{1 + \nu|\nabla^{+,+}[Q^{-1}\mathrm{div}^{-,-}\,\xi^n + Q^{-1}(\beta^{-1}b)](i,j)|}. \tag{6.46}$$

Theorem 3 *In the discrete framework, assuming that $\nu < \frac{1}{8\|Q^{-1/2}\|^2}$, ξ^n converges to a solution ξ of (6.43) where α is given by (6.44). Moreover $div^{-,-}\,\xi^n$ converges to the unique solution $\Pi^Q_{K_d}(-\beta^{-1}b)$ of (6.42).*

In particular, we deduce that $w^n = -\beta Q^{-1}(\mathrm{div}^{-,-}\,\xi^n)$ converges to a solution $w = -\beta Q^{-1}(\mathrm{div}^{-,-}\,\xi)$ of the discrete analogous of (6.29).

Then the solution u of (6.40) may be recovered from (6.23), i.e.,

$$u = Q^{-1}(b + \beta\mathrm{div}^{-,-}\,\xi).$$

Proof. For simplicity, let us denote ∇ and div instead of $\nabla^{+,+}$ and $\mathrm{div}^{-,-}$. Let us fix $n \geq 0$, and let

$$\eta = \frac{\xi^{n+1} - \xi^n}{\nu}, \quad \rho = |\nabla Q^{-1}(\mathrm{div}\,\xi^n + (\beta^{-1}b))|\xi^{n+1},$$

so that

$$\eta = \nabla Q^{-1}(\mathrm{div}\,\xi^n + (\beta^{-1}b)) - \rho.$$

Then

$$\begin{aligned}&\|Q^{-1/2}(\mathrm{div}\,\xi^{n+1} + \beta^{-1}b)\|_X^2 = \|Q^{-1/2}(\mathrm{div}\,\xi^n + \nu\mathrm{div}\,\eta + \beta^{-1}b)\|_X^2\\ =\ &\|Q^{-1/2}(\mathrm{div}\,\xi^n + \beta^{-1}b)\|_X^2 + \nu^2\|Q^{-1/2}\mathrm{div}\,\eta\|_X^2 + 2\nu\langle Q^{-1/2}(\mathrm{div}\,\xi^n + \beta^{-1}b), Q^{-1/2}\mathrm{div}\,\eta\rangle_X\\ \leq\ &\|Q^{-1/2}(\mathrm{div}\,\xi^n + \beta^{-1}b)\|_X^2 + k^2\nu^2\|\eta\|_Y^2 - 2\nu\langle\nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b), \eta\rangle_Y\end{aligned}$$

where k is the norm of the operator $Q^{-1/2}\mathrm{div} : Y \to X$, that we will estimate later on. Now

$$\begin{aligned}&2\langle\nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b), \eta\rangle_Y\\ =\ &\langle\nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b), \eta\rangle_Y + \langle\nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b), \eta\rangle_Y\\ =\ &\langle\nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b), \nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b) - \rho\rangle_Y + \langle\eta + \rho, \eta\rangle_Y\\ =\ &\|\nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b)\|_Y^2 + \|\eta\|_Y^2 - \langle\nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b) - \eta, \rho\rangle_Y\\ =\ &\|\nabla Q^{-1}(\mathrm{div}\,\xi^n + \beta^{-1}b)\|_Y^2 + \|\eta\|_Y^2 - \|\rho\|_Y^2\end{aligned}$$

Combining the two computations, we have

$$\begin{aligned}\|Q^{-1/2}(\operatorname{div}\xi^{n+1}+\beta^{-1}b)\|_X^2 &\leq \|Q^{-1/2}(\operatorname{div}\xi^{n}+\beta^{-1}b)\|_X^2+(k^2\nu-1)\nu\|\eta\|_Y^2\\ &+ \nu(\|\rho\|_Y^2-\|\nabla Q^{-1}(\operatorname{div}\xi^n+\beta^{-1}b)\|_Y^2).\end{aligned}$$

Observe that

$$\|\rho\|_Y=\|\nabla Q^{-1}(\operatorname{div}\xi^n+\beta^{-1}b)\|_Y\|\xi^{n+1}\|_Y\leq\|\nabla Q^{-1}(\operatorname{div}\xi^n+\beta^{-1}b)\|_Y.$$

By choosing $\nu<\frac{1}{k^2}$ we obtain

$$\|Q^{-1/2}(\operatorname{div}\xi^{n+1}+\beta^{-1}b)\|_X^2 \leq \|Q^{-1/2}(\operatorname{div}\xi^{n}+\beta^{-1}b)\|_X^2,$$

i.e., the sequence $\|Q^{-1/2}(\operatorname{div}\xi^n+\beta^{-1}b)\|_X^2$ is decreasing. Let

$$m=\lim_n\|Q^{-1/2}(\operatorname{div}\xi^n+\beta^{-1}b)\|_X^2$$

and let $\overline{\xi}$ be the limit of a convergent subsequence $\{\xi^{n_k}\}$ of $\{\xi^n\}$. Modulo a extraction of a subsequence, we may assume that $\{\xi^{n_k+1}\}$ converges to a vector $\overline{\xi}'\in Y$. Observe that

$$\|Q^{-1/2}(\operatorname{div}\overline{\xi}'+\beta^{-1}b)\|_X^2=\|Q^{-1/2}(\operatorname{div}\overline{\xi}+\beta^{-1}b)\|_X^2=m.$$

Letting $n_k\to\infty$ in (6.46) we obtain

$$\overline{\xi}'(i,j)=\frac{\overline{\xi}(i,j)+\nu\nabla[Q^{-1}\operatorname{div}\overline{\xi}+Q^{-1}(\beta^{-1}b)](i,j)}{1+\nu|\nabla[Q^{-1}\operatorname{div}\overline{\xi}+Q^{-1}(\beta^{-1}b)](i,j)|}. \tag{6.47}$$

If we repeat now the above calculations we obtain

$$\begin{aligned}\|Q^{-1/2}(\operatorname{div}\overline{\xi}'+\beta^{-1}b)\|_X^2 &\leq \|Q^{-1/2}(\operatorname{div}\overline{\xi}+\beta^{-1}b)\|_X^2+(k^2\nu-1)\nu\|\overline{\eta}\|_Y^2\\ &+ \nu(\|\overline{\rho}\|_Y^2-\|\nabla Q^{-1}(\operatorname{div}\overline{\xi}'+\beta^{-1}b)\|_Y^2),\end{aligned}$$

where

$$\overline{\eta}=\frac{\overline{\xi}'-\overline{\xi}}{\nu},$$

$$\overline{\rho}=\|Q^{-1/2}(\operatorname{div}\overline{\xi}+\beta^{-1}b)\|_X^2\overline{\xi}'.$$

Hence

$$(k^2\nu-1)\|\overline{\eta}\|_Y^2+(\|\overline{\rho}\|_Y^2-\|\nabla Q^{-1}(\operatorname{div}\overline{\xi}'+\beta^{-1}b)\|_Y^2)=0$$

If $\nu<\frac{1}{k^2}$ we deduce that $\overline{\eta}=0$, that is, $\overline{\xi}'=\overline{\xi}$. Hence

$$\nabla Q^{-1}(\operatorname{div}\overline{\xi}+\beta^{-1}b)-\|\nabla Q^{-1}(\operatorname{div}\overline{\xi}+\beta^{-1}b)\|_Y\overline{\xi}=0$$

which is the Euler-Lagrange equation (6.43). One deduces that $\operatorname{div}\overline{\xi}$ is the projection $\Pi^Q_{K_d}(-\beta^{-1}b)$. Since this projection is unique, we deduce that all the sequence $\operatorname{div}\xi^n$ converges to $\Pi^Q_{K_d}(-\beta^{-1}b)$.

Finally, since

$$\|\operatorname{div}\xi\|_X^2 \leq 8\|\xi\|_Y^2$$

for every $\xi \in Y$, we have

$$k^2 \leq 8\|Q^{-1/2}\|$$

where $\|Q^{-1/2}\|$ is the norm of $Q^{-1/2}$ on X. Thus, if $\nu < \frac{1}{8\|Q^{-1/2}\|^2}$ we also have that $\nu < \frac{1}{k^2}$.

6.4 TV regularized irregular to regular sampling, deconvolution, denoising and zoom

First, assume that we represent u by a vector of $(2M+1)\times(2M+1)$ regular samples that we shall call also u by abuse of notation. Let $F = FFT/(2M+1)^2$ be the normalized FFT operator, such that $a = Fu$ is defined by

$$a_k = \frac{1}{(2M+1)^2}\sum_{j=(-M,-M)}^{j=(M,M)} u(j/\alpha)e^{-\frac{2\pi i}{2M+1}\langle j,k\rangle}$$

Here α is the oversampling factor with respect to the critical (integer) sampling rate. Then the inverse operator $F^{-1} = IFFT$ is just the inverse FFT

$$u(j/\alpha) = \sum_{k=(-M,-M)}^{(M,M)} a_k e^{\frac{2\pi i}{2M+1}\langle j,k\rangle}.$$

However the adjoint operator is $F^* = IFFT/(2M+1)^2$, and therefore $F^*F = I/(2M+1)^2$.

Similarly, let's call S_Λ or simply S, the irregular sampling operator such that if $a = Fu$ then

$$u(\lambda_j) = (Sa)_j = \sum_{k=(-M,-M)}^{(M,M)} a_k e^{\frac{2\pi i}{2M+1}\langle \lambda_j,k\rangle}.$$

and also $P = \operatorname{diag}(\{\hat{p}(\frac{2\pi k}{2M+1})\}_{k=(-M,-M)}^{(M,M)})$, $H = \operatorname{diag}(\{\hat{h}(\frac{2\pi k}{2M+1})\}_{k=(-M,-M)}^{(M,M)})$ the diagonal operators containing along their diagonals the Fourier coefficients of the spectral projector p and the blurring kernel h respectively. Similarly $W = \operatorname{diag}(\{w_j\}_{j=1..N})$ is a diagonal matrix containing the weights for the corresponding sampling points $\lambda_j \in \Lambda$.

Then equation (6.4) can be written in vector notation as

$$\min_u \frac{1}{2}\|W(SHPFu - g)\|^2 + \lambda J(u) \tag{6.48}$$

whose Euler-Lagrange equation is readily written as

$$0 \in (F^*P^*H^*S^*W^2SHPFu - F^*P^*H^*S^*W^2g + \lambda\partial J(u)) \tag{6.49}$$

where $\partial J(u)$ is the subdifferential of $J(u)$. Now define

$$\begin{aligned} T &:= S^*W^2S/(2M+1)^2 \quad T' := P^*H^*THP \\ \hat{r} &:= S^*W^2g/(2M+1)^2 \quad \hat{r}' := P^*H^*\hat{r} \end{aligned} \tag{6.50}$$

(the $(2M+1)^2$ normalization factor is included to absorb the corresponding factor from $F^* = F^{-1}/(2M+1)^2$). Recall that in this equation T and $\hat{r}$ can be computed from g and Λ in $O(N \log N)$ time by means of the NFFT [121], where N is the number of samples in Λ. In addition T has Toeplitz structure and the products of T by a vector can be computed in $O(M^2 \log M^2)$ time using the FFT. Therefore products of T' by a vector also take $O(M^2 \log M^2)$ time, since P and H are diagonal.

Summarizing, the Euler-Lagrange equation can be written

$$0 \in (F^{-1}T'Fu - F^{-1}\hat{r}') + \lambda\partial J(u)) \tag{6.51}$$

and solved by looking for a steady state for the evolution problem

$$\frac{\partial u}{\partial t} = -[F^{-1}T'Fu - F^{-1}\hat{r}' + \lambda\partial J(u))] \tag{6.52}$$

by means of the implicit scheme

$$u_{n+1} - u_n = -\tau[F^{-1}T'Fu_{n+1} - F^{-1}\hat{r} + \lambda\partial J(u_{n+1})] \tag{6.53}$$

which can be rewritten as

$$Qu_{n+1} + \tau\lambda\partial J(u_{n+1}) = b_n \tag{6.54}$$

with

$$\begin{aligned} Q &= F^{-1}UF \\ U &= I + \tau T' \\ b_n &= u_n + \tau r' \\ r' &= F^{-1}\hat{r}' \end{aligned} \tag{6.55}$$

Now for any positive τ, Q is hermitian, positive definite (hence invertible) because T is hermitian positive semidefinite. So we can solve equation (6.54) by applying the extension of Chambolle's technique presented in section 6.3 as follows

$$Qu_{n+1} + \tau\lambda\operatorname{div}\xi = b_n \tag{6.56}$$

where ξ is the fixed point of

$$G(\xi) := \nabla[\tau\lambda Q^{-1}(\operatorname{div}\xi) + Q^{-1}b_n] = \alpha\xi \tag{6.57}$$

with the Lagrange multiplier α chosen in such a way that it ensures the constraint $|\xi| \leq 1$. This fixed point is obtained by the iteration

$$\begin{aligned} \alpha_k &= |G(\xi_k)| \\ \xi_{k+1} &= [\xi_k + \nu G(\xi_k)]/(1 + \nu\alpha_k) \end{aligned} \tag{6.58}$$

with ν satisfying the requirements of Theorem 3. In this case $\beta = \tau\lambda$, and $\|Q^{-1/2}\|^2 = \frac{1}{1+\tau r(T')}$ where $r(T')$ denotes the spectral radius of T'. The condition $\nu < \frac{1}{8\|Q^{-1/2}\|^2}$ can be written as

$$\nu < 0.125(1 + \tau r(T'))$$

and it is satisfied in particular if $\nu < 0.125$.

The two inversions of Q that appear in equation (6.57) can be efficiently solved by CG as in the ACT algorithm, since each product of Q by a vector only involves multiplication by Toeplitz or diagonal matrices. In addition, for low τ, the matrix Q is better conditioned than T, which means that CG should converge faster. In fact, in the particular case $P = H = I$ (no deconvolution and no zoom) the condition number of Q can be bounded by

$$\operatorname{cond}(Q) = \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)} = \frac{\lambda_{\max}(T)\tau + 1}{\lambda_{\min}(T)\tau + 1} \leq \tau\lambda_{\max}(T) + 1 \tag{6.59}$$

and according to [67], $\lambda_{\max}(T)$ can be bounded, at least in the separable case, by $(1+\upsilon)^4$, if υ is the "maximal gap" of the sampling set.

Putting all the pieces together we have the following algorithm:

Algorithm 2: TV based restoration algorithm

1. Compute $\hat{r}$ and (the first row and column in) T by means of the NFFT
2. Choose an initial condition $u_0 = 0$ or a better guess based on the irregular samples g.
   Set $n = 0$.
3. While $|u_{n+1} - u_n| > \theta_u\ \tau$ iterate in n
   (a) Update $b_n = u_n + \tau r'$
   (b) Solve $w_n = Q^{-1}b_n$ by CG up to precision θ_g, using w_{n-1} or 0 as initial condition.

(c) Initialize $\xi_0 = 0$ for $n = 0$, or with the last ξ_k from the previous iteration if $n > 0$.
Set $k = 0$.

(d) While $|\operatorname{div} \xi_{k+1} - \operatorname{div} \xi_k| > \theta_z$ iterate in k

i. Solve $v_k = Q^{-1}(\operatorname{div} \xi_k)$ by CG up to precision θ'_z, using as initial condition v_{k-1} for $k > 0$, the last v_k from the previous n-iteration for $k = 0, n > 0$, or 0 $k = n = 0$.

ii. Compute ξ_{k+1} from equation (6.58), i.e.

$$\begin{aligned} G_k &:= \nabla[\tau\lambda v_k + w_n] \\ \alpha_k &:= |G_k| \\ \xi_{k+1} &:= [\xi_k + \nu G_k]/(1 + \nu\alpha_k) \end{aligned}$$

(e) Solve $u_{n+1} = Q^{-1}(b_n + \lambda\tau \operatorname{div} \xi_{k+1})$ by CG up to precision θ_u, with initial condition $w_n + \lambda\tau v_k$

In order to reduce the number of parameters so that the user only needs to specify the TV Lagrange multiplier λ and the precision θ_u of the final solution u, we used the following criteria: The convergence of Chambolle's fixed point iteration in step 3.d.ii imposes $\nu = 0.125$ and the maximal value of $\tau = \lambda^{-1}$. We set the tolerance $\theta_z = \theta_u/2$ since Q^{-1} is contractive (i.e. this operation does not increase the error already present in $b_n + \lambda\tau \operatorname{div} \xi_{k+1}$) and we want an error of order θ_u in u_n. As we want a precision θ_z in $\operatorname{div} \xi_{k+1}$, we set $\theta_g = \theta_u/2$ and $\theta'_z = \theta_z/10$. The reason of the tolerance $\theta'_z = \theta_z/10$ in 3.d.i. is that from the numerical experimentations we get that the maximum number of iterations k in 3.d. is 10.

6.5 Working with local constraints

As noticed in [24], the use of a global constraint in image restoration problems based on total variation regularization may be inadequate in images which present very different regions. If λ controls the importance of the regularization term (as is in our case) and we set this parameter to be a high value, high textured regions will lose a great part of its structure. On the other hand, homogeneous zones are quite well denoised for a high λ. This observation led the authors in [24] to extend the total variation image restoration to the particular case of local constraints adapted to the different regions of the image.

They were concerned with the following problem

$$\min_u J(u) \tag{6.60}$$

$$\int_\Omega |Ku - g|^2 dx = \sigma^2 |\Omega| \tag{6.61}$$

where g is the image to be restored and K is the convolution operator. This minimization problem can be solved via the unconstrained problem

$$\min_u J(u) + \frac{\lambda}{2} \int_\Omega |Ku - g|^2 dx \tag{6.62}$$

where λ is a Lagrange multiplier. The appropriate value of λ is chosen so as to satisfy the constraint (6.61). When using local restrictions the problem is translated into a problem with local Lagrange multipliers

$$\min_u J(u) + \frac{1}{2} \int_\Omega \lambda(x) |Ku - g|^2 dx \tag{6.63}$$

where $\lambda(x) = \sum_{i=1}^r \lambda_i \chi_{O_i}$ and $\{O_1, ..., O_r\}$ is a partition of Ω where each boundary ∂O_i is Lipschitz. Then, the Lagrange multipliers $\lambda_i \geq 0$ are such that

$$\int_{O_i} |Ku - g|^2 dx \leq \sigma^2 |O_i| \qquad \forall\, i = 1, ..., r \tag{6.64}$$

These local restrictions are inequalities because with the equality it is not possible to guarantee that $\lambda_i \geq 0$ (see [24]).

The minimization problem with local constraints is solved with the Uzawa method. The update of parameters λ_i is then

$$\lambda_i = \max(\lambda_i + \rho(I_{O_i}(u^\lambda) - \sigma^2), 0) \qquad \text{where } \; I_{O_i}(u^\lambda) = \frac{1}{|O_i|} \int_{O_i} |Ku - g|^2 dx$$

and u^λ is the solution of the minimization problem (6.63) for fixed values of each λ_i.

Notice that in our optimization problem (6.48) the parameter λ controls the regularization term and not the term of data fidelity as in (6.62). In order to adapt our functional to the use of local constraints on the data fidelity term as in (6.64) we may modify (6.48) in the following way

$$\min_u \frac{1}{2} \int_\Omega \Gamma |(SHPFu - g)|^2 + J(u) \tag{6.65}$$

where $\Gamma = \sum_{i=1}^r \gamma_i \chi_{O_i}$. As before, $\{O_1, ..., O_r\}$ is a partition of Ω where each boundary ∂O_i is Lipschitz and the constraints act locally on each region O_i. Now, the Lagrange multipliers $\gamma_i \geq 0$ are such that

$$\int_{O_i} |SHPFu - g|^2 dx \leq \sigma^2 |O_i| \qquad \forall\, i = 1, ..., r \tag{6.66}$$

Note that problem (6.65) is essentially the same as before simply by setting $W = (\Gamma)^{1/2}$ and $\lambda = 1$ in (6.48). So the structure of the algorithm is basically the same except for the actualization of W, T' and $\hat{r}'$ each time that Γ changes (in the previous case $W = I$ and T', $\hat{r}'$ defined in (6.50) were computed only at the beginning).

As regards the numerical implementation, the principal steps are now

Algorithm 3: TV based restoration algorithm with local constraints

1. Set $\gamma_i^0 \geq 0$ small enough so that

$$I_{O_i}(u^0) = \frac{1}{|O_i|}\int_{O_i} |SHPFu^0 - g|^2 dx \geq \sigma^2 \qquad \forall\, i = 1, ..., r$$

2. Set $u^0 = 0$ or a better guess based on the irregular samples g.

3. Iterate from $m = 0$ until convergence of γ_i^m

   (a) Set $u_0 = u^m$ in Algorithm 2.

   (b) Solve $Qu_{n+1} + \tau\nabla\cdot(\xi) = b_n$ using Algorithm 2 with $\lambda = 1$ and $W = (\Gamma)^{1/2}$.

   (c) Set $u^{m+1} = u_{n^*}$, where n^* is the last time step in (b).

   (d) Update $\gamma_i\ \forall\ i = 1, ..., r$ in the following way:

$$\gamma_i^{m+1} = \max\left(\gamma_i^m + \rho(I_{O_i}(u^{m+1}) - \sigma^2), 0\right)$$

In practice, step 3.b is performed by solving 10 iterations of n in Algorithm 2 to reduce computations without observing great differences in the global convergence.

In order to get a partition $\{O_1, ..., O_r\}$ of Ω we use the Mumford-Shah segmentation algorithm [113] applied on the image g and stopping it to ensure that regions O_i have an area of at least 100 pixels.

6.6 Zoom

The increase of resolution (zoom) of the image g already introduced in section 6.4 with the use of a spectral projector P, is analyzed with more detail in this section.

Let us denote z the zoom factor, then the zoomed restored image u is a vector of size $N \times N$ where $N = z(2M + 1)$ (we recall that the size of g is $(2M + 1) \times (2M + 1)$). The minimization problem (6.48) with $\hat{p}_d = \chi_{[-M,M]^2}$ (we denote $\hat{p}_d$ the discrete version of $\hat{p}$) is a direct extension of the oversampling and denoising method introduced by Malgouyres and Guichard [105, 106] to the more general case of regular sampling, deconvolution, denoising and oversampling. The basic idea is to fit (as possible) the low frequency

components of the restored and zoomed image to the original data and to extrapolate the spectrum to the rest of the frequency domain by means of the total variation. The advantage of this regularization is that it allows to reconstruct some high frequencies which indeed is much more convenient than just filling them with zeros, a technique which is known to produce strong ringing.

In [106] it is proved that the pure zoom problem (without denoising) admits a solution and the uniqueness is only assured in a weak sense: different solutions differ only on a change of contrast. The existence of solutions in the denoising extension is shown in [105]. The only necessary assumption is that $\hat{p}_d = 0$ in the frequency region $(-N/2, N/2]^2 \backslash [-M, M]^2$. Thus, another possibility is to use a prolate function with support $[-M, M]^2$ instead of the ideal window $\hat{p}_d = \chi_{[-M,M]^2}$. Let us call $\hat{p}_s$ the prolate function (a smooth version of $\hat{p}$) and $P_s = \mathrm{diag}(\{\hat{p}_s(\frac{2\pi k}{2M+1})\}_{k=(-M,-M)}^{(M,M)})$ its associated operator. Then two different approaches for the data fidelity term arise:

a) Consider a "smooth low-pass" version of u (as before but P_s instead of P):

$$\|W(SHP_sFu - g)\|^2$$

b) Consider a "smooth low-pass" version of the whole data fidelity term:

$$\|WS(HP_sFu - P_sS^*g)\|^2 \tag{6.67}$$

Remember that S^* is an operator that computes the Fourier coefficients of an irregular sampled image. Indeed, if we denote $p_s(x) = F^{-1}(\hat{p}_s(\xi))$, the operations on g in (6.67) are consistent since

$$\begin{aligned} F((g * p_s)(x)) &= F\left(\sum_{\lambda_j \in \Lambda} g_{\lambda_j} \delta_{\lambda_j} * p_s(x)\right) = F\left(\sum_{\lambda_j \in \Lambda} g_{\lambda_j} p_s(x - \lambda_j)\right) \\ &= \sum_{\lambda_j \in \Lambda} g_{\lambda_j} e^{-i2\pi\lambda_j\xi} P_s(\xi) = P_sS^*g(\xi) \end{aligned}$$

6.7 Experiments

6.7.1 Irregular to regular sampling and denoising

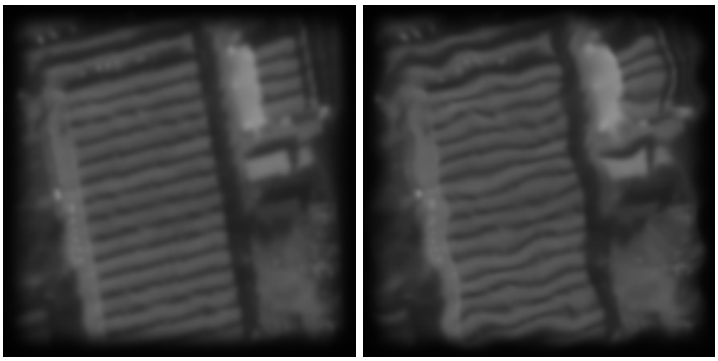
In order to test and compare the performance of the different algorithms we worked with the reference image shown in figure 6.1(a). Then we simulated different perturbations $\varepsilon(x)$, see equation (6.5), such that $\mathrm{supp}(\hat{\varepsilon}) \subseteq [-\frac{1}{2T_\varepsilon}, \frac{1}{2T_\varepsilon}]^2$ for $T_\varepsilon = 20$, and for different standard deviations A (typically 0.25, 1), see figures 6.1(c) and 6.1(d). With the

given perturbation we simulated the perturbed images g with a high precision (usually 10^{-8}) using the transposed NFFT [121] (see figure 6.1(b) for an example with $A = 1$). Finally we added some white noise to the irregular samples with standard deviation σ, for different values such as $\sigma = 0.064$ (i.e. 10^{-3} times smaller than the standard deviation of the image, $SNR = 60dB$), $\sigma = 0.64$ (i.e. 10^{-2} times smaller than the standard deviation of the image, $SNR = 40dB$), and $\sigma = 2$ gray levels. In this way, we obtained perturbed images such as the ones shown in figure 6.1.

The aim of this section is to compare the results of our algorithm with those obtained with the original ACT algorithm, i.e., without the regularization of the total variation. In both cases we have used different types of weights D in order to regularize the solution.

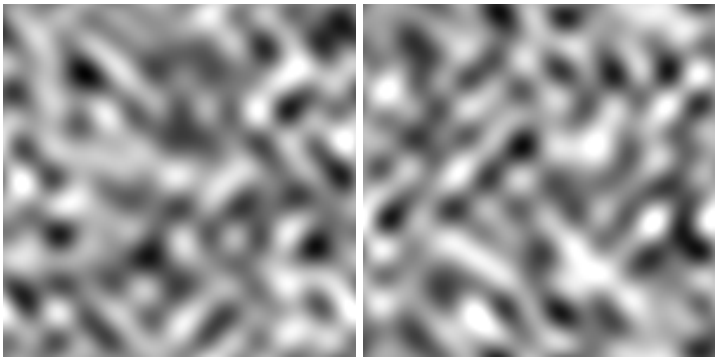
Figure 6.2 shows the evolution of the relative error (in norm L^2) along the iterations of the CG in the ACT algorithm (Algorithm 1). From the observation of the graphics in figure 6.2 we can say that the ACT algorithm attains a level of relative error which agrees with the SNR. The presence of the weights that regularize the solution and impose a specific spectral decay make the algorithm more stable. As we can observe, the algorithm without weights begins to diverge after reaching the minimum error and the rate of divergence grows as the standard deviation A of the perturbation is larger. In addition, the relative error is slightly smaller using weights and the errors in the high frequency band are decreased significantly. However, the presence of weights makes the algorithm have a poorer convergence rate. This decrease in the convergence rate when using weights is also observed in the case of the proposed algorithm which combines ACT algorithm with a total variation regularization (see figure 6.3). Notice that the horizontal axis in figure 6.3 corresponds to iterations of the CG in step 3.e. of Algorithm 2 at each time iteration n. We also notice that, in figure 6.3, the error is much more reduced at specific instants, they correspond to a change from time step n to $n + 1$. This algorithm seems to be more stable than the ACT algorithm for small perturbation amplitudes.

Figures 6.4 to 6.7 show some examples of restoration with both algorithms (ACT algorithm and TV based algorithm) for perturbed images in figures 6.1(e) and 6.1(f). They show also the root mean square errors (RMS) in space with respect to the original image 6.1(a) (the restored images always correspond to a final iteration before the algorithm begins to diverge).



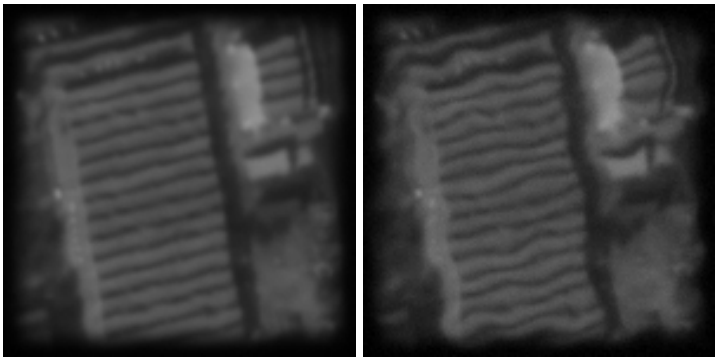
(a) original image

(b) perturbed image



(c) perturbation in the horizontal direction

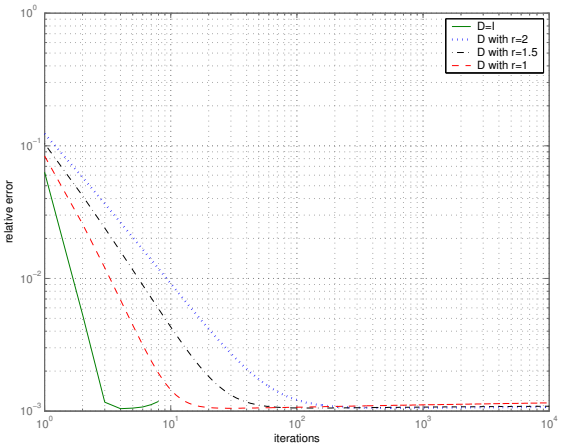
(d) perturbation in the vertical direction



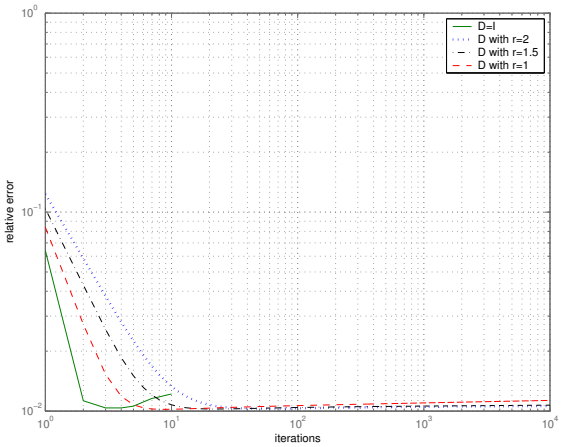
(e) perturbed and noised image with $A = 0.25$, $\sigma = 0.64$

(f) perturbed and noised image with $A = 1$, $\sigma = 2$

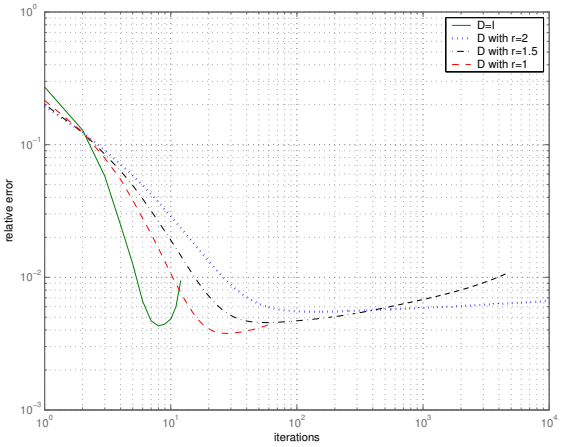
Figure 6.1: *Original image, perturbed image, perturbation and two examples of perturbed and noised images.* The original image has 149x149 pixels and was multiplied by a smooth window on the borders in order to avoid periodization artifacts. The perturbation is a colored noise with standard deviation of A pixels, and spectral contents inside $[-\frac{1}{2T_\varepsilon}, \frac{1}{2T_\varepsilon}]^2$ for $T_\varepsilon = 20$. The perturbed image has been generated taking $A = 1$. Finally, two perturbed and noised images are shown for different values of A and white noise of standard deviation σ.



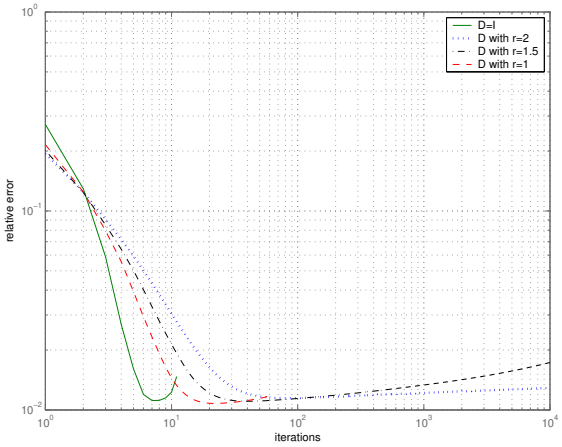
(a) $A = 0.25$, $\sigma = 0.064$



(b) $A = 0.25$, $\sigma = 0.64$

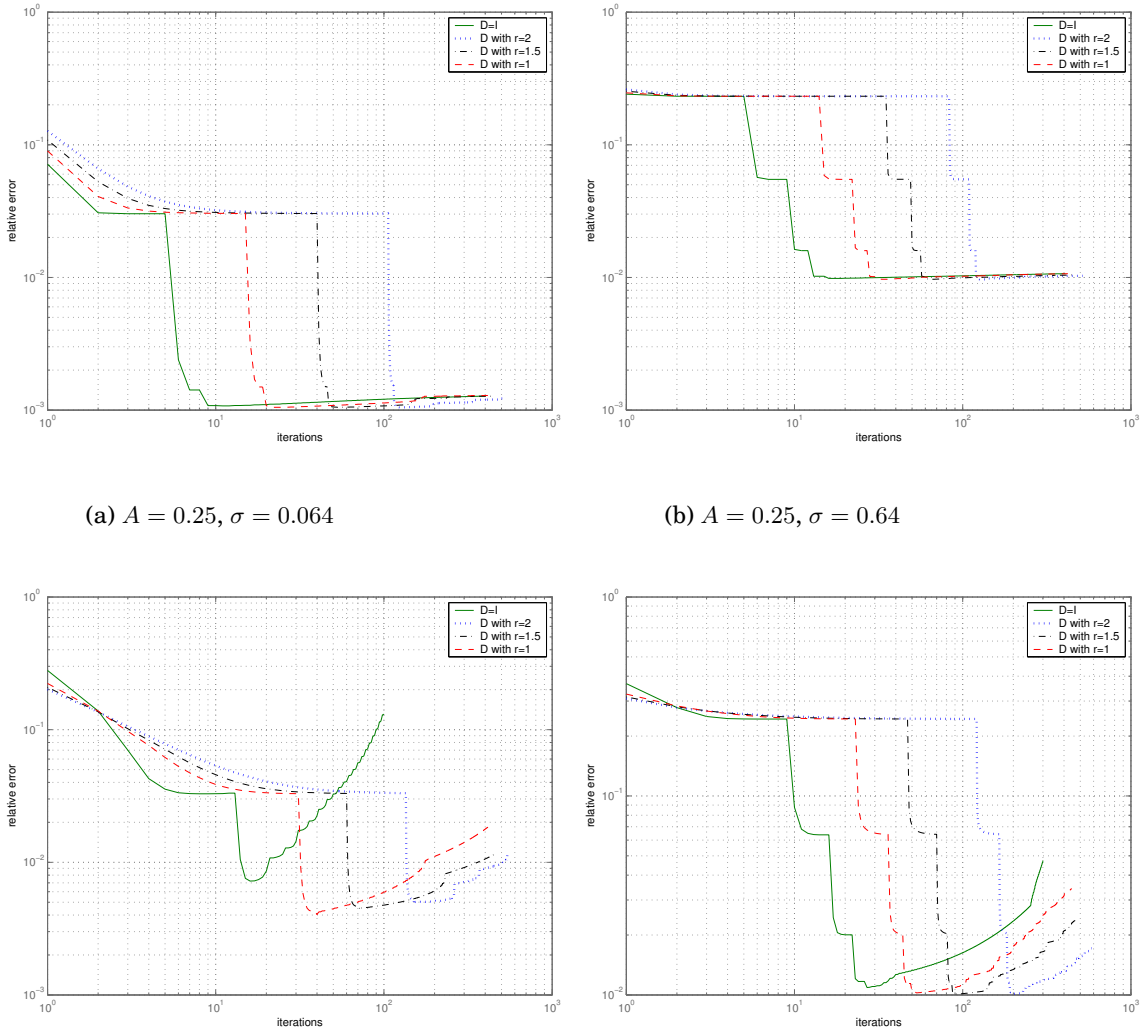


(c) $A = 1$, $\sigma = 0.064$



(d) $A = 1$, $\sigma = 0.64$

Figure 6.2: *Convergence graphics of the ACT algorithm.* We show the relative error (in norm L^2) with respect to the original image 6.1(a) along iterations of the ACT algorithm (with different type of weights) for different values of perturbation amplitude A and noise level σ. The weights impose a spectral decay $(1+\omega)^{-r}$, we test values $r = 0$ (no weights $D = I$) and $r = 1, 1.5, 2$.



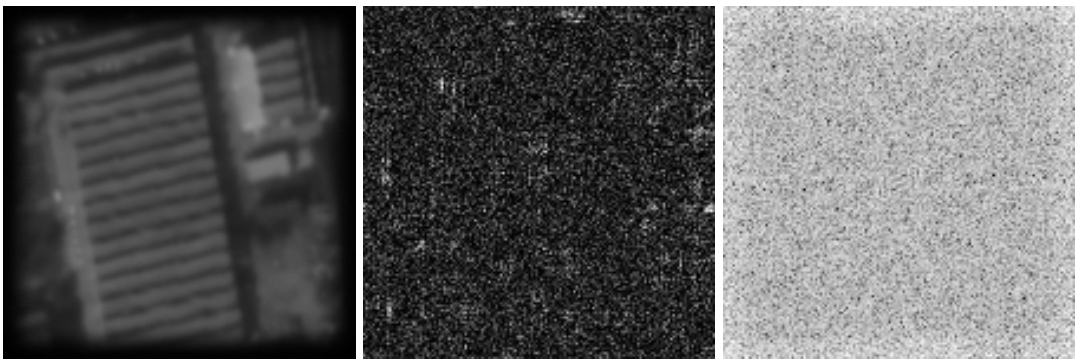
(a) $A = 0.25$, $\sigma = 0.064$

(b) $A = 0.25$, $\sigma = 0.64$

(c) $A = 1$, $\sigma = 0.064$

(d) $A = 1$, $\sigma = 0.64$

Figure 6.3: *Convergence graphics of the TV based algorithm.* We show the relative error (in norm L^2) with respect to the original image 6.1(a) along iterations of the proposed TV based algorithm (with different type of weights) for different values of perturbation amplitude A and noise level σ. The weights impose a spectral decay $(1+\omega)^{-r}$, we test values $r = 0$ (no weights $D = I$) and $r = 1, 1.5, 2$.



(a) Results with ACT algorithm (space RMS = 0.73)

(b) Results with ACT algorithm using weights D with $r = 2$ (space RMS = 0.66)

Figure 6.4: *Irregular to regular sampling with the ACT algorithm.* Perturbation amplitude $A = 0.25$. Noise level $\sigma = 0.64$. The left column shows the restored image; the middle column the error in the spatial domain; the right column the error in the Fourier domain. The first line corresponds to the result without regularization and the second one imposes a quadratic spectral decay.

(a) Results with ACT algorithm (space RMS = 2.35)

(b) Results with ACT algorithm using weights D with $r = 2$ (space RMS = 2.07)

Figure 6.5: *Irregular to regular sampling with the ACT algorithm.* Perturbation amplitude $A = 1$. Noise level $\sigma = 2$. The left column shows the restored image; the middle column the error in the spatial domain; the right column the error in the Fourier domain. The first line corresponds to the result without regularization and the second one imposes a quadratic spectral decay.

(a) Results with TV minimization (space RMS = 0.61)

(b) Results with TV minimization using weights D with $r = 2$ (space RMS = 0.61)

Figure 6.6: *Irregular to regular sampling with the TV minimization.* Perturbation amplitude $A = 0.25$. Noise level $\sigma = 0.64$. The left column shows the restored image; the middle column the error in the spatial domain; the right column the error in the Fourier domain. The first line corresponds to the result without regularization and the second one imposes a quadratic spectral decay.

(a) Results with TV minimization (space RMS = 1.53)

(b) Results with TV minimization using weights D with $r = 2$ (space RMS = 1.53)

Figure 6.7: *Irregular to regular sampling with the TV minimization.* Perturbation amplitude $A = 1$. Noise level $\sigma = 2$. The left column shows the restored image; the middle column the error in the spatial domain; the right column the error in the Fourier domain. The first line corresponds to the result without regularization and the second one imposes a quadratic spectral decay.

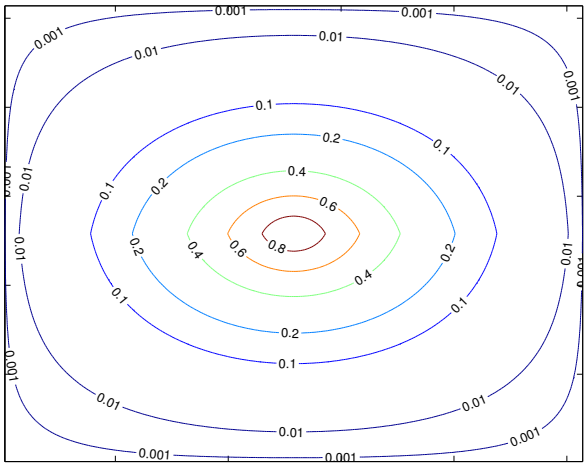


Figure 6.8: *Some level lines in* $[-1/2, 1/2]^2$ *of MTF (6.69).*

6.7.2 Adding deconvolution

Test of convergence

With the aim to test the ability of the algorithm to face the problems of irregular to regular sampling, deconvolution and denoising at the same time we make an analysis of convergence like in the previous section. In addition, we compare the results to those obtained with an extension of the ACT algorithm so as to include deconvolution.

Let us firstly explain how we extend the ACT algorithm to the deconvolution case. In the irregular to regular sampling the ACT algorithm is based on solving the linear problem (6.8) via the optimization problem (6.10). If we want to restore a filtered image the system to solve will be now:

$$g = SHa$$

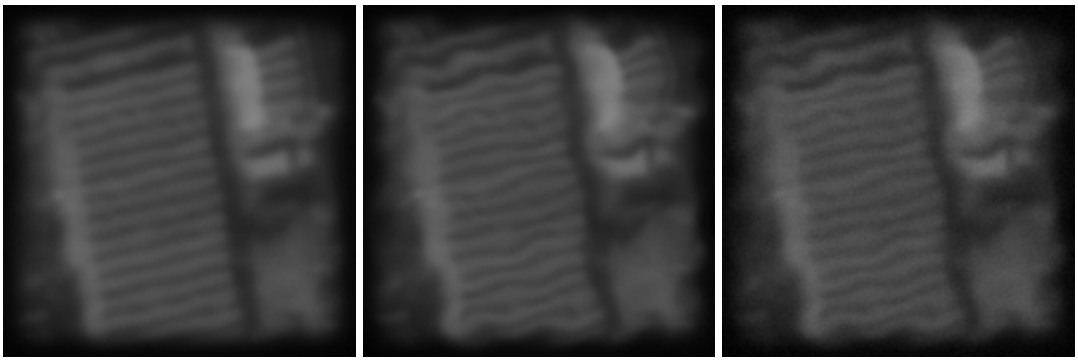
remember that $a = Fu$ are the Fourier coefficients of the restored image u. Then, like in the original ACT algorithm it is more convenient to solve the normal equations as an optimization problem:

$$\min_a \|H^*S^*WSHa - H^*S^*Wg\|^2 = \|\overline{T}a - \overline{b}\|^2 \tag{6.68}$$

where $\overline{T} = H^*TH$ is still a Toeplitz matrix and $\overline{b} = H^*b$. Note that the ACT algorithm does not need structural changes in order to include deconvolution.

We use the modulation transfer function corresponding to SPOT 5 HRG satellite with Hipermode sampling (see appendix C and [2] for more details):

$$\hat{h}(\xi,\eta) = e^{-4\pi\beta_1|\xi|}e^{-4\pi\alpha\sqrt{\xi^2+\eta^2}}\operatorname{sinc}(2\xi)\operatorname{sinc}(2\eta)\operatorname{sinc}(\xi), \qquad \xi,\eta \in [-1/2, 1/2] \tag{6.69}$$



(a) $A = 0.25$, $\sigma = 0.64$ (b) $A = 1$, $\sigma = 0.64$ (c) $A = 1$, $\sigma = 2$

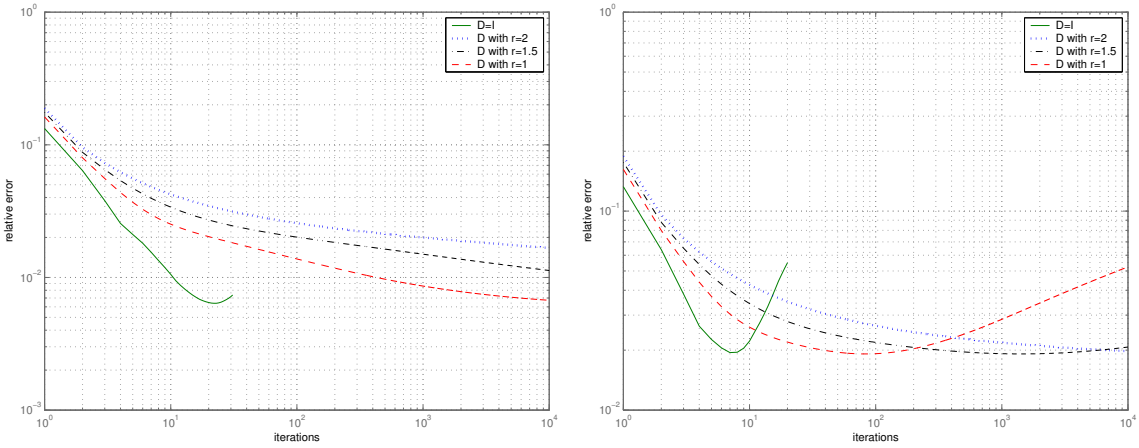
Figure 6.9: *Filtered, perturbed and noised images.* The original image is in figure 6.1(a). We filtered this image with filter (6.69). The perturbation is a colored noise with standard deviation of A pixels, and spectral contents inside $[-\frac{1}{2T_\varepsilon}, \frac{1}{2T_\varepsilon}]^2$ for $T_\varepsilon = 20$ (see horizontal and vertical perturbations in figures 6.1(c) and 6.1(d)). We added white noise of standard deviation σ.

where $\mathrm{sinc}(\xi) = \sin(\pi\xi)/(\pi\xi)$, $\alpha = 0.58$ and $\beta_1 = 0.14$. We show in figure 6.8 some level lines of this MTF. Then, we filter the original image 6.1(a) with filter (6.69) and simulate the same perturbation in sampling as in the previous section (figure 6.1(c) represents the horizontal perturbation and figure 6.1(d) the vertical one). Finally, we add some white noise of standard deviation σ to obtain images 6.9(a) ($A = 0.25$, $\sigma = 0.64$), 6.9(b) ($A = 1$, $\sigma = 0.64$) and 6.9(c) ($A = 1$, $\sigma = 2$).

The relative errors (in norm L^2) along iterations for both algorithms can be seen in figures 6.10 and 6.11. From these results we can conclude that the TV based algorithm is always (for small and large values of A) stable in the deconvolution case, unlike the ACT algorithm. Note how the use of weights does not improve the solution, still, it reduces the rate of convergence. Some restored images and their corresponding errors in space and frequency for both algorithms are shown in figures 6.12 and 6.13.

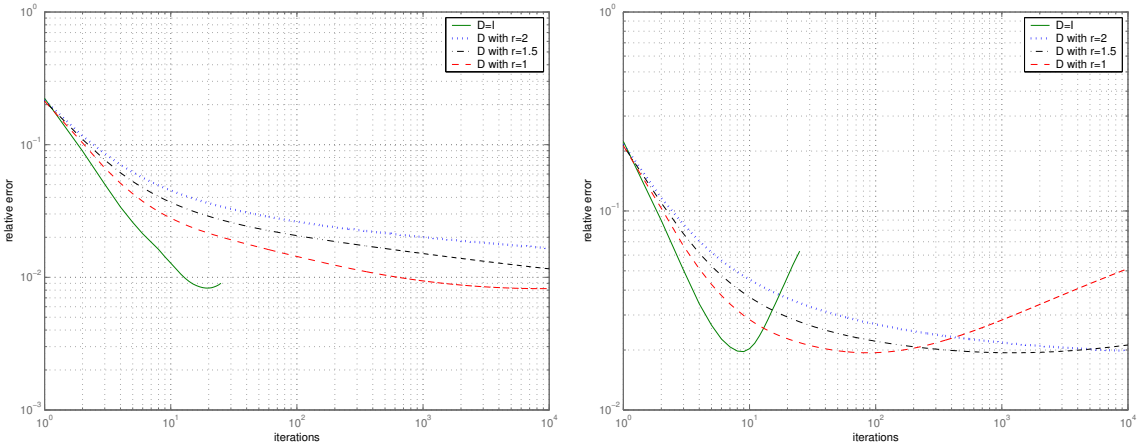
Global regularization parameter

In this section we show more examples of the algorithm containing deconvolution. For that we use the image in figure 6.14(a) and we apply a degradation with the modulation transfer function (6.69). Then we apply a perturbation in sampling which has the same characteristics as the one used in previous examples ($T_\varepsilon = 20$) and we add white noise of standard deviation $\sigma = 1$. Figure 6.14(b) is an example of a small perturbation



(a) $A = 0.25$, $\sigma = 0.064$

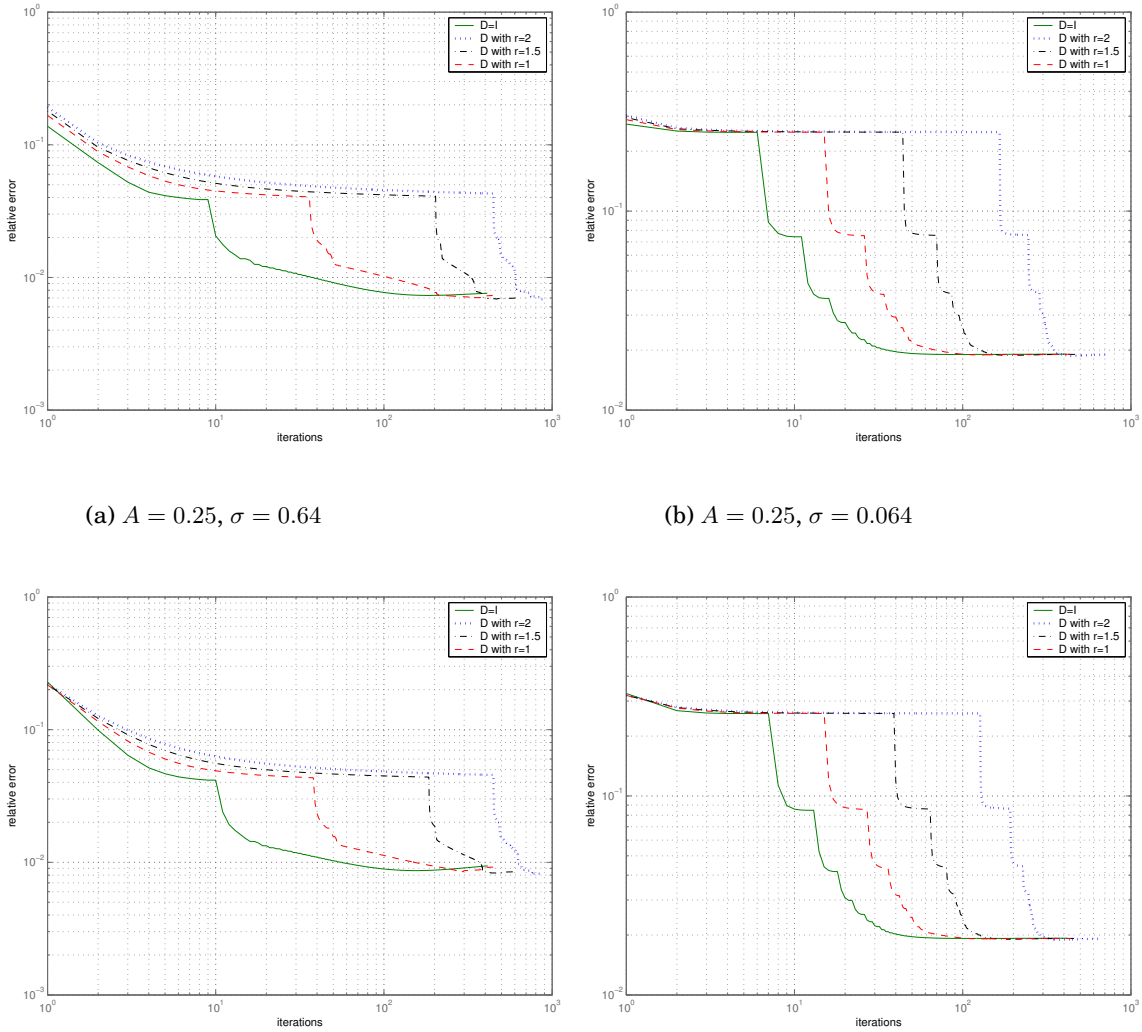
(b) $A = 0.25$, $\sigma = 0.64$



(c) $A = 1$, $\sigma = 0.064$

(d) $A = 1$, $\sigma = 0.64$

Figure 6.10: *Convergence graphics of the ACT algorithm extended to include deconvolution.* We show the relative error (in norm L^2) with respect to the original image 6.1(a) along iterations of the ACT algorithm with deconvolution (with different type of weights) for different values of perturbation amplitude A and noise level σ. The weights impose a spectral decay $(1+\omega)^{-r}$, we test values $r = 0$ (no weights $D = I$) and $r = 1, 1.5, 2$.



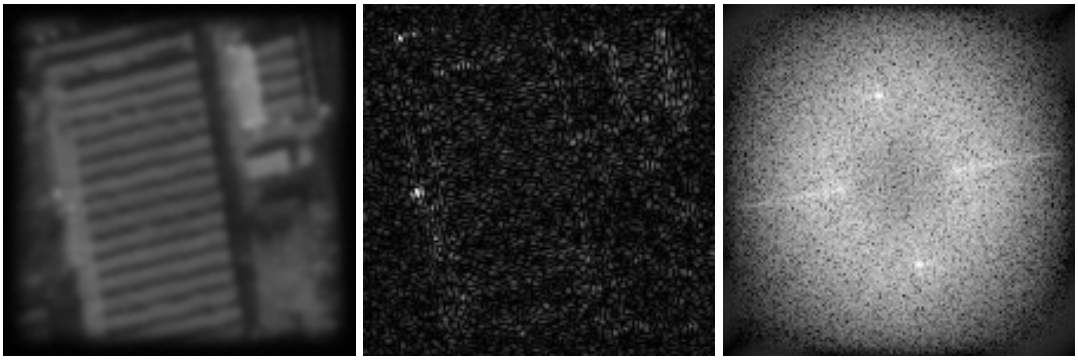
(a) $A = 0.25$, $\sigma = 0.64$

(b) $A = 0.25$, $\sigma = 0.064$

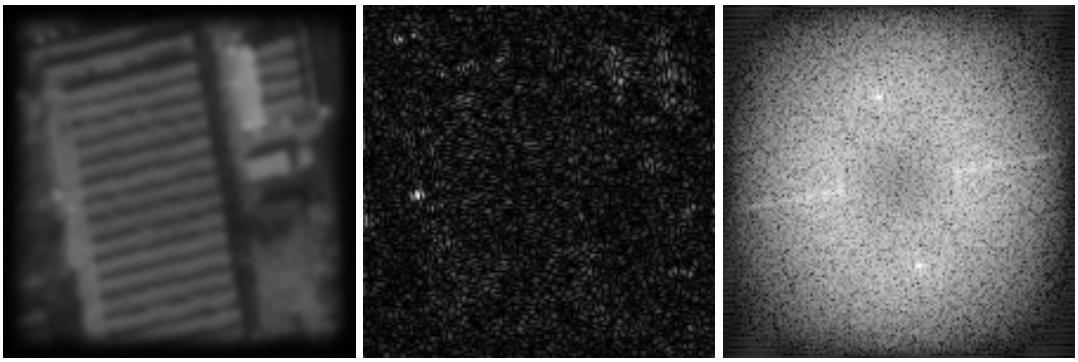
(c) $A = 1$, $\sigma = 0.64$

(d) $A = 1$, $\sigma = 0.064$

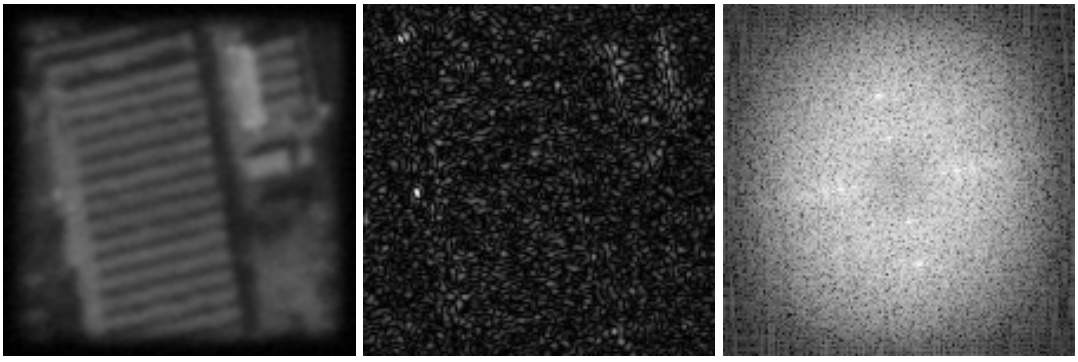
Figure 6.11: *Convergence graphics of the TV based algorithm (deconvolution case).* We show the relative error (in norm L^2) with respect to the original image 6.1(a) along iterations of the proposed TV based algorithm (with different type of weights) for different values of perturbation amplitude A and noise level σ. The weights impose a spectral decay $(1+\omega)^{-r}$, we test values $r = 0$ (no weights $D = I$) and $r = 1, 1.5, 2$.



(a) $A = 0.25$ and $\sigma = 0.64$ (space RMS = 1.01)

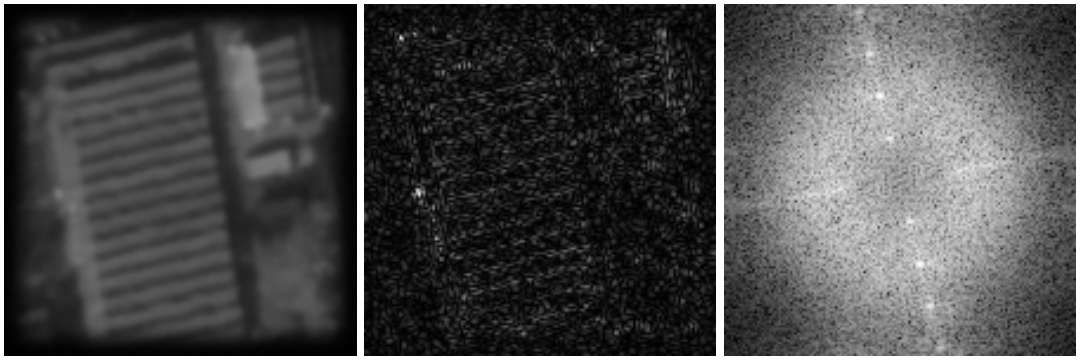


(b) $A = 1$ and $\sigma = 0.64$ (space RMS = 1.02)

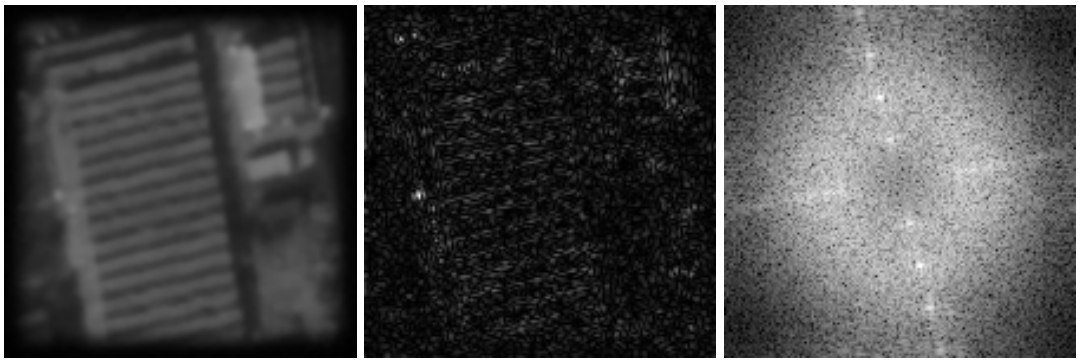


(c) $A = 1$ and $\sigma = 2$ (space RMS = 2.02)

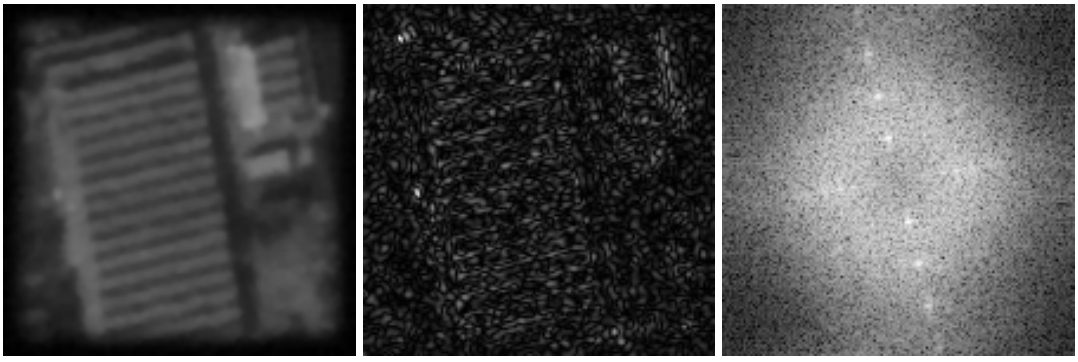
Figure 6.12: *Irregular to regular sampling, deconvolution and denoising with the ACT algorithm.* Restored images and errors for different perturbation amplitudes A and noise levels σ. The left column shows the restored image; the middle column the error in the spatial domain; the right column the error in the Fourier domain. All the images are restored with regularization weights imposing a spectral quadratic decay (D with $r = 2$).



(a) $A = 0.25$ and $\sigma = 0.64$ (space RMS = 0.98)



(b) $A = 1$ and $\sigma = 0.64$ (space RMS = 0.99)



(c) $A = 1$ and $\sigma = 2$ (space RMS = 1.89)

Figure 6.13: *Irregular to regular sampling, deconvolution and denoising with the TV.* Restored images and errors for different perturbation amplitudes A and noise levels σ. The left column shows the restored image; the middle column the error in the spatial domain; the right column the error in the Fourier domain. All the images are restored without regularization by weights ($D = I$).

($A = 0.25$) and figure 6.14(c) is generated with a larger perturbation ($A = 1$).

Some results of the TV based algorithm for different values of the regularization parameter λ are shown in figure 6.15. These examples put in evidence the drawbacks of using a global regularization parameter. In particular, notice how textures are lost for a large λ and how much noise is present for a smaller λ. In the next subsection we will see how these problems are solved using local parameters. We have also tested in figure 6.16 the ACT algorithm extended to the deconvolution case with regularization weights imposing a spectral quadratic decay. Notice how the root mean square error (RMS, measured in space) is larger than in the TV based algorithm and that visually the restored images are not as good as the ones in figure 6.15 (obtained with the TV based algorithm).

Local parameters

We can see in figure 6.17 the restored images corresponding to perturbed images 6.14(b) and 6.14(c). For that, we have used Algorithm 3, i.e. the one which uses local data fidelity parameters. Now the textures in buildings are recovered and noise is not so evident as in the case with a small global parameter. Note also how the RMS errors are reduced with respect to the global parameter case (figure 6.15) and especially with respect to the ACT algorithm (figure 6.16).

In order to test the performance of the local constraints we have measured in three regions O_i the integrals

$$I_{O_i}(u^p) = \frac{1}{|O_i|} \int_{O_i} |W(SHPFu^p - g)|^2 dx$$

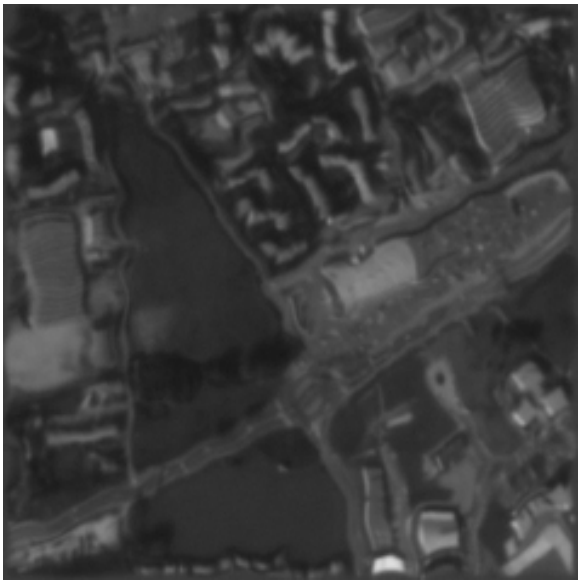
along the iterations m and n of Algorithm 3, i.e. $p = n(m)$ because Algorithm 3 iterates in n at each step m and we measure I_{O_i} along the iterations n for every step m in the update of γ_i^m. The three regions O_i for $i = 1, 2, 3$ are shown in figure 6.18(b). The segmentation of image 6.14(b) can be seen in figure 6.18(a). Figure 6.19 contains the three graphics corresponding to each marked region in figure 6.18(b). The local constraint $I_{O_i} \leq \sigma^2$, see also (6.66), in each of the three regions is verified and the convergence is quite fast.

6.7.3 Adding zoom

Finally we show some examples with the full restoration model: irregular to regular sampling, deconvolution, zoom and denoising. Zoom is added by using a projector P as described in section 6.6 (in experiments we use the ideal window $\hat{p}$ instead of the prolate function $\hat{p}_s$). We test the zoom with images in figure 6.14.

(a) original image

(b) filtered, perturbed and noised image with $A = 0.25$, $\sigma = 1$



(c) filtered, perturbed and noised image with $A = 1$, $\sigma = 1$

Figure 6.14: *Original image and some filtered, perturbed and noised images.* The original image has 237x237 pixels. The perturbation is a colored noise with standard deviation of A pixels, and spectral contents inside $[-\frac{1}{2T_\varepsilon}, \frac{1}{2T_\varepsilon}]^2$ for $T_\varepsilon = 20$. Two filtered, perturbed and noised images are showed for different values of A and white noise of standard deviation $\sigma = 1$

(a) $A = 0.25$, restored with $\lambda = 0.5$ (RMS = 9.79)

(b) $A = 0.25$, restored with $\lambda = 0.01$ (RMS = 7.65)

(c) $A = 1$, restored with $\lambda = 0.5$ (RMS = 9.86)

(d) $A = 1$, restored with $\lambda = 0.01$ (RMS = 7.79)

Figure 6.15: *Restored images with a global TV regularization parameter.* The first line contains two restoration examples of figure 6.14(b) for two different values of λ. The second line contains two more examples of restoring figure 6.14(c) also for two different values of λ. Notice how noise is reduced and textures (high frequencies) are lost if we use a large λ.

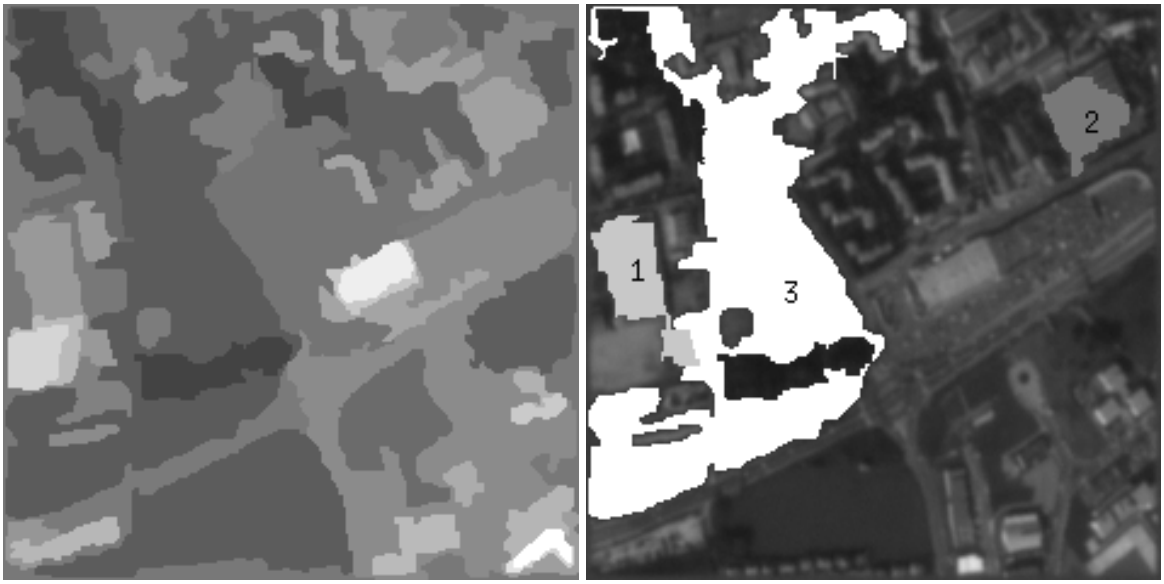
(a) $A = 0.25$ (RMS = 10.19) (b) $A = 1$ (RMS = 10.16)

Figure 6.16: *Restored images with the ACT algorithm extended to the deconvolution case.* The left image contains the restoration of figure 6.14(b). The right image contains the restoration of figure 6.14(c). Both of them use regularization weights imposing a spectral quadratic decay (D with $r = 2$).

(a) restoration of image 6.14(b) where $A = 0.25$ (space RMS = 7.18)

(b) restoration of image 6.14(c) where $A = 1$ (space RMS = 7.36)

Figure 6.17: *Restored images with local data fidelity parameters.* The left example is a restoration of image in figure 6.14(b). The right one restores image in figure 6.14(c).



(a) segmentation of image 6.14(b)

(b) image 6.14(b) with the three segmented regions to study

Figure 6.18: *Segmented image and the three regions under study.* The image 6.14(b) has been segmented using the Mumford-Shah functional on the image 6.14(b) filtered with a grain filter to ensure regions with area greater than 100 pixels.

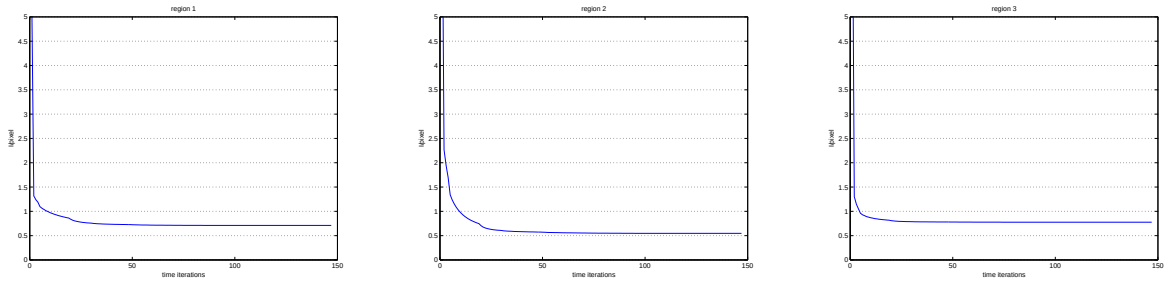


Figure 6.19: *Convergence along iterations of I_{O_i} for three different regions.* Regions under study are shown in figure 6.18(b) and I_{O_i} corresponds to the restored image 6.17(a).

Global regularization parameter

If we restore images 6.14(b) and 6.14(c) with Algorithm 2, i.e. with a global constraint ($\lambda = 0.5$) we obtain the images in figure 6.20.

Local parameters

The restoration applied to images 6.14(b) and 6.14(c) as before but now using local restrictions (Algorithm 3) results in images in figure 6.21. We have measured the evolution of I_{O_i} in the same three regions (see figure 6.18(b)) as in the previous section. The three evolution graphics are in figure 6.22, as we can observe, local constraints (6.66) are satisfied also in the zoom case.

6.8 Conclusions

We have proposed a general algorithm to solve the image restoration problem considering several different aspects of it, namely: irregular sampling, denoising, deconvolution, and also antialiasing and zoom. Our algorithm is based on an extension of the algorithm proposed by Chambolle [40] (see also [20]) for Total Variation based image denoising, combined with irregular to regular sampling algorithms proposed by Gröchenig and his coauthors [67, 80].

As regards the irregular to regular sampling and denoising problem, we have compared the results of the ACT algorithm of Gröchenig et al. with those obtained with our proposed model. The ACT algorithm makes use of Fourier weights in order to regularize and stabilize the solution. In the TV based algorithm the regularization is done by the Total Variation and so the inclusion of the same kind of Fourier weights does not really benefit the algorithm. Whereas both approaches manage to control the artifacts on high frequencies, TV minimization was found to be a better denoiser, and better reduce errors for fair noise levels and perturbation amplitudes. In particular, the error measures behave better. However, the RMS errors of both algorithms are comparable in the case of small perturbations and small noise levels if we use the regularized version of the ACT algorithm.

To test the proposed algorithm in the case of deconvolution (with irregular to regular sampling and denoising) we have compared it with a direct extension of the ACT algorithm that includes also deconvolution. We can conclude that the TV based algorithm is more stable in this case. On the other hand, the ACT algorithm starts to diverge after

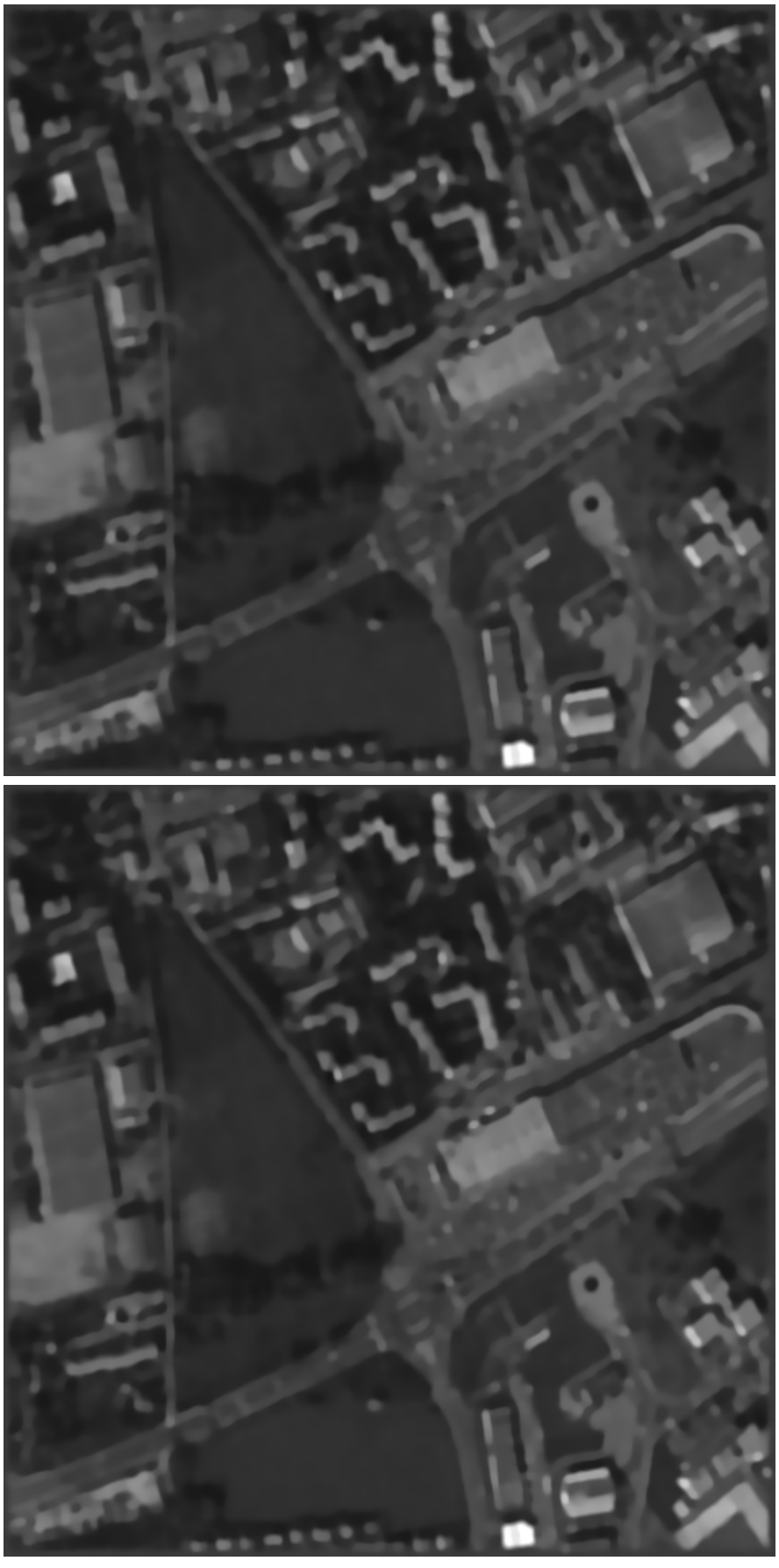


Figure 6.20: *Restored and zoomed images with a global TV regularization parameter.* The top example is a zoomed restoration of image in figure 6.14(b). The bottom one restores image in figure 6.14(c). Both of them use $\lambda = 0.5$.

Figure 6.21: *Restored and zoomed images with local data fidelity parameters.* The top example is a zoomed restoration of image in figure 6.14(b). The bottom one restores image in figure 6.14(c).

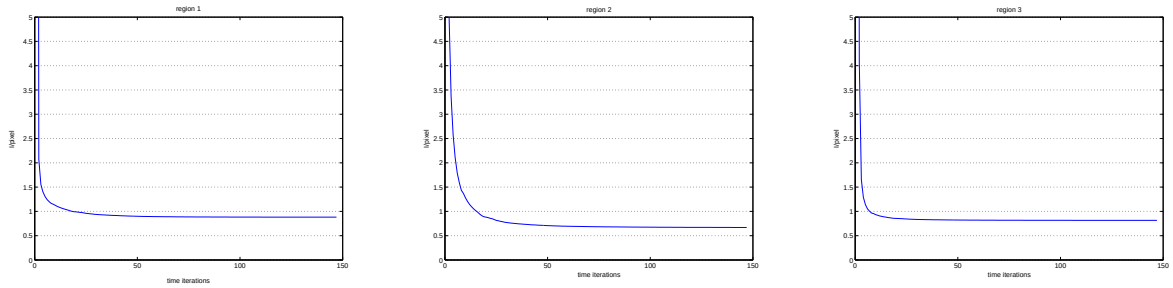


Figure 6.22: *Convergence along iterations of I_{O_i} for three different regions.* Regions under study are shown in figure 6.18(b) and I_{O_i} corresponds to the restored image on top of figure 6.21.

reaching the minimum so it is very important to choose a good stopping criterion. In addition, the root mean square errors are smaller with the TV based algorithm, especially if we use local constraints.

The first proposed TV based algorithm has a global TV regularization parameter: λ. We have seen that for large values of λ the image is quite well denoised but high frequencies or textures are lost. In contrast, for smaller values of λ textures are recovered but the noise is very evident. Thus, we use the technique introduced in [24] that uses local constraints on the data fidelity term and solves the above mentioned problem.

Finally, we have shown some results with the full restoration model, i.e. including also zoom. To conclude we can say that in general the TV based is more robust and more versatile in the sense that it is easily extended to the case of zoom, antialiasing and local constraints. Moreover, the regularized version of the ACT algorithm which is much more stable needs the a priori knowledge of the spectral decay of the image.

As future work, a better error estimate is needed to stop the CG iterations at the right point when inverting Q. Currently we perform many more iterations than needed, because of a bad error estimate. In the zoom case we have used only the ideal window as a projector although we proposed to use also a prolate function in two different ways. These other implementations have to be tested and compared to the one presented here in the experiments. We have always worked with a MTF restricted to $[-1/2, 1/2]^2$, when

this is not true the antialiasing has to be considered. Our algorithm can be easily extended to the antialiasing problem, it suffices to adapt the projector to the appropriate region were sampled coefficients contain a minimal amount of aliasing as proposed in [2, 3].

Appendix C

Modulation Transfer Function

Here we describe how the MTF, the Modulation Transfer Function, of a general satellite can be modeled. More details can be found in [125] and [2], moreover in [2] specific examples of MTF for different acquisition systems are shown.

Remember that the MTF, that we denote by $\hat{h}$, is the Fourier transform of the impulse response of the system. There are different parts in the acquisition system that contribute to the global transfer function (let $(\xi, \eta) \in [-1/2, 1/2]$ be the coordinates in the frequency domain):

- Sensors: every sensor has a sensitive region where all the photons that arrive are integrated. This region can be approximated by a unit square $[-c/2, c/2]^2$ where c is the distance between consecutive sensors. Its impulse response is then the convolution of two pulses, one in each spatial direction. The corresponding transfer function also includes the effect of the conductivity (diffusion of information) between neighbours sensors and modeled as an exponential decay:

$$\hat{h}_S(\xi, \eta) = \operatorname{sinc}(\xi c)\operatorname{sinc}(\eta c) e^{-2\pi\beta_1 c|\xi|} e^{-2\pi\beta_2 c|\eta|}$$

where $\operatorname{sinc}(\xi) = \sin(\pi\xi)/(\pi\xi)$.

- Optical system: it is considered as an isotropic low-pass filter

$$\hat{h}_O(\xi, \eta) = e^{-2\pi\alpha c\sqrt{\xi^2+\eta^2}}$$

- Motion: each sensor counts the number of photons that arrive to its sensitive region during a certain period of acquisition time. During the sampling time the system has moved a distance τ and so does the sensor, this produces a motion blur effect

in the motion direction (d_1, d_2):

$$\hat{h}_M(\xi, \eta) = \mathrm{sinc}(\langle(\xi, \eta), (d_1, d_2)\rangle\tau)$$

Finally, the global MTF is the result of the product of each of these intermediate transfer functions modeling the different aspects of the satellite:

$$\hat{h}(\xi, \eta) = \hat{h}_S\hat{h}_O\hat{h}_M$$

Appendix D

Bounded Variation Functions

Let us first recall the definition of functions of bounded variation. A function $u \in L^1(\Omega)$ whose partial derivatives in the sense of distributions are measures with finite total variation in Ω is called a function of bounded variation. The class of such functions will be denoted by $BV(\Omega)$. Thus $u \in BV(\Omega)$ if and only if there are Radon measures $\mu_1, \ldots, \mu_N$ defined in Ω with finite total mass in Ω and

$$\int_\Omega u D_i \varphi dx = -\int_\Omega \varphi d\mu_i$$

for all $\varphi \in C_0^\infty(\Omega)$, $i = 1, \ldots, N$. Thus the gradient of u is a vector valued measure with finite total variation

$$\| Du \| = \sup\{\int_\Omega u \, div\, \varphi \, dx : \varphi \in C_0^\infty(\Omega, \mathbb{R}^n), |\varphi(x)| \leq 1 \text{ for } x \in \Omega\}.$$

The space $BV(\Omega)$ is endowed with the norm

$$\| u \|_{BV} = \| u \|_{L^1(\Omega)} + \| Du \| .$$

For further information concerning functions of bounded variation we refer to [65] and [149].

We recall that $BV(\Omega) \subseteq L^p(\Omega)$ where $p = \frac{N}{N-1}$, if $N \geq 2$, and $p = \infty$, if $N = 1$. Moreover, the embedding is compact in $L^q(\Omega)$ for any $1 \leq q < p$.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In the first part of this dissertation we have reviewed some theoretical aspects concerning the shallow water equations. Then we have extended an existing numerical method [66] for hyperbolic systems of conservation laws, i.e. homogeneous systems, to the more general case of non homogeneous systems. The interest of this scheme is that it uses the Marquina's flux splitting technique [59], originally proposed and studied in the homogeneous case and whose potential in gas dynamic applications has been proved experimentally. In particular, we extend this scheme in order to include the source term due to the topography which is present in the shallow water equations. After some conclusions from the analysis of the extended scheme at the steady state we propose a combined algorithm that uses two flux decompositions like in the Marquina's technique when adjacent states are not close and a single decomposition – as traditionally used in shallow water applications – otherwise. In addition, we propose a special treatment of the proposed scheme at wet/dry fronts and at situations of dry bed generation. The performance of the combined scheme is evaluated with classical experiments in the shallow water literature. We also identify specific cases where the combined scheme performs better than the Marquina's technique and the one of single flux decomposition.

As a second contribution of this thesis we have introduced a digital Day for Night algorithm. This algorithm is based on physical data and visual perception experimental data that explains the modification in perception at low light conditions, like at night, with respect to better illumination conditions as happens at daytime. In order to simulate the loss of visual acuity at those situations of poor illumination we introduce a

novel diffusion equation, which is well-posed, has existence and uniqueness results, and is also monotonicity preserving, so no ringing may occur. The robustness of the equation makes it suitable for video sequences.

The third part deals with some geometric approaches for inpainting surface holes. First, we improve with an automatic initialization the variational problem presented in [139]. Then we propose other methods based on the mean curvature. One of them is based on a variational model which integrates a power of the modulus of the Laplacian of a distance function (when we use the power two we are dealing with the Willmore functional). The other one is heuristic and is based on the diffusion of the mean curvature of the level sets of an implicit function which is then reconstructed with the prescribed diffused curvature. Finally, we also study simpler interpolation methods based on the Laplace equation and the Absolutely Minimizing Lipschitz Extension (AMLE).

We introduce in the last part a variational model to restore satellite images and which considers the following aspects of the acquisition system model: noise, modulation transfer function (MTF) of the system, and sampling process (regular or irregular). We concentrate on the special case of irregular sampling. The resolution of the variational problem is done combining and extending the technique proposed by Chambolle [40] for denoising images and the algorithm of Gröchenig and Strohmer [80] that recovers a uniform sampling from irregular samples. The restoration method that we propose manages to do irregular to regular sampling, denoising, deconvolution and zoom at the same time. We also comparethe restoration results with those obtained with the algorithm of Gröchenig and Strohmer.

7.2 Future Work

Most of the approaches here introduced can be still improved. Moreover, they generate new perspectives of work and have other possible applications. We outline the future plans for each of the subjects involved in this thesis:

PART I: Numerical simulation of Shallow Water Equations

- Go on with the validation of the 2D version of the proposed scheme with more experiments.

- Simulation of water avalanches or dam-breaks over dry beds with variable topography.

- In [82] we studied some erosion and sedimentation models which contained the shallow water equations. The numerical scheme here proposed can be extended to correctly discretize these models.

Part II: Day for night

- Use a more complete reflexion model in the first step when estimating the reflectance values.
- Include the simulation of artificial lights.
- Emulate aspects of the film developing process.

PART III: Inpainting Surface Holes

- Develop a different numerical approximation of the variational problem based on the Laplacian of a distance function.
- Study the problem of video inpainting.

PART IV: Restoration of irregularly sampled images

- Implement and test the zoom approach with the prolate function with the two different approaches proposed in section 6.6
- Consider the more general case of a MTF not restricted to $[-1/2, 1/2]^2$ and include antialiasing techniques.

Bibliography

[1] A. Aldroubi and K. Gröchenig. Non-uniform sampling and reconstructionin shift-invariant spaces. *SIAM Rev.*, 2001. 125, 127

[2] A. Almansa. *Echantillonnage, Interpolation et Détéction. Applications en Imagerie Satellitaire*. PhD thesis, Ecole Normale Supérieure de Cachan, 94235 Cachan cedex, France, December 2002. 123, 125, 126, 153, 169, 171

[3] A. Almansa, S. Durand, and B. Rouge. Measuring and improving image resolution by adaptation of the reciprocal cell. *Journal of Mathematical Imaging and Vision*, 21(3):235–279, November 2004. 123, 169

[4] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel. Axioms and fundamental equations of image processing. *Archive Rat. Mech. and Anal.*, pages 200–257, 1993. 74

[5] L. Ambrosio, N. Fusco, and D. Pallara. Functions of Bounded Variation and Free Discontinuity Problems. *Oxford Mathematical Monographs*, 2000. 101

[6] L. Ambrosio and S. Masnou. A direct variational approach to a problem arising in image reconstruction. *Interfaces and Free Boundaries*, 5:63–81, 2003. 95

[7] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH 1998*, pages 415–421, 1998. 92

[8] Amira. Amira Visualization and Modeling System. http://www.AmiraVis.com. 110

[9] F. Andreu-Vaillo, V. Caselles, and J.M. Mazón. *Parabolic Quasilinear Equations Minimizing Linear Growth Functionals*. Birkhauser Verlag, 2004. 124

[10] G. Aronsson. Extension of functions satisfying Lipschitz conditions. *Ark. for Math.*, 6:551–561, 1967. 104, 105, 106

[11] G. Aronsson. On the partial differential equation $u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy} = 0$. *Ark. for Math.*, 7:395–425, 1968. 104, 105, 106

[12] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic projection for shallow water flows. *SIAM J. Sci. Comput.*, 25:2050–2065, 2004. 26, 50, 51

[13] P. Azmi and F. Marvasti. Comparison between several iterative methods of recovering signals from nonuniformly spaced samples. *International Journal of Sampling Theory in Signal and Image Processing*, 1(3):207–224, 2002. 125

[14] C.L. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. In *Proc. ACM SIGGRAPH 1995*, pages 109–118, 1995. 92

[15] C. Ballester, M. Bertalmío, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and grey levels. *IEEE Trans. Image Processing*, 10:1200–1211, 2001. 92, 93, 95, 96, 97, 98, 111

[16] C. Ballester, V. Caselles, and J. Verdera. Dissoclusion by joint interpolation of vector fields and gray levels. *Multiscale Model. Simul.*, 2(1):80–123, 2003. 92, 93, 95, 96, 97, 98, 111

[17] T. Battacharya, E. DiBenedetto, and J. Manfredi. Limits as $p \to \infty$ of $\delta_p u_p = f$ and related extremal problems. In *Rendiconti Sem. Mat. Fascicolo Speciale NonLinear PDEs*, pages 15–68. Univ. di Torino, 1989. 106

[18] Ph. Bénilan and M.G. Crandall. The continuous dependence on φ of solutions of $u_t - \delta\varphi(u) = 0$. *Indiana Math. J.*, 30(2):161–177, 1981. 75

[19] A. Bermúdez and Vázquez M. E. Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids*, 23(8):1049–1071, 1994. v, ix, 26, 27, 29, 36

[20] A. Bermúdez and C. Moreno. Duality methods for solving variational inequalities. *Comput. Math. Appl.*, 7(1):43–58, 1981. 131, 165

[21] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 92

[22] M. Bertalmío, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR), Hawaii*, pages 355–362, December 2001. 92, 93

[23] M. Bertalmío, V. Caselles, G. Haro, and G. Sapiro. *Mathematical Models in Computer Vision: The HandBook*, chapter PDE-Based Image and Surface Inpainting, pages 33–61. Springer, 2005.

[24] M. Bertalmío, V. Caselles, B. Rougé, and A. Solé. TV based image restoration with local constraints. *Journal of Scientific Computing*, 19(1-3):95–122, 2003. 141, 142, 168

[25] M. Bertalmío, P. Fort, and D. Sánchez-Crespo. Real-time, accurate depth of field using anisotropic diffusion and programmable graphics cards. In *Proceedings of 2nd 3DPVT*. IEEE Computer Society Press, Sept. 2004. 79

[26] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Computer Graphics (SIGGRAPH 2000)*, pages 417–424, July 2000. 91, 92, 93, 95

[27] M. Bertalmío, L. Vese, G. Sapiro, and S. Osher. Simultaneous texture and structure image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003. 92, 93

[28] A. Beurling. . In L. Carlesson, P. Malliavin, J. Neuberger, and J. Wermer, editors, *The collected works of Arne Beurling*, volume 2, pages 341–365. Birkhauser, Boston, MA, 1989. 127

[29] G. Beylkin. On the fast Fourier transform of functions with singularities. *ACHA*, 2:363–381, 1995. 127, 128

[30] G. Beylkin. On applications of unequally spaced fast Fourier transform. In *Mathematical Geophysics Summer School*, Stanford, August 1998. 127

[31] P. Brufau, Vázquez M. E., and P. García-Navarro. A numerical model for the flooding and drying of irregular domains. *International Journal for Numerical Methods in Fluids*, 39:247–275, 2002. 39, 55, 57

[32] P. Brufau and P. García-Navarro. Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique. *Journal of Computational Physics*, 186:503–526, 2003. 39

[33] J. Burguete, P. García-Navarro, and R. Aliod. Numerical simulation of runoff extreme rainfall events in a mountain water catchment. *Natural Hazards and Earth System Sciences*, 2:109–117, 2002. 39

[34] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W. R. Fright, B.C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. ACM SIGGRAPH 2001*, 2001. 92

[35] V. Caselles, R. Donat, and G. Haro. Flux-gradient and source term balancing for certain high resolution shock-capturing schemes. Preprint, 2004.

[36] V. Caselles, G. Haro, G. Sapiro, and J. Verdera. On Geometric Variational Models for Inpainting Surface Holes. Preprint, 2005.

[37] V. Caselles, J.M. Morel, and C. Sbert. An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 7(3):376–386, March 1998. 102, 103, 104

[38] M.J. Castro, J.A. García-Rodríguez, J.M. González-Vida, J. Macías, C. Parés, and M.E. Vázquez-Cendón. Numerical simulation of two-layer shallow water flows through channels with irregular geometry. *Journal of Computational Physics*, 195(1):202–235, 2003. 7

[39] M. Castro Díaz, T. Chacón Rebollo, A. Domínguez Delgado, and E.D. Fernández Nieto. Well-Balanced schemes for shallow water equations with sediment transport. In *Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering*, 2004. 7

[40] A. Chambolle. An algorithm for total variation minimization and applications. *Preprint Ceremade 02040*, 2002. vii, xi, 3, 121, 124, 131, 132, 165, 176

[41] R.H. Chan, T.F. Chan, and H.M. Zhou. Continuation method for total variation denoising problems. Technical Report 95-18, University of California, Los Angeles, 1995. 124

[42] T. Chan and J. Shen. Mathematical models for local non-texture inpaintings. *SIAM J. Appl. Math.*, 62(3):1019–1043, 2001. 92, 93, 95

[43] T. Chan and J. Shen. Non-texture Inpainting by Curvature-Driven Diffusions (CDD). *J. Visual Comm. Image Rep.*, 12(4):436–449, 2001. 95

[44] T.F. Chan, G.H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20(6):1964–1977, 1999. 124

[45] T.F. Chan, S.H. Kang, and J. Shen. Euler's elastica and curvature based inpaintings. *SIAM J. Appl. Math.*, 62(2):564–592, 2002. 95

[46] T.F. Chan and P. Mulet. On the convergence of the Lagged Diffusivity Fixed Point Method in total variation image restoration. *SIAM Journal on Numerical Analysis*, 36(2):354–367, 1999. 124

[47] E. Chasseigne and J.L. Vázquez. Theory of extended solutions for fast diffusion equations in optimal classes of data. radiation from singularities. *Archive Rat. Mech. Anal.*, 164:133–187, 2002. 75

[48] Y. Chen and G. Medioni. Description of complex objects from multiple range images using an inflating balloon model. *Computer Vision and Image Understanding*, 61(3):325–334, May 1995. 92

[49] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A finite element method for surface restoration with smooth boundary conditions. *Computer Aided Geometric Design*, 21(5):427–445, 2004. 92, 96

[50] U. Clarenz, M. Rumpf, and A. Telea. Finite elements on point based surfaces. *Computers and Graphics*, 2005. to appear. 92

[51] T. N. Cornsweet and J. I. Yellott. Intensity-independent spatial summation. *Journal of Optical Society of America*, 2(10):1769–1786, 1985. 67, 73, 74

[52] N. Crnjaric-Zic, S. Vukovic, and L. Sopta. Extension of ENO and WENO schemes to one-dimensional sediment transport equations. *Computers and Fluids*, 33:31–56, 2004. 7

[53] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. ACM SIGGRAPH 1996*, volume 5, pages 303–312, 1996. 92

[54] R. Dautray and J.L. Lions. *Analyse Mathématique et Calcul Numérique pour les sciences et les techniques, vol. 5*. Masson, 1988. 133

[55] J. Davis, S. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *First International Symposium on 3D Data Processing, Visualization, and Transmission*, 2002. 92, 93, 98, 105, 111

[56] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of ACM SIGGRAPH 1997*, July 1997. 92

[57] A. I. Delis and T. Katsaounis. Relaxation schemes for the shallow water equations. *International Journal for Numerical Methods in Fluids*, 41:695–719, 2003. 26

[58] H. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces using anisotropic basis functions. In *Proceedings IEEE Int. Conference on Computer Vision 2001*, pages 606–613, July 2001. 92

[59] R. Donat and A. Marquina. Capturing Shock Reflections: An Improved Flux Formula. *Journal of Computational Physics*, 125:42–58, 1996. v, ix, 1, 17, 19, 22, 26, 27, 57, 175

[60] F. Durand and J. Dorsey. Interactive tone mapping. In *Proc. Eurographics Workshop on Rendering*. Springer Verlag, June 2000. 70, 72, 76

[61] H. Edelsbrunner and E.P. Mucke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, January 1994. 92

[62] A. A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision, Corfu, Greece*, pages 1033–1038, Sept. 1999. 91

[63] I. Ekeland and R. Temam. *Convex Analysis and variational Problems*. Studies in Advanced Math., CRC Press, 1992. 132

[64] P. Ettedgui. *Screencraft: Cinematography*. Focal Press, 1999. 70

[65] L.C. Evans and R.F. Gariepy. *Measure Theory and Fine Properties of Functions*. 2000. 123, 173

[66] R. P. Fedkiw, B. Merriman, R. Donat, and S. Osher. The Penultimate Scheme for Systems of Conservation Laws: Finite Difference ENO with Marquina's Flux Splitting. *Progress in Numerical Solutions of Partial Differential Equations*, Arcachon, France, edited by M. Hafez, July 1998. v, ix, 18, 19, 20, 22, 23, 27, 28, 29, 33, 38, 57, 175

[67] H. G. Feichtinger, K. Gröchenig, and T. Strohmer. Efficient numerical methods in non-uniform sampling theory. *Numerische Mathematik*, 69:423–440, 1995. 121, 124, 125, 127, 140, 165

[68] H.G. Feichtinger, C. Cenker, and M. Herrmann. Iterative algorithms in irregular sampling: A first comparison of methods. In *Conf. ICCCP'91*, pages 483–489, March 1991. 125

[69] J. A. Ferwerda, S. N. Pattanaik, P. Shirley, and D. P. Greenberg. A model of visual adaptation for realistic image synthesis. In *Proceedings of SIGGRAPH 1996*, pages 249–258. ACM Press / ACM SIGGRAPH, 1996. 67, 70, 71, 72, 77

[70] T. Gallouët, J. M. Hérard, and N. Seguin. Some approximate Godunov schemes to compute shallow-water equations with topography. *Computers and Fluids*, 32:479–513, 2003. 39, 50, 53

[71] Ll. Gascón and J. M. Corberán. Construction of Second-Order TVD schemes for nonhomogeneous hyperbolic conservation laws. *Journal of Computational Physics*, 172:261–297, 2001. 27, 28, 29, 57

[72] E. Godlewski and P.A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer, 1996. 9, 18

[73] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition edition, 1996. isbn 0801854148. 129

[74] J.M. González-Vida. *Desarrollo de esquemas numéricos para el tratamiento de frentes seco-mojado en sistemas de aguas someras*. PhD thesis, Universidad de Málaga, 2003. 7

[75] L. Gosse. A well-balanced flux-vector splitting scheme designed for hyperbolic systems of conservation laws with source terms. *Int. Journal of Computers and Mathematics*, 39:135–159, 2000. 26

[76] N. Goutal and F. Maurel. In *Proceedings of the 2nd Workshop on Dam-Break Wave Simulation*. EDF-DER Report HE-43/97/016/B, 1997. 43, 45

[77] J. M. Greenberg and A. Y. LeRoux. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM J. Numerical Analysis*, 33:1–16, 1996. 26, 36

[78] K. Gröchenig. Reconstruction algorithms in irregular sampling. *Math. Comp.*, 59(181–1924), 1992. 129

[79] K. Gröchenig. *Foundations of Time-Frequency Analysis*. Birkhäuser, 2001. 127

[80] K. Gröchenig and T. Strohmer. Numerical and theoretical aspects of non-uniform sampling of band-limited images. In F. Marvasti, editor, *Theory and Practice of Nonuniform Sampling*. Kluwer/Plenum, 2000. vii, xi, 2, 121, 124, 125, 127, 129, 130, 165, 176

[81] F. Guichard and J.M. Morel. *Image Iterative Smoothing and P.D.E.'s'*. Book in preparation, 2000. 75

[82] G. Haro. Numerical simulation of water erosion models and some physical models in image processing. Diploma of Advances Studies Degree Thesis, Universitat Pompeu Fabra, November 2003. 177

[83] G. Haro, M. Bertalmío, and V. Caselles. Visual acuity in day for night. In *Proceedings of 2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*. in conjunction with IEEE-ICCV, Nice, France, October 2003.

[84] G. Haro, M. Bertalmío, and V. Caselles. Visual acuity in day for night. Preprint, 2004.

[85] A. Harten, B. Engquist, S. Osher, and S.R. Chakracarthy. Uniformly high-order accurate non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987. 26

[86] A. Harten and S. Osher. Uniformly high-order accurate non-oscillatory schemes, I. *SIAM J. Numerical Analysis*, 24(2):279–309, 1987. 23, 24, 26

[87] D. Heeger and J. Bergen. Pyramid based texture analysis/synthesis. In *Computer Graphics SIGGRAPH 1995*, pages 229–238, July 1995. 92

[88] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. ACM SIGGRAPH 1992*, pages 71–78, 1992. 92

[89] Hunt. Light and dark adaptation in the perception of color. *J. Optical Soc. America A*, 42(3):190, 1952. 70

[90] K. Ito and K. Kunisch. An active set strategy based on the augmented lagrangian formulation for image restoration. *M2AN Math. Model. Numer. Anal.*, 33:1–21, 1999. 108

[91] H.W. Jensen, F. Durand, M.M. Stark, S. Premoze, J. Dorsey, and P. Shirley. A physically-based night sky model. In *Proceedings of SIGGRAPH 2001*. ACM Press / ACM SIGGRAPH, 2001. 66

[92] R. Jensen. Uniqueness of lipschitz extensions: Minimizing the sup norm of the gradient. *Arch. Rat. Mech. Anal.*, 123:51–74, 1993. 104, 105, 106

[93] G. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202, 1996. 26

[94] P. Juutinen. Absolutely minimizing lipschitz extensions on a metric space. *Annales Acadademiae Scientiarum Fennicae Mathematica*, 27:57–67, 2002. 106

[95] M. I. Kadec. The exact value of the Payley-Wiener constant. *Soviet Math. Doklady*, 5(559–561), 1964. 126

[96] G. Kanizsa. *Gramática de la visión*. Paidos, 1986. 94

[97] M. Kawahara and T. Umetsu. Finite element method for moving boundary problems in river flow. *International Journal for Numerical Methods in Fluids*, 6:365–386, 1986. 54, 55, 57

[98] N. Fusco L. Ambrosio and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems*. Oxford Mathematical Monographs, 2000. 123

[99] S. Osher L. Rudin and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992. 123

[100] H.J. Landau. Necessary density conditions for sampling and interpolation of entire functions. *Acta Math*, 117:37–52, 1967. 127

[101] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser Verlag, 1992. 9, 18, 42

[102] R.J. LeVeque. Balancing source terms and flux gradients in high resolution Godunov methods. *Journal of Computational Physics*, 146:346, 1998. 26, 29, 48

[103] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In *Computer Graphics SIGGRAPH 2000 Proceedings*, pages 269–276, July 2000. 92, 111, 114

[104] S. Lumet. *Making Movies*. Alfred A. Knopf, 1995. 65, 70

[105] F. Malgouyres. Total Variation based oversampling of noisy images. In *Proc. on Scale-Space 2001, Lecture Notes in Computer Science 2106*, pages 111–122, 2001. 143, 144

[106] F. Malgouyres and F. Guichard. Edge direction preserving image zooming: a mathematical and numerical analysis. *Journal on Numerical Analysis*, 39(1):1–37, 2001. 123, 143, 144

[107] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, Boston, 1998. 126

[108] L. T. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Optical Soc. America-A*, 3(10):1673–1683, 1986. 69

[109] A. Marquina and P. Mulet. A flux-split algorithm applied to conservative models for multicomponent compressible flows. *Journal of Computational Physics*, 185(1):120–138, 2003. 27

[110] S. Masnou and J.M. Morel. Level-lines based disocclusion. In *IEEE Int. Conf. Image Processing*, October 1998. 92, 93, 95

[111] P. Massey and C. B. Foltz. The spectrum of the night sky over Mount Hopkins and Kitt Peak: Changes after a decade. *Publications of the Astronomical Society of the Pacific*, 112(566), 2000. 69

[112] D. Mumford. *Algebraic Geometry and Applications*, chapter Elastica and computer vision, pages 491–506. C. Bajaj ed., Springer-Verlag, Berlin, 1994. 95

[113] D. Mumford and J. J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–684, 1989. 143

[114] J.C.C. Nitsche. Periodic surfaces that are extremal for energy functionals containing curvature functions. In *Proc. Workshop Statistical Thermodynamics and Differential Geometry of Microstructured Materials*. IMA vol. in Math. and its Appl., Springer, H.T. Davis and J.C.C. Nitsche editors, 1993. 96

[115] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation, and Depth*. Springer-Verlag, Berlin, 1993. 94

[116] J.M. Ogden, E.H. Adelson, J.R. Bergen, and P.J. Burt. Pyramid-based computer graphics. *RCA Engineer*, 30(5):4–15, 1985. 92, 93

[117] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2003. 108, 109

[118] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation. *J. Computational Physics*, 79:12–49, 1988. 109

[119] C. Parés and M. Castro. On the well-balance property of roe's method for non-conservative hyperbolic systems. Applications to shallow-water systems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 38(5):821–852, 2004. 26

[120] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE TPAMI*, 12(7):629–639, 1990. 75, 77

[121] D. Potts. NFFT. http://www.math.mu-luebeck.de/potts/nfft/, 2001. 139, 145

[122] S. Rane, G. Sapiro, and M. Bertalmio. Structure and texture filling-in of missing image blocks in wireless transmission and compression applications. *IEEE Trans. Image Processing*, 12:269–303, 2003. 92, 93

[123] M. Rauth. *Gridding geophysical potential fields from noisy scattered data*. PhD thesis, University of Vienna, May 1998. 124, 125, 127

[124] P. L. Roe. Upwind differencing schemes for hyperbolic conservation laws with source terms. In *Proceedings of Nonlinear Hyperbolic Problems, edited by C. Carasso, P. Raviart, and D. Serre, Lecture Notes in Mathematics, Springer-Verlag*, volume 1270, pages 41–51, 1986. 26, 29

[125] B. Rougé. Théorie de l'échantillonage et sattelites d'observation de la terre. Analyse de Fourier et traitement d'images. Journées X-UPS 1998. Centre de Mathématiques – Ecole Polytechnique. France., 1998. 171

[126] C. W. Shu and S. Osher. Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes II (two). *Journal of Computational Physics*, 83:32–78, 1989. 18, 19, 20, 23, 27, 28, 29, 35, 37, 54

[127] E. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proc. 5th IEEE Int'l Conf. on Image Processing*, 1998. 92

[128] D. Slater and G. Healey. Using a spectral reflectance model for the illumination-invariant recognition of local image structure. *IEEE TPAMI*, 19(10), 1997. 69

[129] P. A. Sleigh, P. H. Gaskell, M. Berzins, and N. G. Wright. An unstructured finite-volume algorithm for predicting flow in rivers and estuaries. *Computers and Fluids*, 27(4):479–508, 1998. 39

[130] J. Smoller. *Shock Waves and Reaction-Diffusion Equations*. Springer-Verlag, 1994. 9, 13, 14, 15

[131] B. Stabell and U. Stabell. Effects of rod activity on color perception with light adaptation. *Journal of Optical Society of America*, 19(7):1249–1258, 2002. 70

[132] B. Tang, G. Sapiro, and V. Caselles. Color image enhancement via chromaticity diffusion. *IEEE Transactions On Image Processing*, 10(5):701–707, 2001. 76

[133] W.B. Thompson, P. Shirley, and J.A. Ferwerda. A spatial post-processing algorithm for images of night scenes. *Journal of Graphics Tools*, 7(1):1–12, 2002. 72, 76

[134] E. F. Toro. *Shock-Capturing Methods for Free-Surface Shallow Flows*. John Wiley & Sons, Ltd, 2001. 8, 9, 16, 39, 41, 51

[135] J. Tumblin and H. Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6):42–48, 1993. 67, 68, 69, 71

[136] J. Tumblin and G. Turk. LCIS: A boundary hierarchy for detail-preserving contrast reduction. In *Siggraph 1999, Computer Graphics Proceedings*, pages 83–90, 1999. 75

[137] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proc. ACM SIGGRAPH 1994*, pages 11–318, 1994. 92

[138] J. L. Vázquez. An introduction to the mathematical theory of the porous medium equation. Shape Optimization and Free Boundaries (Montreal, PQ), pages 347–389. NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., Kluwer Acad. Publ., Dordrecht, 1992. 75

[139] J. Verdera, V. Caselles, M. Bertalmío, and G. Sapiro. Inpainting surface holes. In *IEEE International Conference on Image Processing, ICIP*, pages 903–906, Sept. 2003. Barcelona. vi, x, 2, 93, 94, 96, 97, 111, 176

[140] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996. 124

[141] S. Vukovic and L. Sopta. ENO and WENO Schemes with the Exact Conservation Property for One-Dimensional Shallow Water Equations. *Journal of Computational Physics*, 179:593–621, 2002. 26, 27, 29, 35, 37, 42

[142] G. Ward, H. Rushmeier, and C. Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Trans. on Visualization and Computer Graphics*, 3(4), 1997. 67, 70, 71, 72, 74, 76

[143] M.D. Wheeler, Y. Sato, and K. keuchi. Consensus surfaces for modeling 3D objects from multiple range images. In *Proc. IEEE Int. Conference on Computer Vision*, pages 917–924, 1998. 92

[144] R. Whitaker. A level-set approach to 3D reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998. 92

[145] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae (2nd ed.)*. John Wiley & Sons, Inc., New York, 1982. 67, 68, 70

[146] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with Poisson-based gradient field manipulation. *ACM Transactions on Graphics*, 23(3):641–648, 2004. 101

[147] D. H. Zhao, H. W. Shen, G. Q. Tabios, J. S. Lai, and W. Y. Tan. Finite-volume two-dimensional unsteady-flow model for river basins. *Journal of Hydraulic Engineering, ASCE7*, 120:863–883, 1994. 39

[148] H.K. Zhao, S. Osher, and R. Fedkiw. Fast Surface Reconstruction using the Level Set Method. In *Proc. First IEEE Workshop on Variational and Level Set Methods, in conjunction with Proc. IEEE ICCV 2001*, pages 194–202, 2001. 92

[149] W. P. Ziemer. *Weakly Differentiable Functions*. GTM 120, Springer Verlag, 1989. 123, 173