

TOPICS IN PDE-BASED IMAGE PROCESSING

by

Catherine Mareva Dupuis

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Applied and Interdisciplinary Mathematics)
in The University of Michigan
2010

Doctoral Committee:

Associate Professor Selim Esedođlu, Co-Chair
Professor Jeffrey A. Fessler, Co-Chair
Professor Joseph Conlon
Professor Robert Krasny

© Catherine Mareva Dupuis 2010
All Rights Reserved

To my parents André and Marie Claude, and my husband Richard

ACKNOWLEDGEMENTS

As I look back on the five years I spent at the University of Michigan, I realize that several times fate led me onto a better path than the one I had originally intended to follow, starting with the decision to come to the University of Michigan for graduate school, which was not my first choice. I had opted for UCLA, but a twist of fate led me to come to Michigan to be close to the person whom I would soon be united to. The beginning was not easy nor without tears, but I soon learned to enjoy the high quality of Michigan's Mathematics department, and greatly benefited from all my math classes. Another twist of fate led me to work with Selim Esedoğlu, instead of the professor I had originally intended to work with. This was probably the best thing that ever happened to me during my graduate studies at Michigan, and for this I would like to thank Peter Smereka who, given my interests, helped me discern the most suitable advisor for me. I don't think I could have picked a better advisor.

My first thanks thus go to my advisor Selim Esedoğlu who always managed to have time for me whenever I needed. Selim pushed me and encouraged me to do always more than I thought I could, and provided guidance throughout my entire doctorate. I also learned a lot from him, mostly in mathematics (he explains in an exceptionally clear and concise way), but sometimes in other domains of life as well. I actually rather enjoyed our non math related discussions. A very sincere thank you also goes to Selim's tremendous help in my job search last Fall. We spent many hours talking about different universities and departments that would potentially be interested in my research, and his feedback on

my application was very valuable. Also, I would like to thank Selim for enabling me to give a talk at the SIAM Conference on Analysis of Partial Differential Equations held in Miami in December, where I met several people I was interested in working with in my postdoctoral studies.

My second line of thanks goes to my wonderful family who always help and support me, regardless what I decide to take on. Especially during my doctorate, I had many difficult times, not all of them without tears, but my parents were always there to listen and encourage me. I give a special thank you to my Mum who spent many hours with me on the phone, sometimes listening to me cry, but who invariably managed to make me feel better in an incredible way. I also want to thank my Dad who shares my interest in the beauty of mathematics, and who manages to understand my research (sometimes better than I do!). My family is my foundation whose love of learning nurtured the drive in me to get a doctorate, and provided the support for me to do so. Thank you also to my sister Bénédicte, who visited me twice in Michigan, and whom I grew to appreciate from thousands of miles away.

To my husband Richard, thank you for being with me always, for listening to me in my difficult times during my graduate studies, and for supporting and encouraging me whenever I fell down. Thank you Richard for your patience and your unfaltering love, even through hardships and tears. I would like to especially thank you for moving to Ann Arbor after our wedding, and for traveling weekly to Chicago for meetings. I know how much it tires you, but I appreciate your sacrifice for the rest of the week that we can spend together. I am looking forward to our new life in Austin, Texas.

My time at the University of Michigan would not have been the same without my three officemates Richard, Henry and Tom. I would like to thank them for listening to me in difficult times, for encouraging me, and mostly for the nice and fun conversations we had

throughout the years. Thank you also to the people in the graduate office, Tara, Nathalie, Emily, Jessica, Warren and all the others, who provided assistance whenever I needed it.

I thank my co-advisor Jeffrey Fessler for useful discussions and for including me in his imaging group. I appreciated attending his group meetings, during which I got exposed to very interesting problems related to medical imaging applications. Thank you also to my thesis defense committee members Joseph Conlon and Robert Krasny. I would also like to thank Peter Smereka for always taking the time to meet with me whenever I needed additional advice on some important decision.

I would like to acknowledge the funding agencies, awards and fellowships which provided financial support during my graduate studies and allowed me to pursue my research uninterrupted. In particular, I thank the National Science Foundation, the Mathematics department, Rackham graduate school and the Phi Kappa Phi honor society,

Last but not least, I thank all the people that I have met throughout my life and who have made me who I am. Thank you to my grand-mother Marie-Rose who was very hard-working and never gave up, and to my grand-mother Jacqueline for being a faithful correspondent. For all the people who have passed away, and all the people of my family circle and friends, thank you for being there whenever you were needed. And finally, thank you Lord for being with me always, and for guiding my life into the best path.

TABLE OF CONTENTS

| | |
|--|---------------|
| DEDICATION | ii |
| ACKNOWLEDGEMENTS | iii |
| LIST OF FIGURES | viii |
| LIST OF TABLES | xi |
| ABSTRACT | xii |
| CHAPTER | |
| I. Introduction | 1 |
| 1.1 Image Restoration | 2 |
| 1.1.1 Gaussian Filtering | 3 |
| 1.1.2 Energy minimization | 4 |
| 1.1.3 Nonlinear Diffusion | 5 |
| 1.2 Image Segmentation | 8 |
| 1.2.1 Mumford-Shah Functional | 9 |
| 1.2.2 Ambrosio and Tortorelli’s Approximation of the Mumford-Shah Functional | 10 |
| 1.2.3 The Chan-Vese Model | 11 |
| 1.3 Numerical Methods for Curve Evolution | 13 |
| 1.3.1 Front Tracking | 13 |
| 1.3.2 Phase-Field Method | 14 |
| 1.3.3 Level Set Method | 15 |
| 1.3.4 Diffusion Generated Motion | 17 |
| 1.4 Contribution and Organization of this Dissertation | 20 |
| II. Algorithms for Area Preserving Flows and Applications | 23 |
| 2.1 Introduction | 23 |
| 2.2 Proposed schemes | 24 |
| 2.2.1 Algorithms for area preserving flows | 24 |
| 2.2.2 Numerical convergence study | 33 |
| 2.3 Application: large scale simulations of area preserving curvature motion | 37 |
| 2.3.1 Curve shortening at various area fractions | 37 |
| 2.3.2 Numerical results | 38 |
| 2.4 Application to inverse problems | 47 |
| 2.4.1 Preliminaries | 47 |
| 2.4.2 Mumford-Shah based flows and their area preserving analogues | 48 |
| 2.4.3 Algorithms | 52 |
| 2.4.4 Comparisons and applications | 61 |
| 2.5 Conclusion | 72 |

| | |
|--|------------|
| III. Coarsening in High Order, Discrete, Ill-Posed Diffusion Equations | 73 |
| 3.1 Introduction | 73 |
| 3.2 Analysis of the You-Kaveh model | 77 |
| 3.2.1 The equations | 77 |
| 3.2.2 The scheme | 78 |
| 3.2.3 Terminology and basic setup | 81 |
| 3.2.4 Stability of stationary states | 84 |
| 3.3 Upper bounds on the coarsening rate of You-Kaveh type models | 93 |
| 3.3.1 Decay relation | 94 |
| 3.3.2 Interpolation inequality | 95 |
| 3.3.3 Rigorous result in terms of the spike density | 107 |
| 3.4 Upper bounds on coarsening for the $2n$ -th order equations | 110 |
| 3.4.1 Equations and scheme | 110 |
| 3.4.2 Main result | 111 |
| 3.4.3 Decay relation and interpolation inequality | 112 |
| 3.5 Numerical evidence and discussion | 115 |
| 3.5.1 Numerical results | 115 |
| 3.5.2 Discussion | 122 |
| IV. Conclusion and Future Directions | 125 |
| 4.1 Conclusion | 125 |
| 4.2 Future Directions | 126 |
| 4.2.1 Volume preserving flows in higher dimensions | 126 |
| 4.2.2 Fourth order flow for image segmentation and inverse problems | 126 |
| 4.2.3 Second order flow for image segmentation and inverse problems | 126 |
| 4.2.4 Generalizations to all images | 127 |
| 4.2.5 Coarsening in ill-posed diffusion equations for asymptotically constant energy densities | 127 |
| 4.2.6 Convergence of an algorithm for motions with normal velocities of the form $f(\kappa)$ | 128 |
| APPENDIX | 130 |
| BIBLIOGRAPHY | 134 |

LIST OF FIGURES

| <u>Figure</u> | | |
|---------------|---|----|
| 1.1 | Time evolution (coarsening) of an image when taken as initial condition for the Perona-Malik PDE. | 7 |
| 2.1 | Evolution of three circles under area preserving curvature motion. | 36 |
| 2.2 | Comparison between the evolutions of the exact and computed radii obtained from three initial circles taken as initial condition and evolved under area preserving curvature motion. | 36 |
| 2.3 | Collision of two curves during their evolution under area preserving curvature motion. | 38 |
| 2.4 | Evolution of two configurations of droplets (10% and 40% area fraction) under area preserving curvature motion. | 42 |
| 2.5 | Comparison of two configurations of droplets obtained by computing their evolution with time step (δt) and $4\delta t$ | 43 |
| 2.6 | Computed total length $L(t)$ of the boundary of all droplets for various area fractions, and theoretical bound $t^{-\frac{1}{2}}$ | 44 |
| 2.7 | Computed number of connected components $K(t)$ in the droplets' configurations for various area fractions, and bound $\frac{1}{t}$ | 45 |
| 2.8 | Time evolution of the distribution of isoperimetric ratios computed from the evolution of an initial configuration of droplets with 50% area fraction under area preserving curvature motion. | 46 |
| 2.9 | Proportions of eccentric droplets (i.e. with isoperimetric ratio $I > 20$) in the case of 30%, 40% and 50% area fraction. | 46 |
| 2.10 | Illustration of the four-phase image model. | 48 |
| 2.11 | One dimensional signed distance function \tilde{d}_R to the interface $\{0\} \cup \{R\}$ | 56 |
| 2.12 | Profile $f_0(r) = f_0(r, \theta)$ for any $0 \leq \theta < 2\pi$ | 63 |
| 2.13 | Gray scale display of image f_0 and its surface plot. | 64 |
| 2.14 | Piecewise constant Mumford-Shah based energies for various parameter λ and comparison between the radius found by the Mumford-Shah based flow and the area preserving flow. | 66 |
| 2.15 | Simple image f_0 with the two final contours (obtained using the Mumford-Shah based flow and the area preserving flow respectively) graphed for $\lambda = 0.61$ | 67 |

| | | |
|------|--|-----|
| 2.16 | Synthetic image reconstructions comparing the loss of small scale features due to regularizations on noiseless data. | 68 |
| 2.17 | Contours computed using the Mumford-Shah based flow and the area preserving flow on noiseless data. | 69 |
| 2.18 | Synthetic image reconstructions comparing the loss of small scale features due to regularizations on noisy data. | 70 |
| 2.19 | Contours computed using the Mumford-Shah based flow and the area preserving flow on noisy data. | 71 |
| 3.1 | Time evolution (coarsening) of an image when taken as initial condition for the You-Kaveh PDE. | 75 |
| 3.2 | Functions f , R and R' with $R(x) = \frac{x}{1+x^2}$ and $f(x) = \frac{1}{2} \ln(1+x^2)$ | 81 |
| 3.3 | One dimensional evolution of the discrete in space version of the You-Kaveh PDE. | 83 |
| 3.4 | Characteristic function of the set Σ_1 with the locations of the large positive spikes in dashed. | 101 |
| 3.5 | Test function φ with the locations of the large positive spikes in dashed. | 101 |
| 3.6 | Two-dimensional You-Kaveh equation: comparison between the computed rate of decrease of the energy and the theoretical bound. | 117 |
| 3.7 | Two-dimensional You-Kaveh equation: comparison between the computed coarsening rate in number of connected components and the theoretical bound. | 118 |
| 3.8 | Three-dimensional You-Kaveh equation: comparison between the computed rate of decrease of the energy and the theoretical bound. | 118 |
| 3.9 | Three-dimensional You-Kaveh equation: comparison between the computed coarsening rate in number of connected components and the theoretical bound. | 119 |
| 3.10 | Four-dimensional You-Kaveh equation: comparison between the computed rate of decrease of the energy and the theoretical bound. | 119 |
| 3.11 | Four-dimensional You-Kaveh equation: comparison between the computed coarsening rate in number of connected components and the theoretical bound. | 120 |
| 3.12 | Five-dimensional You-Kaveh equation: comparison between the computed rate of decrease of the energy and the theoretical bound. | 120 |
| 3.13 | Five-dimensional You-Kaveh equation: comparison between the computed coarsening rate in number of connected components and the theoretical bound. | 121 |
| 3.14 | Two-dimensional sixth-order equation: comparison between the computed rate of decrease of the energy and the theoretical bound. | 121 |
| 3.15 | Two-dimensional sixth-order equation: comparison between the computed coarsening rate in number of connected components and the theoretical bound. | 122 |

A.1 Gray scale image of $g(z) = \Im(\sqrt{z-1}\sqrt{z+1})$ 132

LIST OF TABLES

Table

| | | |
|-----|--|----|
| 2.1 | Convergence of Algorithm 1. | 34 |
| 2.2 | Convergence of Algorithm 2. | 34 |
| 2.3 | Convergence of Algorithm 3. | 35 |
| 2.4 | Convergence of Algorithm 4. | 35 |
| 2.5 | Proportion of eccentric droplets for various area fractions. | 41 |

ABSTRACT

The content of this dissertation lies at the intersection of analysis and applications of PDE to image processing and computer vision applications. In the first part of this thesis, we propose efficient and accurate algorithms for computing certain area preserving geometric motions of curves in the plane, such as area preserving motion by curvature. These schemes are based on a new class of diffusion generated motion algorithms using signed distance functions. In particular, they alternate two very simple and fast operations, namely convolution with the Gaussian kernel and construction of the distance function, to generate the desired geometric flow in an unconditionally stable manner. We present applications of these area preserving flows to large scale simulations of coarsening, and inverse problems.

In the second part of this dissertation, we study the discrete version of a family of ill-posed, nonlinear diffusion equations of order $2n$. The fourth order ($n = 2$) version of these equations constitutes our main motivation, as it appears prominently in image processing and computer vision literature. It was proposed by You and Kaveh as a model for denoising images while maintaining sharp object boundaries (edges). The second order equation ($n = 1$) corresponds to another famous model from image processing, namely Perona and Malik's anisotropic diffusion, and was studied in earlier papers. The equations studied in this paper are high order analogues of the Perona-Malik equation, and like the second order model, their continuum versions violate parabolicity and hence lack well-posedness theory. We follow a recent technique from Kohn and Otto, and prove a weak upper bound

on the coarsening rate of the discrete in space version of these high order equations in any space dimension, for a large class of diffusivities. Numerical experiments indicate that the bounds are close to being optimal, and are typically observed.

CHAPTER I

Introduction

Introduced in the eighteenth century, Partial Differential Equations (PDE) are an important part of mathematical analysis that benefit from a strong and well established theory. Originally descended from physics, they have been extensively developed in mathematics, and later on in biology, finance and more recently in image processing. Other approaches that are widely used in imaging science include stochastic modeling (mostly based on Markov random field theory [44, 47, 61]) and wavelets (emerged from 1-D signal processing theory [62, 21, 29, 30, 31, 32]). This dissertation focuses on some PDE-based methods for image analysis.

One of the oldest concerns in image processing relates to image restoration (including denoising) and image enhancement. It is well known that most real images are degraded by the presence of noise and/or blur. Such corruption is unavoidable and usually originates in the input device that creates, transmits or records images (e.g., scanner, digital camera, sensor, etc.). These fluctuations in the pixel values of images render their analysis or processing difficult to perform with good success. A preprocessing step to remove or diminish the effects of such degradation is therefore necessary. Even now, such preprocessing is performed in most applications.

1.1 Image Restoration

In mathematics we define a grayscale image $f_0 : \Omega \subset \mathbb{R}^d \mapsto [0, 1]$ ($d = 2, 3$) to be a real, measurable and bounded function, defined on the image domain Ω , typically a rectangle (e.g., the computer screen). The value of the function $f_0(x)$ represents the grey level intensity or brightness of the image at location x . Typically, an image represents a scene containing a background and several objects. In addition, it is common that the values of an image inside the background and inside each object are smoothly varying, but are discontinuous or change abruptly across object boundaries (*edges*), thus creating large image gradients at these locations. We define edges as the locations where the gradient $|\nabla f|$ is large due to a discontinuity or a sharp transition in the image intensity.

The usual presence of noise in an image introduces some random perturbation of the image values, leading to a degraded image. We define the observed image f (a degraded version of the real image f_0) to be the following:

$$(1.1) \quad f(x) = f_0 + n(x),$$

where n is an additive noise (typically modeled as Gaussian). To account for the blur that may also corrupt the image, one may consider the more complicated model

$$(1.2) \quad f(y) = (Af_0)(y) + n(y),$$

where A is a linear operator representing the blur (usually a convolution). Such models lead to very interesting mathematics related to inverse problems as in the imaging section of Chapter II. In the rest of this section however, we consider the simpler denoising model (1.1) with associated problem: Given f , reconstruct f_0 , knowing (1.1).

1.1.1 Gaussian Filtering

In the classical theory, denoising is performed using a low-pass filter. This theory comes from Marr and Hildreth [63] and was later improved by Canny [14]. The low-pass filtering is done by convolving the image with Gaussians of increasing variance t (e.g., time). Witkin [99] observed that convolving the image with a Gaussian kernel of variance t was equivalent to solving the heat equation with the observed image f given as initial data

$$(1.3) \quad \begin{cases} u_t(x, t) = \Delta u(x, t) \\ u(x, 0) = f(x), \end{cases}$$

the solution of which is given in one dimension by

$$u(x, t) = (G_t * f)(x) \text{ for } t > 0,$$

where $G_\sigma(x) = \frac{1}{\sqrt{4\pi\sigma}} e^{-\frac{x^2}{4\sigma}}$ is the one dimensional Gaussian kernel. Using this technique, the detection of an edge is done in the following way: x is determined to be an edge at scale \sqrt{t} if $\Delta u(x, t)$ changes sign and if $|\nabla u(x, t)|$ is “large”. The last condition relies on a threshold defined initially and chosen a priori. In this context “large” means larger than the assigned threshold. Depending on the choice of threshold, small edges may be omitted if the threshold is too large, or some noise edges may still be kept if the threshold is too small. This shows one of the difficulties of this technique. Another observed problem is the location of the detected edges in low resolution images that is often shifted from their true location. This is caused by the smoothing effect of the convolution with Gaussian kernels, which blurs the edges and also moves their location. It is therefore necessary to introduce some locally high-pass filter to avoid losing the edges.

1.1.2 Energy minimization

It turns out that problem (1.3) can be recast into a variational approach with the L^2 energy

$$(1.4) \quad E(u) = \int_{\Omega} |\nabla u|^2 dx.$$

In this case, the energy E is minimized when u is the steady state solution of (1.3) with initial condition $u(x, 0) = f(x)$. In particular, the PDE described in (1.3) is gradient descent for E in equation (1.4). If we rewrite this constrained minimization as an unconstrained one, we obtain the following minimization problem

$$(1.5) \quad \min_u \left\{ F(u) := \int_{\Omega} |f - u|^2 dx + \lambda \int_{\Omega} |\nabla u|^2 dx \right\}.$$

The first term in F measures the fidelity to the data f , while the second term forces the solution u to be regular (in this case smooth), thus accounting for the blurring. The parameter λ is a Lagrange multiplier weighing the relative importance of the two terms in (1.5). The L^2 regularization was first introduced in 1977 by Tikhonov and Arsenin [92], and is sometimes referred to as Tikhonov regularization.

As noticed in the previous section, the solution to (1.3) and equivalently (1.5) introduces an oversmoothing leading to blurry images. Such oversmoothing can be explained by looking at the energy (1.5). The L^p norm with $p = 2$ of the gradient allows us to remove noise but unfortunately penalizes too much the gradients corresponding to edges. One should then decrease p to preserve edges as much as possible. Some work in this direction was done by Rudin, Osher and Fatemi [81] where they replaced the L^2 norm by an L^1 norm of the gradient, also called total variation. They considered the following minimization

$$(1.6) \quad \min_u \left\{ G(u) = \int_{\Omega} |f - u|^2 dx + \lambda \int_{\Omega} |\nabla u| dx \right\}.$$

This minimization is solved in the space of bounded variations ($BV(\Omega)$) which allows for discontinuous functions. Such space is very appropriate for computer vision applications since most images have discontinuities across edges. In their implementation, Rudin *et al.* dynamically updated the Lagrange multiplier $\lambda(t)$. Their result was quite satisfactory since the solution u approached a denoised version of the original image as $t \rightarrow \infty$, while keeping the edges relatively sharp and removing any spurious oscillations.

1.1.3 Nonlinear Diffusion

A different approach consists in looking for a restored image as a version of the initial image at a special scale. More precisely, the image u can be embedded in an evolution process $u(t, x)$. At time $t = 0$, $u(0, x) = f(x)$, where f is the observed image. As an attempt to restore an image while maintaining its edges sharp, Perona and Malik [76, 77] proposed the following nonlinear evolution equation

$$(1.7) \quad \begin{cases} u_t = \nabla \cdot (g(|\nabla u|) \nabla u) \\ u(x, 0) = f(x), \end{cases}$$

where g is a bounded, non-increasing positive function such that $\lim_{x \rightarrow \infty} g(x) = 0$ and $g(0) = 1$. It is easy to understand how the diffusion becomes spatially adaptive: if the image gradient is small, g will be close to one and the image will diffuse at this location. On the other hand, if the image gradient is large (e.g., close to an edge), g will be close to zero, and hence will almost stop the diffusion and keep the edges sharp. The model (1.7) constitutes the first instance of a nonlinear PDE arising in image processing and computer vision applications. Perona and Malik's goal was to devise a method for gradually simplifying (or *coarsening*) an image by diffusing out its details, starting with the smallest scales. An important point was to keep object boundaries (i.e., *edges*) in the image – where the image intensity is expected to be discontinuous or rapidly changing – sharp during the diffusion, until their abrupt disappearance at some point in the coarsening process. In their papers

[76, 77], Perona and Malik proposed and experimented with two choices of g :

$$(1.8a) \quad g(x) = e^{-\left(\frac{x}{b}\right)^2},$$

$$(1.8b) \quad \text{and } g(x) = \frac{1}{1 + \left(\frac{x}{b}\right)^2},$$

where b is a constant that acts like a contrast threshold to be chosen by the user. The second choice of g given in (1.8b) is the most commonly used diffusivity in applications of the model, and constitutes what is typically meant by the Perona-Malik model in the literature.

The time evolution of the (discretized) PDE generates “cartoon-like” images that get simpler with time, while maintaining sharp edges as was intended. An example of the Perona-Malik evolution on an image is shown in Figure 1.1. Unfortunately, the Perona-Malik model turns out to be ill-posed. To see this, consider (1.7) in one space dimension

$$u_t = (R(u_x)_x),$$

where $R(x) = xg(|x|)$, ($R : \mathbb{R} \mapsto \mathbb{R}$), and g is as in (1.8). Expanding the right hand side we obtain

$$u_t = R'(u_x)u_{xx},$$

which becomes backwards parabolic when $R'(u_x) < 0$. For the two choices of g given in (1.8), R is non-increasing whenever the gradient is large. Thus, in the regions of the image with large gradient, the process can be interpreted as a backward heat equation. The situation is the same in higher space dimensions. The ill-posedness signifies that (1.7) may not have a solution in general. A very practical implication of this ill-posedness is that similar initial images may lead to completely divergent evolutions and hence very dissimilar output images at the end of the processing.

Consequently, there is no complete well-posedness theory for (1.7) despite many efforts towards establishing rigorous results [9, 45, 46, 49, 57]. Another active line of research

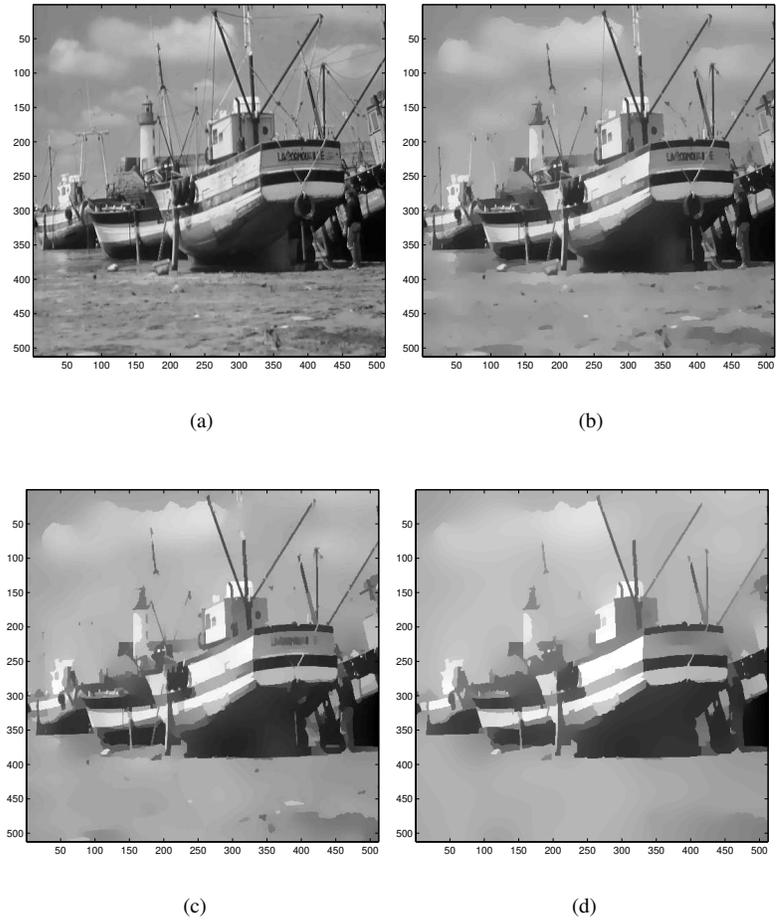


Figure 1.1: Simplification (or coarsening) of the boat image when taken as initial condition for the Perona and Malik equation. Figure 1.1(a) shows the original image, and Figures 1.1(b), 1.1(c) and 1.1(d) display the image at later times in the evolution.

towards understanding (1.7) considers its discrete in-space version [33, 34, 37], while others focus on studying its regularizations to obtain a well-posed equation [2, 7, 8, 18, 71]. Nevertheless, in practice, the Perona-Malik equation is much better behaved than expected, however, this phenomenon is still unexplained. It is likely that the behavior of the associated discrete problem does not reflect the ill-posedness of the continuous version, but this should be investigated further.

The gradual simplification of an image resulting from its evolution according to (1.7) can also be described by coarsening which produces at any time $t > 0$ a simplified version

(in this case piecewise constant) of the original image f . Thus in addition to denoising, the Perona-Malik process may provide a segmentation of the original image f at any scale t , although in practice (1.7) is rarely used solely for this purpose. Nevertheless, this shows that image denoising (and more generally image restoration) and segmentation are not totally disconnected, even though the segmentation problem has its own objectives and its own methodology.

1.2 Image Segmentation

Segmentation is typically used to locate objects or their boundaries (edges) in an image. The goal of segmentation is therefore to partition an image into its constituent parts to obtain a simplified image that is more meaningful and easier to analyze. Equivalently, one would like to have a simplified version of the image made of homogeneous regions that share certain visual characteristics. In particular, the image value should be smoothly varying inside each region, and discontinuous across their boundaries.

There are two main approaches to segmentation: the first one aims at detecting the contours of the objects lying in the original image f . The principle behind these edge detection techniques consists in matching deformable curves to the contours of objects by means of an energy functional. Examples of edge detection models include geodesic active contours [16, 17] and its precursor the Kass, Witkin and Terzopoulos model [55] (also known as snakes). The second approach consists in constructing an approximation of the original image f made up of distinct homogeneous regions separated by sharp discontinuities.

The well-known Mumford and Shah functional follows the second approach and constitutes one of the most studied segmentation models in the computer vision and image processing literature.

1.2.1 Mumford-Shah Functional

In this section, we consider Ω a bounded open set of \mathbb{R}^d , where d is the dimension (either 2 or 3), and f is the observed image to be segmented. In their pioneering work of 1989, Mumford and Shah [70] looked for a segmentation of a piecewise smooth image f into n smoothly varying regions R_i , $i = 1, \dots, n$, $n \in \mathbb{N}^*$, separated by a boundary Γ (the set of discontinuities). The image domain Ω is therefore the union of the n regions and the interface Γ :

$$\Omega = R_1 \cup R_2 \cup \dots \cup R_n \cup \Gamma.$$

Note that in this definition, Γ can contain open and closed curves. The Mumford-Shah energy functional to be minimized measures the discrepancy between the given image f and the current segmentation image u , with segmentation interface Γ . It is given by

$$(1.9) \quad E(u, \Gamma) := \lambda \int_{\Omega} (u - f)^2 dx + \mu \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + |\Gamma|,$$

where $|\Gamma|$ is the total length of the arcs making up Γ , and λ and μ are positive constants.

The Mumford-Shah minimization problem becomes

$$(1.10) \quad \min_{u, \Gamma} \left\{ E(u, \Gamma) = \lambda \int_{\Omega} (u - f)^2 dx + \mu \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + |\Gamma| \right\}.$$

In other words, the minimization (1.10) searches for a pair (u, Γ) , where $\Gamma \subset \Omega$ is the set of discontinuities and $|\Gamma|$ is the length of the curves making up Γ . The first term in the Mumford-Shah energy (1.9) is the usual fidelity term which measures the misfit between the given image f and the output image u . By minimizing E this term looks for images u that are close to f . The second term is a regularization term which measures the smoothness of the image u inside each region R_i . By minimizing E , it ensures that u has slow variation inside each region. The third term penalizes the length of the boundaries so that the segmentation is achieved using the shortest interface. The presence of this term

ensures that no trivial segmentation is obtained and also provides a way to select a scale in the segmentation. Each of the three terms described in (1.9) is equally important, and removing any one of them would lead to a trivial segmentation. Indeed if the first term is removed, the minimizer is trivially $\Gamma = \emptyset$ and $u = 0$. If only the second term is removed, the minimizer is $u = f$ and $\Gamma = \emptyset$. Now if E does not contain the length term, take Γ to be a fine grid of N horizontal and vertical lines and let u be constant on each of the N^2 grid squares (the constant is the average of f on each grid square). We can see that by letting $N \rightarrow \infty$, we can make E as small as desired. Hence all three terms are necessary.

The lack of differentiability of the Mumford and Shah functional (1.9) for a suitable norm, does not allow the use of Euler-Lagrange equations. Moreover, the discretization of the discontinuity set Γ is a complex problem. It is therefore common to approximate the Mumford-Shah functional by a sequence of regular functionals $E_\epsilon(u, \Gamma)$ defined on Sobolev spaces. As $\epsilon \rightarrow 0$, the sequence of functionals E_ϵ Γ -converges to the original Mumford-Shah functional, where the notion of Γ -convergence was introduced by De Giorgi to give meaning to the convergence of a sequence of functionals [27, 26]. Several ways of approximating the Mumford-Shah functional have been proposed in the literature, (see e.g. [10, 11, 19, 20, 48]) but the most commonly used in applications is the one proposed by Ambrosio and Tortorelli [3].

1.2.2 Ambrosio and Tortorelli's Approximation of the Mumford-Shah Functional

Ambrosio and Tortorelli [3] showed that the Mumford-Shah functional can be approximated by the following sequence of elliptic functionals

$$(1.11) \quad E_\epsilon(z, u) = \lambda \int_{\Omega} (f - u)^2 dx + \int_{\Omega} z^2 |\nabla u|^2 dx + \mu \int_{\Omega} \left(\frac{(1 - z^2)}{4\epsilon} + \epsilon |\nabla z|^2 \right) dx,$$

where $u, z \in W^{1,2}(\Omega \setminus \Gamma)$ and $\epsilon > 0$. Intuitively, the image domain Ω and interface Γ are replaced by the ‘‘interface-detection’’ function z which takes the value zero around Γ

and one everywhere else. The small parameter ϵ describes the thickness of the tubular neighborhood of Γ in which the function z makes the transition between zero and one. The term

$$\int_{\Omega} \left(\frac{(1-z^2)}{4\epsilon} + \epsilon |\nabla z|^2 \right) dx$$

approximates the length term $|\Gamma|$ in the Γ -convergence sense. As $\epsilon \rightarrow 0$, the function z becomes steeper around Γ and converges to a function Z which takes the value zero on Γ and one everywhere else. Ambrosio and Tortorelli [3] proved that the sequence of functionals (1.11) Γ -converges to the Mumford-Shah functional as $\epsilon \rightarrow 0$.

We observe that Ambrosio and Tortorelli's approximation is an example of a phase-field method which is a numerical technique for solving interfacial problems (see Section 1.3.2). A natural method to numerically compute a solution of the Mumford and Shah minimization problem is to consider one of its approximations (the most common being the Ambrosio and Tortorelli's approximation) and then discretize it using for example a finite difference scheme. However, since the full model can be quite difficult to implement, it is typical to consider simplified versions of the Mumford-Shah functional. One of the most popular simplifications considers the restriction of the Mumford and Shah energy (1.9) to piecewise constant functions. This simplified model was first introduced by Chan and Vese [23].

1.2.3 The Chan-Vese Model

Chan and Vese [23] proposed to simplify the full Mumford and Shah functional by restricting its minimization to piecewise constant functions of the form

$$(1.12) \quad u(x) = c_1 \mathbf{1}_{\Sigma}(x) + c_2 \mathbf{1}_{\Sigma^c}(x),$$

where c_1 and c_2 are constants, and Σ is the objects set. Note that Σ may have several connected components. In this setting, Chan and Vese looked at the following minimization

$$(1.13) \quad \min_{\Sigma, c_1, c_2} \left\{ E(c_1, c_2, \Gamma) = \lambda_1 \int_{\Sigma} (f - c_1)^2 dx + \lambda_2 \int_{\Sigma^c} (f - c_2)^2 dx + \mu |\partial \Sigma| + \nu |\Sigma| \right\},$$

where $|\Sigma|$ denotes the area of the set Σ , $|\partial \Sigma|$ its perimeter, and ν , λ_1 , λ_2 and μ are positive parameters. Note that this is more general than (1.9) restricted to piecewise constant functions, since the piecewise constant Mumford-Shah model is a particular case of the minimization (1.13) when $\nu = 0$, $\lambda_1 = \lambda_2 = 1$. An important advantage of this framework over previous methods (e.g. snakes and active contours [16, 17, 55]) is that the stopping criterion does not depend on the image gradient. This makes the model more robust. Besides, it is observed that the numerical method used by Chan and Vese is able to find features that are difficult to get, such as interior contours of objects, and seems to avoid getting stuck in local minima.

Chan and Vese implemented (1.13) using the level set method (see Section 1.3.3), however piecewise constant models such as (1.13) can also be treated via the phase-field method. In this case, the functional $E(c_1, c_2, \Gamma)$ in (1.13) is approximated by a sequence of functionals

$$(1.14) \quad E_{\epsilon}(c_1, c_2, z) = \lambda_1 \int_{\Omega} |f - c_1|^2 z^2 dx + \lambda_2 \int_{\Omega} |f - c_2|^2 (1 - z)^2 dx + \mu \int_{\Omega} \left(\frac{W(z)}{\epsilon} + \epsilon |\nabla z|^2 \right) dx,$$

where $W(z) = z^2(1 - z)^2$ is often called the double well potential. Intuitively, the double well potential function forces the function z to be one in Σ and zero in Σ^c . On the interface $\partial \Sigma$, z makes a transition between zero and one on an ϵ -thick layer. The term

$$\int_{\Omega} \left(\frac{W(z)}{\epsilon} + \epsilon |\nabla z|^2 \right) dx$$

approximates the length term $|\partial \Sigma|$ in the Γ -convergence sense, as shown by Modica and

Mortola in [68]. The sequence of functionals (1.14) therefore Γ -converges to the original energy $E(c_1, c_2, \Gamma)$ defined in equation (1.12).

Vese and Chan also generalized their piecewise constant model to a multiphase component framework with n constants instead of two [96]. In their implementation, they used only $\log n$ level set functions (which is the minimum needed). In addition, they extended the model to piecewise smooth functions with spatially dependent constants c_i . In both cases, their numerical implementation using the level set method, which was quite simple, provided good results.

1.3 Numerical Methods for Curve Evolution

Computing the motion of interfaces, e.g., curves in the plane or surfaces in space, is an essential component of many applications. For instance, in image processing and computer vision, many popular variational models for segmentation and reconstruction involve initializing a curve or surface and then evolving it towards features that are of interest (e.g., edges). Typically, the resulting motion is by a normal speed that includes geometric terms such as curvature of the interface and its derivatives. Due to recent research in this direction, there is now a wide variety of numerical schemes for simulating geometric motions of interfaces including front tracking, phase-field, level set and diffusion generated motion algorithms.

1.3.1 Front Tracking

Front tracking methods [13] involve an explicit discretization of the interface and approximate its motion by moving marker particles representing the interface. The two main benefits of front tracking methods are computational speed and high accuracy. In particular, these methods are very well suited for two dimensional motions of curves that do not cross. However, they become very difficult to implement whenever topological changes

occur, particularly in dimensions higher than two. Another problem with explicit methods is that they become unstable as the points of the interface get close to each other, or inaccurate as the points move away from each other. This makes it necessary to reparameterize and redistribute the points regularly throughout the computations, which is a major drawback since every reparameterization slightly moves the location of the interface, hence leading to algorithms that can often get stuck.

1.3.2 Phase-Field Method

The phase-field method uses an implicit representation of the interface through an auxiliary field variable (the “phase field”), which takes different values in different phases. In the simplest case, the phase field takes two different values (for example 0 and 1) in each of the phases, with a smooth transition around the interface. The thickness of the transition layer around the interface, usually denoted ϵ , is a parameter in the scheme. Examples of phase field methods include Ambrosio and Tortorelli’s approximation of the full Mumford-Shah functional and the approximation of the piecewise constant Mumford-Shah model, described by equation (1.11) and (1.14) respectively. Like the level set technique (see Section 1.3.3), the implicit representation of the interface enables this method to naturally handle topological changes. However, there are a few difficulties with the phase-field technique: first, the small parameter ϵ contributes to the stiffness of the problem, namely computations can take a very long time since the small parameter ϵ appears in a disadvantageous way in the CFL condition. Second, the ϵ -thick transition layer in the phase-field function needs to be resolved for accurate results, leading to impracticable sizes since the grid size Δx needs to be small enough to resolve the transition. Thus in practice, accuracy near the interface is often lost.

1.3.3 Level Set Method

The level set method is an implicit scheme introduced by Osher and Sethian in [73] as a simple way to compute and analyze the motion of an interface Γ bounding a region Σ . Note that the region Σ may have several connected components. Assuming Γ moves under the velocity field \vec{v} , the idea of the level set technique is to embed the interface Γ as the zero level set of a smooth (at least Lipschitz continuous) function $\phi(x, t)$ such that

$$\Gamma(t) = \{x : \phi(x, t) = 0\},$$

where ϕ is positive inside Γ and negative outside:

$$\Sigma(t) = \{x : \phi(x, t) > 0\}.$$

The motion of the curve Γ is obtained by convecting the values of ϕ with the velocity field \vec{v} in the following way:

$$\frac{\partial \phi}{\partial t} + \vec{v} \cdot \nabla \phi = 0,$$

or equivalently

$$(1.15) \quad \frac{\partial \phi}{\partial t} + v_N |\nabla \phi| = 0,$$

where v_N is the normal component of \vec{v} defined as $v_N = \vec{v} \cdot \frac{\nabla u}{|\nabla u|}$. We observe that the velocity field \vec{v} is only defined on the interface Γ and is arbitrary elsewhere. This brings forth the first issue of the level set technique, namely the problem of velocity extension. The velocity field \vec{v} is *a priori* only known on the interface, but is needed further away from the boundary to evolve ϕ according to equation (1.15) at least in a neighborhood of the interface. It is therefore necessary to extend \vec{v} to a smooth velocity field defined on a neighborhood of Γ . There are several ways to extend the velocity, one of the simplest being the extension of the velocity as a constant in the direction normal to the interface.

For more details on velocity extensions, we refer the reader to the book by Osher and Fedkiw [72] and the work of Tsai and Osher [94]. The second important issue of the level set method pertains to the reinitialization of the level set function. During the evolution, ϕ might either become too steep (in which case the algorithm can get stuck) or too flat (in which case instabilities are created). It is therefore necessary to periodically reinitialize the level set function to the signed distance function to the zero level set of ϕ . The zero level set of course remains unchanged. More details about this technique can be found in [72, 94].

Despite these issues, the level set technique remains very powerful. In particular, numerical implementations of (1.15) are simple and convenient, especially considering the fact that (1.15) can handle topological changes, which explicit methods (e.g. front tracking, etc.) cannot do easily. In the level set formulation, the curvature of the curve $\Gamma(t)$ can be written as

$$(1.16) \quad \kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right).$$

Moreover, the characteristic function of the set Σ , $\mathbf{1}_\Sigma(x)$, can also be written using the level set function ϕ as

$$\mathbf{1}_\Sigma(x) = H(\phi(x)),$$

where H is the one dimensional Heaviside function. The line or surface integral of a quantity ρ can therefore be written as

$$(1.17) \quad \int_{\mathbb{R}^d} \rho(x, t) |\nabla H(\phi)| dx,$$

and the surface or volume integral of g over Σ as

$$(1.18) \quad \int_{\mathbb{R}^d} \rho(x, t) H(\phi) dx,$$

where d is the dimension, either 2 or 3.

1.3.4 Diffusion Generated Motion

In this section, we describe two variants of a general class of algorithms called diffusion generated motion. The approach is based on alternately diffusing and resetting to its original type, the initial level set describing the interface (e.g., characteristic function or signed distance function).

The Merriman, Bence and Osher (MBO) scheme

The first diffusion generated motion algorithm (also called MBO scheme) was introduced by Merriman, Bence and Osher [65, 66, 67] as an efficient way to generate the motion of an interface by its mean curvature (or simply the curvature in 2D). The MBO algorithm is obtained by time splitting the well-known Allen-Cahn phase-field equation for motion by mean curvature. The resulting scheme alternates two steps, namely convolution and thresholding. More precisely, consider a set Σ in the domain $\Omega \subset \mathbb{R}^2$, with boundary $\partial\Sigma$ evolving with normal speed $v_N = \kappa$, where $\kappa(x)$ is the mean curvature of the interface $\partial\Sigma$ at the point $x \in \partial\Sigma$. Then $\partial\Sigma$ is the $\frac{1}{2}$ level set of the characteristic function $\mathbf{1}_\Sigma$ of the set Σ :

$$\partial\Sigma =: \left\{ x : \mathbf{1}_\Sigma(x) = \frac{1}{2} \right\}.$$

Given an initial set Σ_0 defined through its characteristic function $\chi_0 := \mathbf{1}_{\Sigma_0}$, and a time step size $\delta t > 0$, the MBO scheme generates a discrete sequence $\{\Sigma_j\}_{j \in \mathbb{N}}$ at subsequent times $j(\delta t)$ in the following way: from the set Σ_{j-1} , obtain the set Σ_j by alternating between the following two operations:

1. *Diffusion step.* From the set Σ_{j-1} defined through its characteristic function $\chi_{j-1} := \mathbf{1}_{\Sigma_{j-1}}$ at $(j-1)\delta t$, solve the following initial value problem for a length of time δt :

$$\begin{cases} u_t = \Delta u \\ u(x, 0) = \chi_{j-1}(x), \end{cases}$$

which is equivalent to forming the function

$$\mathcal{L}(x) = (G_t * \chi_{j-1})(x),$$

where G_t is the d -dimensional Gaussian kernel given by

$$(1.19) \quad G_t(x) = \frac{1}{(4\pi t)^{\frac{d}{2}}} e^{-\frac{|x|^2}{4t}}.$$

2. *Sharpening step.*

$$\chi_j(x) = \begin{cases} 0 & \text{if } \mathcal{L}(x) < \frac{1}{2}, \\ 1 & \text{else.} \end{cases}$$

The location of the interface $\partial\Sigma_j$ is given by the level set $\{x \in \Omega : \chi_j(x) = \frac{1}{2}\}$. The time discretization $\{\partial\Sigma_j\}_{j \in \mathbb{N}}$ has been proven to convergence to motion by mean curvature in the limit $\delta t \rightarrow 0^+$ (see [6, 39, 64]). One of the main advantages of this algorithm is its unconditional stability. In fact, the scheme remains monotone (i.e., preserves the order of sets) for all choices of δt , independent of the spatial resolution. In addition, its computational complexity is low ($O(N \log N)$ due to the FFT used in the convolution step), which makes it computationally more attractive than standard level set techniques that involve the solution of a nonlinear and degenerate PDE [73]. Nevertheless, there exist some semi-implicit schemes for level set methods introduced by Smereka in [91]. Several generalizations of the basic MBO scheme have been proposed for generating more complicated interfacial motions including anisotropic curvature motion, motion by curvature plus constant, and motion of multiple junctions [36, 64, 83, 84, 85, 86, 87, 88].

Despite its computational advantages, the MBO scheme inherits a major drawback from its construction using characteristic functions, namely inaccuracy of uniform grids. Indeed, characteristic functions cannot resolve the location of the interface better than the spatial grid size. Consequently, unless the grid size is refined concurrently with the

time step size, the approximate motion generated by the scheme gets stuck. It is therefore necessary to discretize the scheme with methods that can provide subgrid accuracy in the location of the interface. This was done by Ruuth in [84] through the efficient use of an unequally spaced FFT. Such an adaptive strategy is especially needed for simulating high-order motions.

To address this issue, Esedoğlu, Ruuth and Tsai [35] proposed to represent the interface using signed distance functions rather than characteristic functions. Their choice was motivated by the fact that unlike characteristic functions, signed distance functions can be represented on uniform grids at subgrid accuracies due to their Lipschitz continuity.

Diffusion Generated Motion using Signed Distance Functions

Esedoğlu, Ruuth and Tsai [35] recently proposed a new class of algorithms that generates a variety of interfacial motions with high accuracy on uniform grids. These algorithms are similar to the MBO scheme in flavor – they also alternate two steps with the diffusion step being in character the same – but they differ by the fact that they use the signed distance function to the interface instead of the characteristic function of the region. The new schemes thus simulate the motion of an interface by alternately diffusing and redistancing the signed distance function to the interface. The diffusion step consists of convolving the signed distance function with an appropriate kernel, usually chosen to be the Gaussian kernel, and the redistancing step simply consists of constructing the signed distance function to the interface from the previously diffused signed distance function (which at that point is no longer a signed distance function to the interface). The redistancing step replaces in essence the computationally very efficient thresholding step (i.e., sharpening step) of the MBO scheme. However, no efficiency is lost by having to construct the signed distance function to the interface every two steps, since there exist fast algorithms for computing signed distance functions (e.g., fast marching, fast sweeping, etc. [24, 82, 90, 93, 95]).

This new class of diffusion generated motion algorithms therefore provides an efficient and highly accurate technique for simulating a wide range of interfacial motions. Among such motions are the ones with normal speed

$$v_N = \kappa + S,$$

where κ is the mean curvature and $S : \mathbb{R}^d \rightarrow \mathbb{R}$ a given function, for which we describe the scheme. From an initial set Σ_0 defined through its signed distance function $d_0(x)$ and a time step size $\delta t > 0$, generate a time discrete approximation $\{\partial\Sigma_j\}_{j \in \mathbb{N}}$ at times $j(\delta t)$ by alternating the following two operations:

1. *Diffusion step.* From the set Σ_{j-1} defined through its signed distance function d_{j-1} at time $(j-1)\delta t$, form the level set function

$$\mathcal{L}(x) = \left(G_t * d_{j-1}\right)(x) + S_{j-1}(x)(\delta t),$$

where G_t is given in (1.19).

2. *Redistancing step.* Construct d_j , the signed distance function to the zero level set of \mathcal{L} , defining the new set Σ_j (and thus its boundary $\partial\Sigma_j$)

$$d_j(x) = \mathbf{Redist}(\mathcal{L}(x)).$$

High order in time versions of these schemes may also be obtained by combining two updates in time, with time step δt and $2\delta t$ respectively.

1.4 Contribution and Organization of this Dissertation

This dissertation provides both computational and theoretical results with applications to image processing and computer vision.

In the realm of applications, we propose efficient and accurate algorithms for computing certain area preserving geometric motions of curves in the plane, such as area

preserving motion by curvature. Our schemes are based on the class of diffusion generated motion using signed distance functions introduced by Esedoğlu, Ruuth and Tsai [35], and thus generate the desired geometric flow in an unconditionally stable manner. We check the numerical convergence of these algorithms and provide a few test problems that demonstrate their high accuracy. In addition, we present applications of these area preserving flows to large scale simulations of area preserving motion by curvature and inverse problems. The large scale computations of area preserving curvature motions are made possible by the computational efficiency of our schemes (which allows for very large grid sizes) and their unconditional stability (which allows us to take adaptive time steps). In the second application of our area preserving schemes, we investigate whether certain commonly used regularizations in medical imaging inverse problems (e.g. tomography) introduce significant bias in reconstructions. One such regularization (as used in the Mumford-Shah functional [70]) takes the form of a penalty on the perimeter of reconstructed objects. We propose a reconstruction algorithm based on area preserving flows, and show that perimeter regularizations do indeed introduce some shrinkage. We compare the performance of our area preserving flows with the performance of a Mumford-Shah based model (using perimeter regularization) and show that our model provides improved results over the Mumford-Shah based model.

The second topic of this dissertation relates to coarsening in high order analogues of the Perona-Malik equation. These high order analogues include the You-Kaveh equation [100] proposed as an improvement over the Perona-Malik model, especially regarding the staircasing artifact introduced by the Perona-Malik evolution. In this dissertation, we study the discrete version of a family of ill-posed, nonlinear diffusion equations of order $2n$. The fourth order ($n = 2$) version of these equations constitutes our main motivation, as it appears prominently in the image processing and computer vision literature. It was

proposed by You and Kaveh as a model for denoising images while maintaining sharp object boundaries. We follow a recent technique by Kohn and Otto, and prove a weak upper bound on the coarsening rate of the discrete in space version of these equations in any space dimension. Numerical experiments indicate that the bounds are close to being optimal, and are typically observed. These bounds provide a first step towards understanding the gradual simplification of an image occurring through a denoising process, with the ultimate goal of automatically selecting the stopping time of the evolution for a given level of image simplification.

The remaining chapters of this thesis are as follows. Chapter II introduces new algorithms for area preserving flows and provides applications to large scale simulations of area preserving curvature motion and inverse problems, while Chapter III focuses on the rigorous result related to coarsening in high order ill-posed nonlinear diffusion equations for image processing. To conclude, we discuss the relevance of our results in Chapter IV, and propose natural and new directions for future research.

CHAPTER II

Algorithms for Area Preserving Flows and Applications

2.1 Introduction

Motion by mean curvature has been extensively studied in the mathematics literature [41, 43, 54] and in applications such as crystal growth and image processing [1, 69]. Under this geometric flow each point x on a curve Γ moves with normal velocity $v_N = \kappa(x)$, where $\kappa(x)$ is the mean curvature of the curve at $x \in \Gamma$. It is also called the Euclidean curve shortening flow since the Euclidean perimeter of a curve shrinks as quickly as possible when evolving according to this motion. In addition, convex curves shrink to a point in finite time [43, 54]. Several algorithms have been proposed to simulate this motion, such as finite elements [97], level set methods [73, 91] and diffusion generated motion techniques [65, 67, 84]. A variant of this motion is the geometric flow that decreases the total perimeter of a collection of curves as quickly as possible while preserving the total enclosed area: it is referred to as area (or volume) preserving mean curvature motion. This motion finds applications in various fields including image processing [15, 42], and arises physically as a limit of the nonlocal Allen-Cahn equation modeling the phase separation in binary alloys [12, 50, 80]. The area preserving mean curvature motion is described by the normal velocity $v_N = \kappa - \bar{\kappa}$, where $\bar{\kappa}$ denotes the average mean curvature of the collection of curves. Several schemes based on the level set method [15, 75, 102] and threshold

dynamics [89] have been proposed for approximating this motion.

We describe new schemes for area preserving flows based on a new class of algorithms that generate the desired interfacial motion by alternating two simple and efficient steps: construction of the signed distance function, and convolution with a kernel (usually a Gaussian kernel). The resulting schemes are unconditionally stable, and have low, namely $O(N \log N)$, per time step cost, where N is the total number of grid points. In addition, we present applications of these efficient area preserving algorithms to large scale simulations of area preserving curvature motion, and to image reconstruction from medical imaging inverse problems.

2.2 Proposed schemes

In this section we introduce our area preserving schemes, and present systematic studies of their numerical convergence and accuracy.

2.2.1 Algorithms for area preserving flows

Building on the distance function dynamics for curvature motions described in Chapter I, we propose new and efficient algorithms for area preserving flows in two dimensions. These algorithms generate interfacial motions with normal velocities

$$(2.1) \quad v_N = \kappa - \bar{\kappa} + S,$$

where κ denotes the curvature, $\bar{\kappa}$ the average curvature and $S = S(x, t)$ is an additional normal speed term that may depend on space and time. Such terms arise for example in computer vision applications from data fitting terms in variational models (see Section 2.4). The core of our algorithm is the scheme for area preserving curvature motion which evolves interfaces with normal velocity $v_N = \kappa - \bar{\kappa}$. Under this motion, m disjoint curves Γ_i will evolve to decrease their total length while maintaining the total enclosed

area constant. In our algorithms, we use the fact that both κ and $\bar{\kappa}$ can be calculated very easily using the signed distance function to the interface.

Let us now recall a few well known properties of the signed distance function that hold more generally in \mathbb{R}^n (see e.g. [28, 40]). Consider the set $\Sigma \subset \Omega$ defined through its signed distance function d , and let $\partial\Sigma$ be its boundary. The first property of d is based on the fact that the normals to a smooth interface do not focus immediately, so that the signed distance function is smooth in a tubular neighborhood T of $\partial\Sigma$, and linear with slope one along the normals, namely

$$(2.2) \quad |\nabla d| = 1 \text{ for all } x \in T, \text{ with boundary condition } d|_{x \in \partial\Sigma} = 0.$$

The second property is that the Laplacian of the signed distance function d at a point x gives, up to a multiplicative constant depending on the dimension, the mean curvature of the isosurface of d passing through x :

$$(2.3) \quad \Delta d(x) = (n - 1)H(x),$$

where $H(x)$ denotes the mean curvature of the level set $\{\xi : d(\xi) = d(x)\}$, and n is the dimension. In two dimensions, we will denote $\kappa(x)$ the curvature of the curve $\partial\Sigma = \{\xi : d(\xi) = 0\}$, so that equation (2.3) simplifies to

$$(2.4) \quad \Delta d(x) = \kappa(x).$$

Before moving on to the expression of the average curvature $\bar{\kappa}$ in terms of the signed distance function d , we need to recall some simple definitions and properties. The average curvature of a curve C is defined as

$$(2.5) \quad \bar{\kappa} = \frac{1}{|C|} \int_C \kappa ds,$$

where $|C|$ denotes the length of the curve C . For a two dimensional connected set Σ of genus p , the average curvature of its boundary $\partial\Sigma$ can be expressed as

$$\bar{\kappa} = \frac{2\pi(1-p)}{|\partial\Sigma|},$$

where $|\partial\Sigma|$ is the total length of the boundary of Σ . The genus number p can be interpreted as the number of “holes” in the set Σ . If the set Σ is made up of K connected components $\Sigma = \bigcup_{j=1}^K \Sigma_j$, with Σ_j being a surface of genus p_j , then the average curvature of $\partial\Sigma$ becomes

$$(2.6) \quad \bar{\kappa} = \frac{2\pi\left(K - \sum_{j=1}^K p_j\right)}{|\partial\Sigma|},$$

From expression (2.6), we see that the only quantity left to compute is the perimeter of the set Σ ,

$$\begin{aligned} |\partial\Sigma| &= \int_{\partial\Sigma} ds \\ &= \int_{\partial\Sigma} |\nabla d|^2 ds \quad (\text{since by definition of the signed distance function, } |\nabla d| = 1) \\ &= \int_{\partial\Sigma} \nabla d \cdot \nu ds \quad (\text{where } \nu = |\nabla d| \text{ is the outward unit normal}) \\ &= \int_{\Sigma} \Delta d dx. \quad (\text{by the divergence Theorem}) \end{aligned}$$

Thus we have

$$(2.7) \quad |\partial\Sigma| = \int_{\Sigma} \Delta d dx,$$

which provides a simple relation between the length of the boundary of a set and its signed distance function. In our computations, we use equations (2.6) and (2.7) to compute the average curvature of a set. Notice that all the computations are done using only the signed distance function to the interface. Note also that under area preserving curvature motion, the boundaries of sets will evolve to decrease their total length while maintaining the total surface area at its initial value. Consequently, the final state of the evolution is a disk with

area equal to the initial total area. In particular, disks preserve their circular symmetry under this motion (one disk is stationary).

The complete algorithm for the general motion with normal speed $v_N = \kappa - \bar{\kappa} + S$ builds on the area preserving curvature motion scheme by simply shifting the location of the redistancing process by a constant determined by $S(x, t)$. For clarity in the description of the algorithms, we define

$$\#\{d > 0\} := \text{number of connected components of } \{x : d(x) > 0\}.$$

Algorithm 1. *Given the initial set Σ_0 defined through its signed distance function $d_0(x)$ and a time step $\delta t > 0$, generate the sets Σ_j via their signed distance functions $d_j(x)$ at the subsequent discrete times $t = j(\delta t)$ by alternating the following steps:*

1. *Using G_t in (1.19), form*

$$\mathcal{L}(x) = (d_{j-1} * G_{\delta t})(x) - \frac{2\pi (\#\{d_{j-1} > 0\} - \#\{d_{j-1} < 0\} + 1)}{\int_{d_{j-1} > 0} \Delta d_{j-1}(x) dx} (\delta t) + S_{j-1}(x)(\delta t).$$

2. *Construct the signed distance function d_j using*

$$d_j(x) = \mathbf{Redist}(\mathcal{L}(x)).$$

We also propose a second-order in time version of Algorithm 1 to achieve quadratic convergence in time.

Algorithm 2. Multi-step, second-order in time version. Given the initial set Σ_0 defined through its signed distance function $d_0(x)$ and a time step $\delta t > 0$, generate the sets Σ_j via their signed distance functions $d_j(x)$ at the subsequent discrete times $t = j(\delta t)$ by alternating the following steps:

1. Using G_t in (1.19), form

$$\mathcal{L}_1(x) = \left(d_{j-2} * G_{2\delta t} \right)(x) - \frac{2\pi \left(\#\{d_{j-2} > 0\} - \#\{d_{j-2} < 0\} + 1 \right)}{\int_{d_{j-2} > 0} \Delta d_{j-2}(x) dx} (2\delta t) + S_{j-2}(x)(2\delta t)$$

$$\mathcal{L}_2(x) = \left(d_{j-1} * G_{\delta t} \right)(x) - \frac{2\pi \left(\#\{d_{j-1} > 0\} - \#\{d_{j-1} < 0\} + 1 \right)}{\int_{d_{j-1} > 0} \Delta d_{j-1}(x) dx} (\delta t) + S_{j-1}(x)(\delta t).$$

2. Construct the signed distance function d_j using

$$d_j(x) = \mathbf{Redist} \left(\frac{1}{3} (4\mathcal{L}_2(x) - \mathcal{L}_1(x)) \right).$$

The overall computational complexity of the general algorithm is $O(N \log N)$. Indeed, counting connected components can be performed in $O(N)$ operations where N is the total number of grid points. The convolution is done in $O(N \log N)$ operations using the FFT. Also there exist algorithms, such as fast marching and fast sweeping, that construct the signed distance function in $O(N \log N)$ operations [24, 82, 90, 93, 95]. All the other terms used in the scheme, including the integral, can be done in $O(N)$ operations. Thus, we see that the per time step cost of the complete algorithm is $O(N \log N)$. In addition, due to its unconditional stability, there is no restriction on the time step size.

In the case of area preserving curvature motion (i.e., for $S = 0$), we propose slightly modified versions of Algorithms 1 and 2 that consist in matching exactly, at each time step $j(\delta t)$, the current area with the initial area. This strategy is similar to the one used by Ruuth and Wetton [89] in the case of threshold dynamics for area preserving curvature

motion. For clarity of the exposition, we define the area operator $\mathcal{A} : C^2(\mathbb{R}^2, \mathbb{R}) \rightarrow \mathbb{R}$ applied to a smooth function ϕ to be

$$\mathcal{A}(\phi) := \int_{\phi(x) > 0} dx = |\{x : \phi(x) > 0\}|,$$

which computes the area of the zero super level set of ϕ . Now the idea behind the modification of Algorithms 1 and 2 is to use Newton's method at each time step to find the stationary point λ_j^* of the function $\lambda_j \mapsto a(\lambda_j) - a_0$, where a_0 is the initial area, and $a(\lambda_j)$ the current area at time step j defined as

$$a(\lambda_j) = \mathcal{A}(d_{j-1} * G_{\delta t} - \lambda_j).$$

The initial condition for the Newton iteration is taken to be

$$\lambda_j^0 = \frac{2\pi (\#\{d_{j-1} > 0\} - \#\{d_{j-1} < 0\} + 1)}{\int_{d_{j-1} > 0} \Delta d_{j-1}(x) dx} (\delta t),$$

which approximates the average curvature. Note that this expression is used in Algorithms 1 and 2. The signed distance function d_j is then constructed using

$$d_j = \mathbf{Redist}(d_{j-1} * G_{\delta t} - \lambda_j^*).$$

To modify Algorithm 2, we proceed in the same way, namely at each time step $j(\delta t)$, we compute $\lambda_{j,1}^*$ and $\lambda_{j,2}^*$ coming from the result of a Newton iteration using d_{j-2} with time step $2\delta t$ and d_{j-1} with time step δt respectively. The final update in this case becomes

$$(2.8) \quad d_j = \mathbf{Redist}\left(\frac{1}{3} \left(4(d_{j-1} * G_{\delta t} - \lambda_{j,2}^*) - (d_{j-2} * G_{\delta t} - \lambda_{j,1}^*)\right)\right).$$

The modified algorithm reads as follows

Algorithm 3. Area Preserving Curvature Motion. Given the initial set Σ_0 with area a_0 defined through its signed distance function $d_0(x)$, a time step $\delta t > 0$ and a constant $\eta > 0$, generate the sets Σ_j via their signed distance functions $d_j(x)$ at the subsequent discrete times $t = j(\delta t)$ by alternating the following steps:

1. Using Newton's method with initial guess

$$\lambda_j^0 = \frac{2\pi (\#\{d_{j-1} > 0\} - \#\{d_{j-1} < 0\} + 1)}{\int_{d_{j-1} > 0} \Delta d_{j-1}(x) dx} (\delta t),$$

find λ_j^* such that $|\mathcal{A}(d_{j-1} * G_{\delta t} - \lambda_j^*) - a_0| < \eta$, and form

$$\mathcal{L}(x) = (d_{j-1} * G_{\delta t})(x) - \lambda_j^*.$$

2. Construct the signed distance function d_j using

$$d_j(x) = \mathbf{Redist}(\mathcal{L}(x)).$$

The multi-step version of Algorithm 3 can be obtained by following the same pattern as Algorithm 2 and using the update described in equation (2.8).

Algorithm 4. Area Preserving Curvature Motion: second-order in time version. Given Σ_0 with area a_0 having distance function $d_0(x)$, $\delta t > 0$ and $\eta > 0$, generate Σ_j at times $t = j(\delta t)$ by alternating the following steps:

1. Using Newton's method with initial guesses

$$\lambda_{j,1}^0 = \frac{2\pi \left(\#\{d_{j-2} > 0\} - \#\{d_{j-2} < 0\} + 1 \right)}{\int_{d_{j-2} > 0} \Delta d_{j-2}(x) dx} (\delta t)$$

$$\text{and } \lambda_{j,2}^0 = \frac{2\pi \left(\#\{d_{j-1} > 0\} - \#\{d_{j-1} < 0\} + 1 \right)}{\int_{d_{j-1} > 0} \Delta d_{j-1}(x) dx} (\delta t),$$

find $\lambda_{j,1}^*$ and $\lambda_{j,2}^*$ such that

$$\left| \mathcal{A}(d_{j-2} * G_{\delta t} - \lambda_{j,1}^*) - a_0 \right| < \eta \text{ and } \left| \mathcal{A}(d_{j-1} * G_{\delta t} - \lambda_{j,2}^*) - a_0 \right| < \eta,$$

and form

$$\mathcal{L}_1(x) = (d_{j-2} * G_{\delta t})(x) - \lambda_{j,1}^*$$

$$\mathcal{L}_2(x) = (d_{j-2} * G_{\delta t})(x) - \lambda_{j,2}^*.$$

2. Construct the signed distance function d_j using

$$d_j(x) = \mathbf{Redist} \left(\frac{1}{3} (4\mathcal{L}_2(x) - \mathcal{L}_1(x)) \right).$$

Note that although Algorithms 3 and 4 preserve area more accurately than Algorithms 1 and 2, they will be inapplicable in situations where the total area may change due to the presence of a non-zero term S , such as the imaging application described in Section 2.4.

A variant of area preserving curvature motion considers component-wise area preserving curvature motion. In this case, the area of each connected component is preserved during the entire evolution rather than the total area. This variant is used in the imaging application of Section 2.4, where each connected component represents a different object in the image. For simplicity in the description of the component-wise algorithm, we define

the following function:

$\mathcal{L}_{out} = \mathbf{ComponentwiseAP}(\mathcal{L}, d, \delta t)$

1. Find the number of connected components q of the set Σ defined through its signed distance function d
2. For each of the connected component Σ_k , extract its signed distance function d^k ($k = 1 \cdots q$)

(a) Compute the local average curvature

$$\bar{\kappa}_k = \frac{2\pi(2 - \#\{d^k < 0\})}{\int_{d^k > 0} \Delta d^k(x) dx}$$

(b) In an ϵ -neighborhood of Σ_k , update

$$\mathcal{L}(x_k) = \mathcal{L}(x_k) - \bar{\kappa}_k(\delta t),$$

for $x_k \in N_\epsilon(\Sigma_k)$, where $N_\epsilon(\Sigma_k) =: \{x : |x - y| < \epsilon, y \in \Sigma_k, \epsilon > 0\}$.

3. $\mathcal{L}_{out} = \mathcal{L}$; Return \mathcal{L}_{out} .

The component-wise area preserving algorithm reads as follows:

Algorithm 5. Component-wise Area Preserving Flow. Given the initial set Σ_0 defined through its signed distance function $d_0(x)$ and a time step $\delta t > 0$, generate the sets Σ_j via their signed distance function $d_j(x)$ at the subsequent discrete times $t = j(\delta t)$ by alternating:

1. Using G_t in (1.19), form

$$\mathcal{L}(x) = d_{j-1} * G_{\delta t} + S_{j-1}(\delta t).$$

2. Update \mathcal{L} using

$$\mathcal{L} = \mathbf{ComponentwiseAP}(\mathcal{L}, d_{j-1}, \delta t).$$

3. Construct the signed distance function d_j using

$$d_j(x) = \mathbf{Redist}(\mathcal{L}(x)).$$

The second order in time version of Algorithm 5 is obtained by following the idea used in Algorithms 2 and 4.

2.2.2 Numerical convergence study

In this section we describe some convergence studies done with the algorithms for area preserving curvature motion introduced in the previous section. In the computations presented below, we used a second order accurate procedure to construct the signed distance function in a tubular neighborhood of the interface. For details on more sophisticated algorithms for constructing signed distance functions, we refer the reader to [22, 24, 82, 90, 93, 101].

We investigate the convergence of Algorithms 1 and 2 for $S = 0$, i.e., for the area preserving curvature motion algorithm. The results are displayed in Tables 2.1 and 2.2. The convergence test is done on an ellipse and the evolution is computed over the time

interval $[0, 0.01]$. The initial condition is an ellipse \mathcal{E} with major axis $a = 0.45$ and minor axis $b = 0.2$ on the domain $[0, 1]^2$. The major and minor axes of the final curve (which is no longer an ellipse) are 0.4 and 0.22 respectively. At the final time $T = 0.01$ we measure the quantity $\int_{\Sigma(T)} (x^2 + y^2) d\Omega$, where $\Sigma(t) \in \mathbb{R}^2$ evolves under area preserving curvature motion with initial condition $\Sigma(0) = \mathcal{E}$. We compare this quantity with the exact quantity $\int_{\Sigma_e(T)} (x^2 + y^2) d\Omega$, where the exact evolution $\Sigma_e(t)$ is computed using a front tracking technique with a very fine discretization of the parameterized curve. We also display the error in area and its associated convergence rate.

| Resolution | # of Time Steps | Relative error in $\int_{\Sigma(T)} (x^2 + y^2) d\Omega$ (in %) | Order | Error in area | Order |
|--------------------|-----------------|---|-------|---------------|-------|
| 33×33 | 20 | 0.0167 | – | 0.000921 | – |
| 65×65 | 40 | 0.3218 | –4.27 | 0.000561 | 0.72 |
| 129×129 | 80 | 0.2046 | 0.65 | 0.000271 | 1.05 |
| 257×257 | 160 | 0.1002 | 1.03 | 0.000120 | 1.18 |
| 513×513 | 320 | 0.0487 | 1.04 | 0.000055 | 1.12 |
| 1025×1025 | 640 | 0.0240 | 1.02 | 0.0000263 | 1.06 |
| 2049×2049 | 1280 | 0.0119 | 1.01 | 0.0000128 | 1.03 |

Table 2.1: Convergence of Algorithm 1 for $S = 0$. The initial condition is an ellipse with major axis $a = 0.45$ and minor axis $b = 0.2$ on $[0, 1]^2$. The evolution was computed for $t \in [0, 0.01]$.

| Resolution | # of Time Steps | Relative error in $\int_{\Sigma(T)} (x^2 + y^2) d\Omega$ (in %) | Order | Error in area | Order |
|--------------------|-----------------|---|-------|---------------|-------|
| 33×33 | 20 | 0.8098 | – | 0.00164 | – |
| 65×65 | 40 | 0.4919 | 0.72 | 0.000710 | 1.20 |
| 129×129 | 80 | 0.0951 | 2.37 | 0.000150 | 2.24 |
| 257×257 | 160 | 0.00738 | 3.69 | 0.0000210 | 2.84 |
| 513×513 | 320 | 0.000928 | 2.99 | 0.00000434 | 2.27 |
| 1025×1025 | 640 | 0.000196 | 2.25 | 0.00000109 | 1.99 |
| 2049×2049 | 1280 | 0.00000915 | 4.42 | 0.000000265 | 2.04 |

Table 2.2: Convergence of Algorithm 2 for $S = 0$. The initial condition is an ellipse with major axis $a = 0.45$ and minor axis $b = 0.2$ on $[0, 1]^2$. The evolution was computed for $t \in [0, 0.01]$.

As can be seen in Table 2.1, Algorithm 1 settles into a clearly first order convergence rate. Table 2.2 shows the convergence rate for Algorithm 2, which on this example turns out to be significantly higher than second order, perhaps due to some cancellation of errors. In any case, Algorithm 2 achieves very high accuracy even on very modest sized grids.

We now present the convergence of Algorithms 3 and 4 on the same ellipse test. The

results are displayed in Tables 2.3 and 2.4 respectively. We stress that these last two algorithms cannot be extended to the more general velocity (2.1) with nonzero $S(x, t)$. In both convergence tests, the area was preserved up to an error of 10^{-14} or less. Table 2.3 displays the convergence of Algorithm 3 and Table 2.4 the convergence of Algorithm 4.

| Resolution | # of Time Steps | Relative error in $\int_{\Sigma(T)} (x^2 + y^2) d\Omega$ (in %) | Order |
|--------------------|-----------------|---|-------|
| 33×33 | 20 | 0.6256 | – |
| 65×65 | 40 | 0.0588 | 3.41 |
| 129×129 | 80 | 0.0213 | 1.47 |
| 257×257 | 160 | 0.0194 | 0.13 |
| 513×513 | 320 | 0.0117 | 0.73 |
| 1025×1025 | 640 | 0.00631 | 0.89 |
| 2049×2049 | 1280 | 0.00327 | 0.95 |

Table 2.3: Convergence of Algorithm 3. The initial condition is an ellipse with major axis $a = 0.45$ and minor axis $b = 0.2$ on $[0, 1]^2$. The evolution was computed for $t \in [0, 0.01]$.

| Resolution | # of Time Steps | Relative error in $\int_{\Sigma(T)} (x^2 + y^2) d\Omega$ (in %) | Order |
|--------------------|-----------------|---|-------|
| 33×33 | 20 | 0.3040 | – |
| 65×65 | 40 | 0.0112 | 4.76 |
| 129×129 | 80 | 0.00735 | 0.61 |
| 257×257 | 160 | 0.00689 | 0.09 |
| 513×513 | 320 | 0.00197 | 1.81 |
| 1025×1025 | 640 | 0.000488 | 2.01 |
| 2049×2049 | 1280 | 0.000117 | 2.06 |

Table 2.4: Convergence of Algorithm 4. The initial condition is an ellipse with major axis $a = 0.45$ and minor axis $b = 0.2$ on $[0, 1]^2$. The evolution was computed for $t \in [0, 0.01]$.

In addition to the convergence studies described above, we tested our area preserving curvature motion algorithm on an initial configuration containing three circles with radii 0.15, 0.2 and 0.22 on $[0, 1]^2$. Since circles remain circles under this motion, we monitored the evolution of each of the three radii for $t \in [0, 0.1]$, and compared it with the exact evolution obtained by numerical integration of the coupled ODEs for the radii. The circles were placed far apart initially so that no collision occurred during the evolution. Figure 2.1 shows the three circles in the initial condition (thick line) and the final curves at time $t = 0.1$ (fine line). The computed evolution of the three radii is compared to the exact one in Figure 2.2.

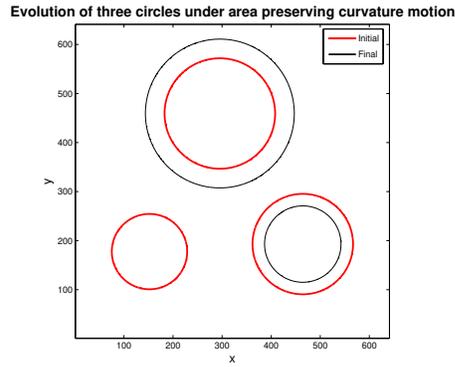


Figure 2.1: Evolution of three circles under area preserving curvature motion for $t \in [0, 0.1]$. The initial condition is shown by the thick curves, and the final curves at $t = 0.1$ are displayed by the thin curves. In this configuration, the largest circle grew while the other two shrunk (the smallest one actually disappeared).

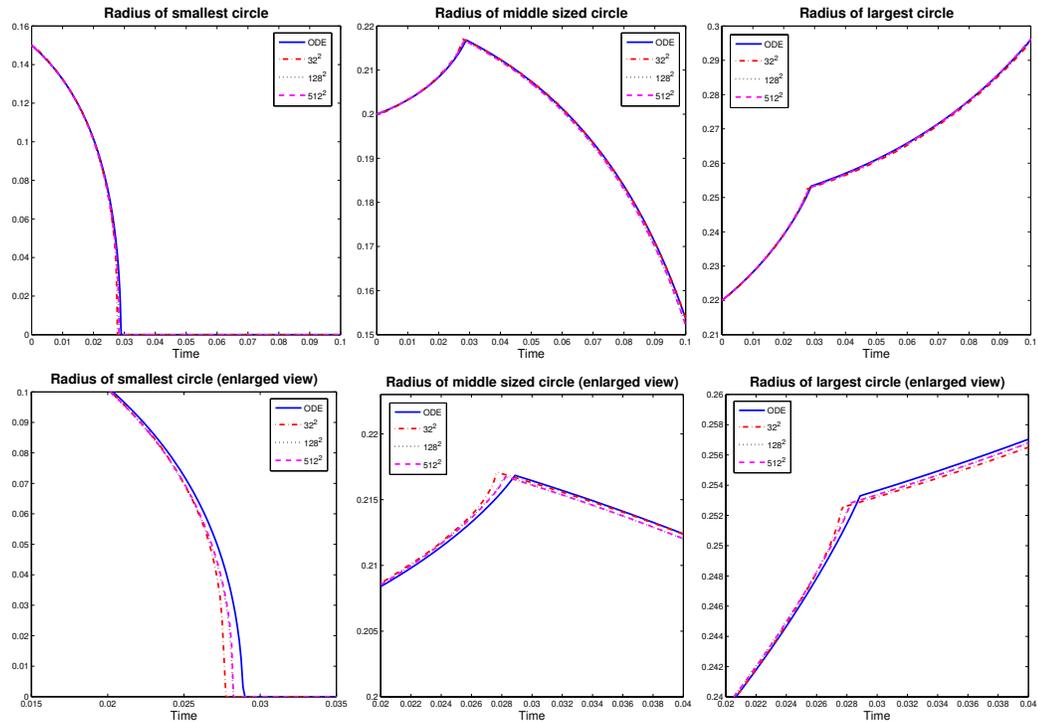


Figure 2.2: Comparison between the evolutions of the exact and computed radii obtained from three initial circles taken as initial condition and evolved under area preserving curvature motion. The initial three circles have the following radii: 0.15, 0.2 and 0.22. In each of the plots, we superimpose the exact evolution (in bold) and the evolution of the radii computed on a 32^2 , 128^2 and 512^2 grid. The top row shows the entire evolution, while the bottom row shows an enlarged view of the plots where the graph is not differentiable (corresponding to the disappearance of the smallest circle). This computation was performed with Algorithm 4.

2.3 Application: large scale simulations of area preserving curvature motion

In this section, we demonstrate the capacity of our proposed algorithms to handle large scale simulations with very good accuracy.

2.3.1 Curve shortening at various area fractions

Geometrically in two dimensions, the area preserving curvature flow describes the shortening of a curve (or interface) separating two phases, while maintaining the area of each phase equal to their respective initial area. A natural question therefore arises: at what rate does the total length of the curve decrease? Scaling arguments [25] suggest that the total length L decreases as a power law in time according to

$$L(t) \sim t^{-\frac{1}{2}}.$$

Dai [25] also obtained a rigorous result for the rate of decrease of L in the case of a dilute mixture. Specifically, he showed that for a collection of non-intersecting and convex plane curves, the total length $L(t)$ cannot decrease faster than $t^{-\frac{1}{2}}$ in a time average sense. In the simplified case that Dai considered, there is no coalescence and the only singularity is the disappearance of curves. In the general case however, collisions of curves will occur causing singularities in the curvature to appear at the times of first intersections. In fact, at the points of intersections, the curvature will be infinite, leading to an immediate smoothing and a fast decay of the sum of the lengths pertaining to the merging curves. Figure 2.3 illustrates this point on a simple example of two curves intersecting each other during their evolution under area preserving curvature motion.

In this context, we refer to the phase enclosed by the curves as Phase 1. Phase 2 denotes its complement. As a demonstration of the proposed algorithms, we present some simulations of area preserving curvature motion on very large collections of closed curves (or droplets). We also measure certain statistics related to the configurations of droplets

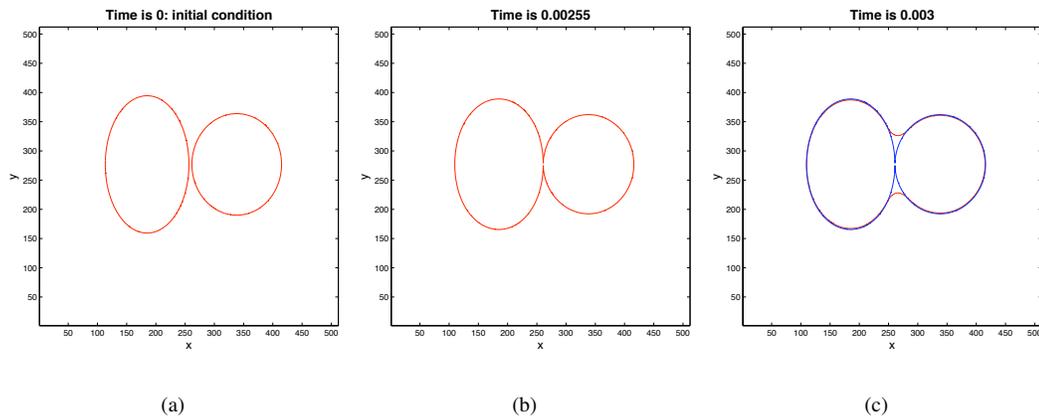


Figure 2.3: Coalescence of two curves during their evolution under area preserving curvature motion. Figure 2.3(a) shows the initial condition and Figure 2.3(b) displays the new curve just as the two previous ones collided. The curvature of the new curve is very large at the point of intersection. Figure 2.3(c) shows the new curve shortly after the collision occurred superimposed with the curve shown in Figure 2.3(b).

during their evolution. As Otto observed, there is no lower bound possible on the energy (length) that would hold for any initial data. It should not be difficult to construct configurations of droplets that would coarsen faster. In this work, we are concentrating on the area preserving curvature motion evolution of droplets starting from random initial data. In particular, we are interested in the effect of collisions that occur throughout the evolution. Due to the high accuracy achieved by our algorithms, we were able to resolve droplets with width and length as small as 15 pixels. As a result, our simulations evolved configurations containing up to 25000 droplets on a 4097^2 grid. Our computations considered various area fractions of Phase 1 ranging from 10% to 50%, which equivalently considered area fractions of Phase 2 ranging from 50% to 90%.

2.3.2 Numerical results

In this section, we present the results of our simulations. We construct the initial data by generating random sets of points from a uniform distribution, and placing a disk centered at each of the points with radius randomly chosen from a uniform distribution. The droplets are obtained by taking the union of the disks. From such initial configurations

(i.e., randomly generated droplet configurations), it would be reasonable to expect a certain collision rate during the evolution under area preserving curvature motion. Figure 2.4 illustrates the area preserving motion by curvature evolution on initial data with 10% and 40% area fraction. To avoid boundary effects, the computations are done on a slightly larger domain than $[0, 1]^2$. Additionally, to prevent premature mergings of nearby droplets, we divide, at each time step of the computations, the sets of droplets into subsets containing droplets that are further apart, and update the signed distance function of each subset separately. This allows individual grains to evolve independently. Last, since the average size of the droplets increases during the evolution, we perform our computations using an adaptive time step regulated by the value of the average size of the droplets. This adaptivity is made possible by the unconditional stability of our algorithms. In Figure 2.5, we demonstrate that taking larger time steps does not significantly change the configurations of droplets; indeed even with large time steps, the algorithm is able to resolve the dynamics.

The first quantity that we study, is the rate of decrease of the total length L . Figure 2.6 compares the rate of decrease of L (which is also the energy dissipated by the evolution) with the theoretical bound $t^{-\frac{1}{2}}$ for various area fractions. Note that there are rigorous results indicating this bound only in the case of convex curves that never collide. Nevertheless, for these randomly generated initial conditions, the rate of decrease of L , obtained from our computations, closely follows the theoretical bound.

Another quantity that we study is the number of connected components K in the configurations of droplets. Based on the rate of decay of L , and using a simple heuristic argument on a uniform configuration of disks, we can show that the number of connected components should essentially decay as $\frac{1}{t}$. In Figure 2.7 we compare the numerically observed rate of decrease of K for various area fractions with the bound $\frac{1}{t}$. For the randomly

generated initial conditions chosen in our simulations, the plots show very good agreement between the computed rate and the bound $\frac{1}{t}$. Intuition also suggests that at any given time, a certain population of droplets will be just about to collide, generating configurations at which the energy decrease rate must be elevated (the proportion of droplets just about to collide to all droplets would of course depend on the area fraction). However, it appears that even if collisions between droplets cause a deviation in the coarsening rate, our numerical experiments, despite their large size, are still not large enough to discern such a difference – perhaps the effect is very small. Nevertheless, as one would expect, we observe in our numerical simulations that at any time during the evolution (outside of the transient initial period and the final stage where only a few droplets remain), there is a constant proportion of eccentric droplets (i.e., droplets that are the result of a recent collision). This observation agrees with the expectation that collisions occur at a definite rate in proportion to the number of droplets. To exhibit this behavior, we measure the isoperimetric ratio

$$(2.9) \quad I(C) := \frac{P^2}{A}$$

of each droplet to characterize their shape. In (2.9), C is a closed curve, P is its perimeter and A its area. Since the isoperimetric ratio is minimized by a circle, we have that for any closed curve

$$I \geq 4\pi \sim 12.57,$$

where $I = 4\pi$ when C is a circle. For an ellipse with minor axis b and major axis $a = 3b$, the isoperimetric ratio is approximately $I \sim 18.95$. For a more elongated ellipse with minor axis b and major axis $a = 4b$ the isoperimetric ratio is approximately $I \sim 23.42$. From these references, we look at the proportion of droplets with isoperimetric ratio $I > 20$. Figure 2.8 shows the distribution of isoperimetric ratios at different times throughout the

evolution for a configuration of droplets with 50% area fraction. Table 2.5 displays the proportion of eccentric droplets for the various area fractions studied in our computations. Figure 2.9 shows the proportion of eccentric droplets and the total number of droplets for configurations with area fractions 30%, 40% and 50%. In each case, the proportion of eccentric droplets decreases very quickly during the transient initial phase (as is the total number of droplets) and then stabilizes itself around a constant value. The proportion of eccentric droplets remains constant until the total number of droplets becomes too small.

| Area fraction | Proportion of eccentric droplets | # of droplets at onset of constant proportion | # of droplets at the end of constant proportion |
|---------------|----------------------------------|---|---|
| 10% | 0.12% | 5054 | 193 |
| 20% | 0.46% | 2623 | 78 |
| 30% | 1.18% | 2932 | 91 |
| 40% | 2.49% | 2766 | 34 |
| 50% | 3.81% | 3885 | 49 |

Table 2.5: Proportion of eccentric droplets (the isoperimetric ratio of which satisfies $I > 20$) for various area fractions. The proportion of eccentric droplets increases with the area fraction, as one would expect.

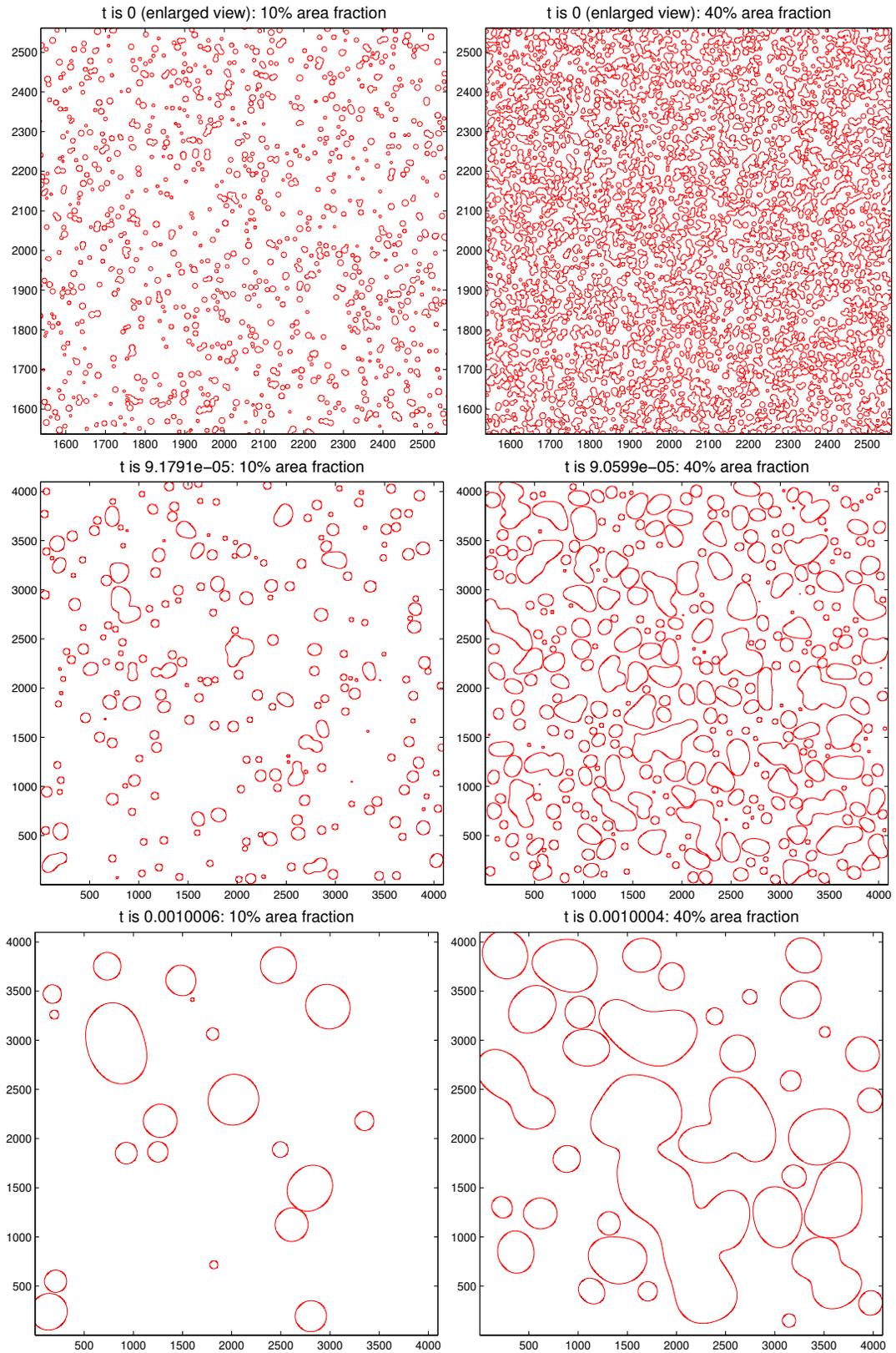


Figure 2.4: Evolution of two configurations of droplets under area preserving curvature motion. The left-hand column displays the evolution of a configuration of droplets with 10% area fraction. The right-hand column shows the evolution for a configuration with 40% area fraction. Because of the very large number of droplets in the early configurations, we only show a subset of these configurations in the top two plots. These subsets have been enlarged for a better view.

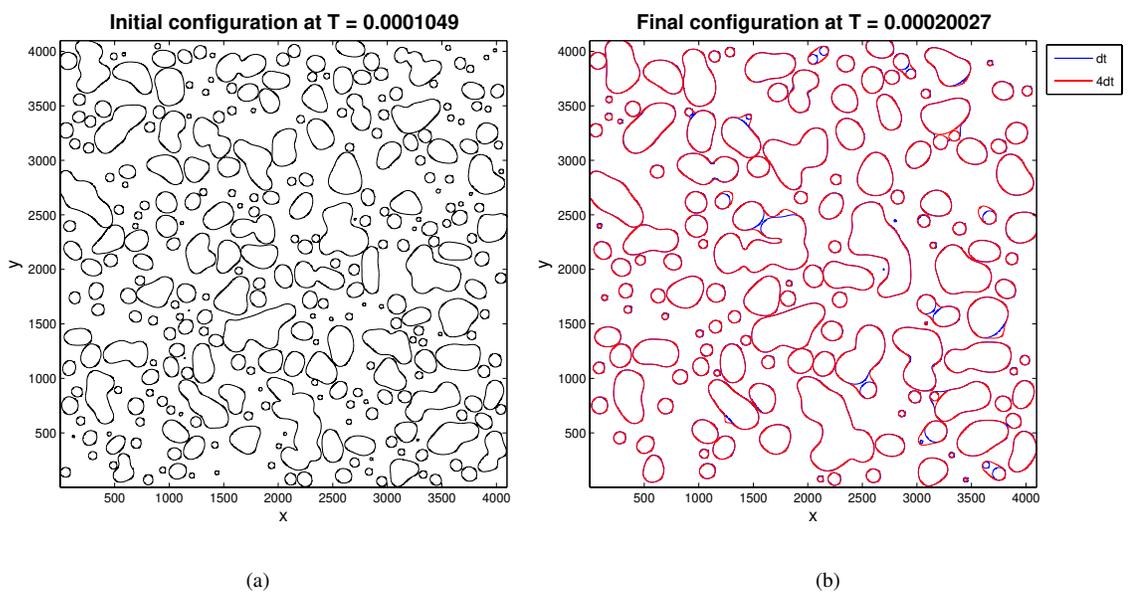


Figure 2.5: Comparison of two configurations of droplets obtained by computing their evolution with the standard time step (δt) and the adaptive one $4\delta t$. In this example, $\delta t = 4.7710^{-7}$. Figure 2.5(a) shows the initial configuration and Figure 2.5(b) displays the configurations obtained from computations using δt and $4\delta t$. Except for minor differences, both computations are able to resolve the dynamics, thus allowing the use of larger time steps.

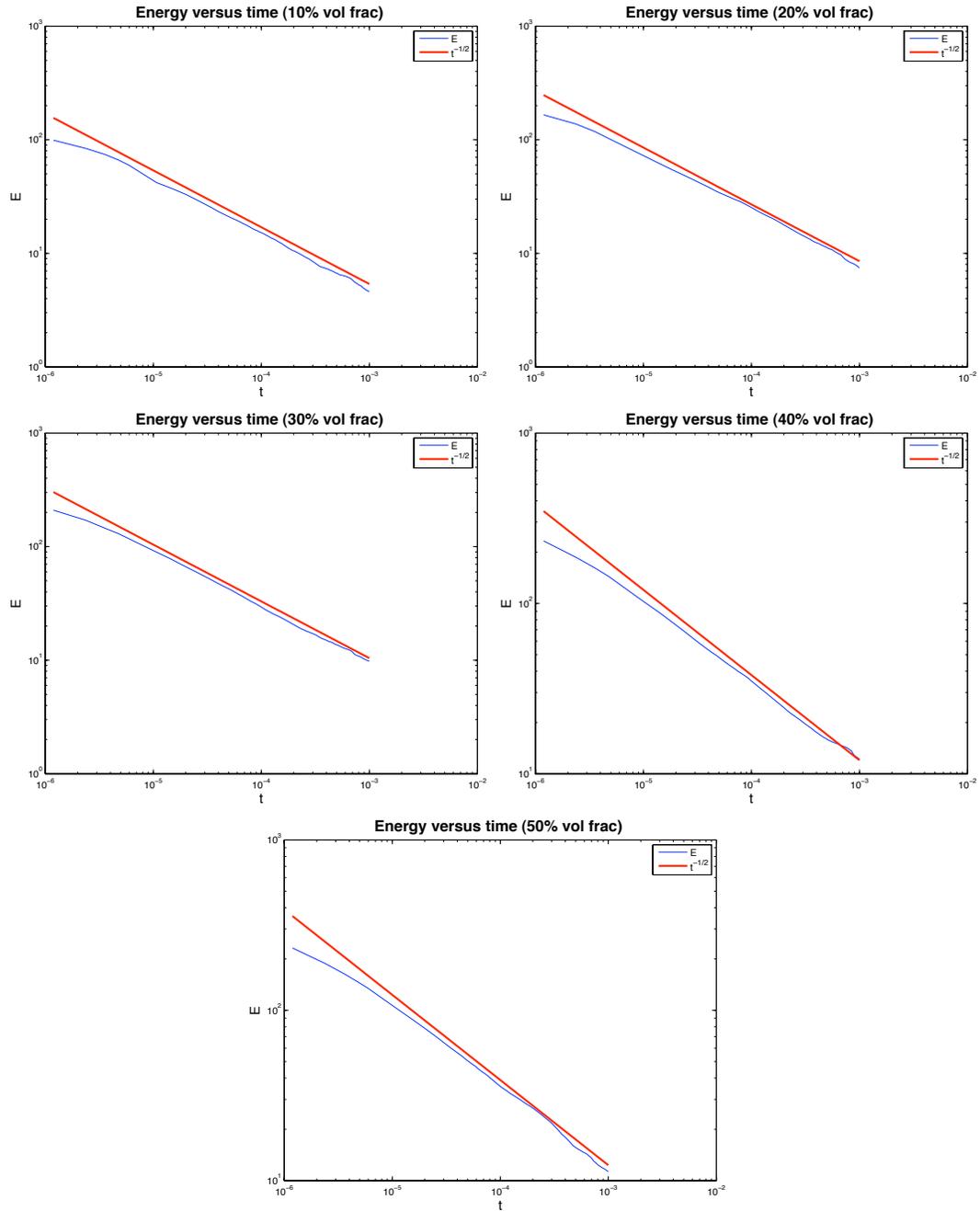


Figure 2.6: Loglog plot of the total length L of the boundary of all droplets (also the energy being decreased by the evolution) versus time for various area fractions. The thick line corresponds to the theoretical bound $t^{-\frac{1}{2}}$. From top to bottom, and left to right, the plots correspond to area fractions ranging from 10% to 50%. The plots acknowledge good agreement between the theoretical bound $t^{-\frac{1}{2}}$ and the numerically observed rates (fine line).

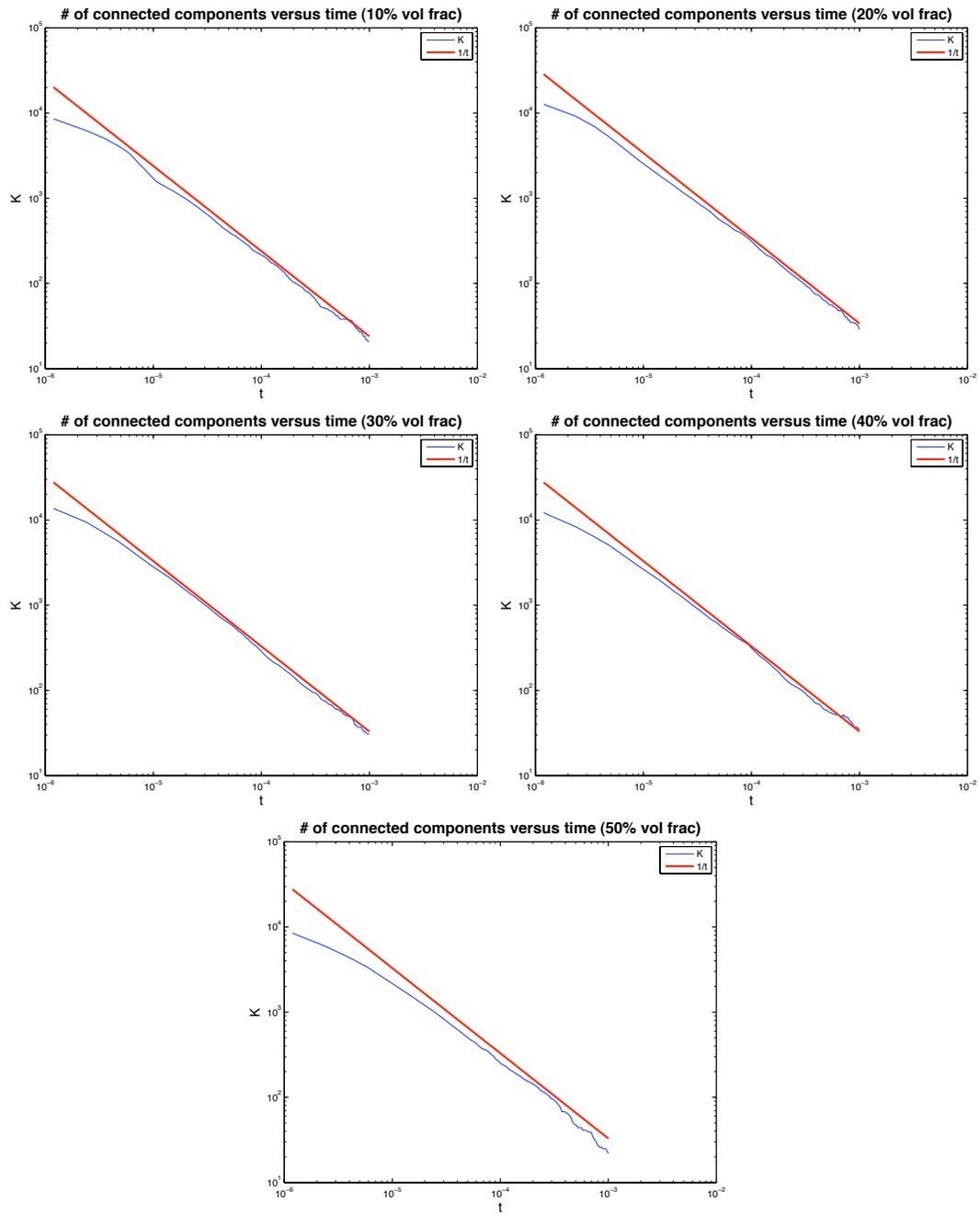


Figure 2.7: Loglog plot of the number of connected components K versus time for various area fractions. From top to bottom, and left to right, the plots correspond to area fractions ranging from 10% to 50%. The plots acknowledge good agreement between the bound $\frac{1}{t}$ (thick line) and the numerically observed rates (fine line).

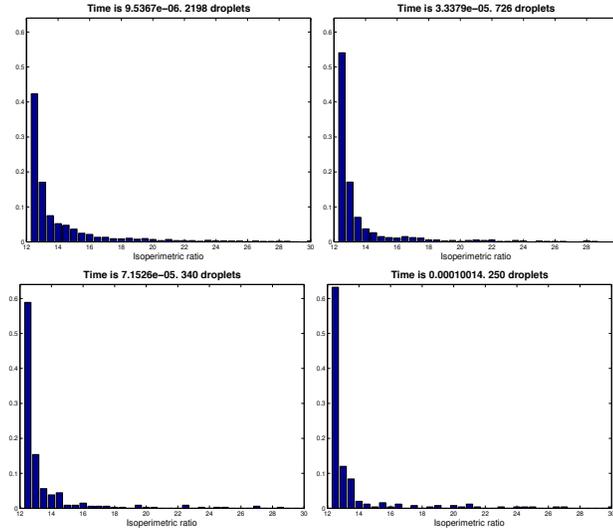


Figure 2.8: Time evolution of the distribution of isoperimetric ratios computed from the evolution of an initial configuration of droplets with 50% area fraction under area preserving curvature motion. During the evolution, the distribution of isoperimetric ratios remains quite wide. This width underlines the existence, at all times, of a certain proportion of droplets that resulted from collisions.

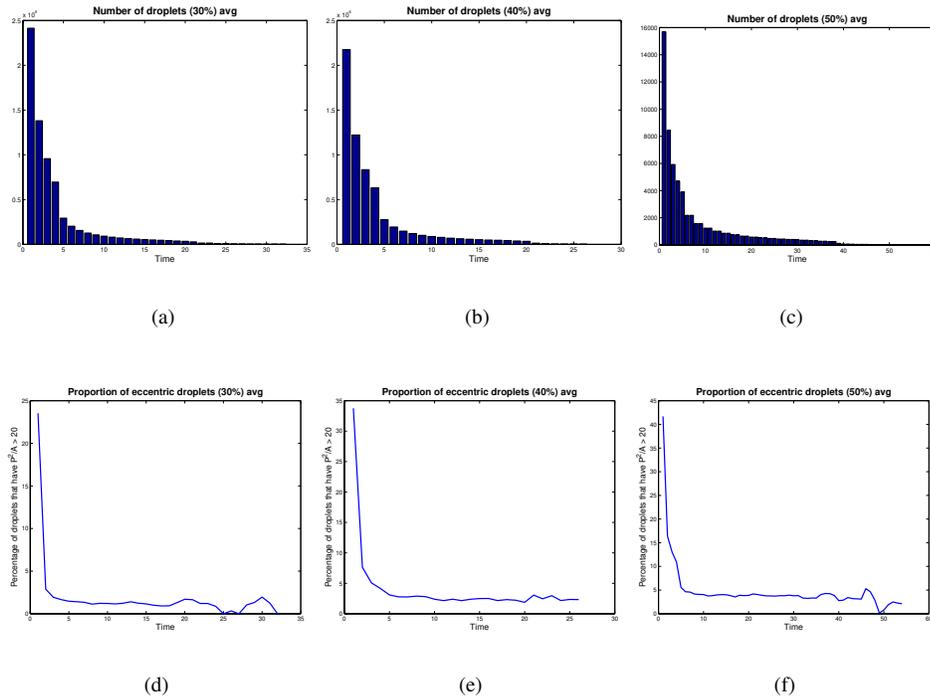


Figure 2.9: Proportions of eccentric droplets (i.e. with isoperimetric ratio $I > 20$) in the case of 30%, 40% and 50% area fraction. Figures 2.9(d), 2.9(e) and 2.9(f) clearly show that after a transient initial time, the proportion of eccentric droplets seems to stabilize around a constant value. Figures 2.9(a), 2.9(b) and 2.9(c) corroborate the fact that the proportion of eccentric droplets remains constant from the time when the total number of droplets is on average 3000 until the time when it is around 50.

2.4 Application to inverse problems

In this section, we investigate the application of area preserving flows to certain inverse problems that arise in medical imaging, such as tomographic inversion. In particular, we study whether the common choice of perimeter penalty as a regularization term, for example as in the Mumford-Shah functional [70], leads to significant bias in reconstructions, and explore the use of area preserving curvature motion as a regularization that may abate such artifacts.

2.4.1 Preliminaries

The well known Mumford and Shah model [70] has been extensively used as a regularizer in inverse problems (see e.g. [4, 5, 53, 79]). When the forward map (i.e., the observed quantity) is modeled as

$$(2.10) \quad g(y) = (Af_0)(y) + n(y),$$

where f_0 is the unknown original image to be recovered, A is an operator and n is noise, these models have the general form

$$(2.11) \quad E(u, \Gamma) = \lambda \int_{\Omega} (g - Au)^2 dy + \mu \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + |\Gamma|,$$

where Γ is the discontinuity set of u and $|\Gamma|$ the total length of the arcs making up Γ . The minimizer u_0 of (2.11) gives an approximation to the original image f_0 . In a number of applications, the original image can be assumed to be well approximated either by a piecewise constant or a simpler piecewise smooth image. In those cases, simplified versions such as piecewise constant Mumford-Shah (see for example the Chan-Vese model in [23]) constitute popular alternatives to (2.11).

We consider images f_0 that can be accurately approximated by multi-phase piecewise

smooth functions u . In particular, we consider two-phase and four-phase approximations,

$$(2.12) \quad u(x) = c_1(x)\mathbf{1}_{\Sigma}(x) + c_2(x)\mathbf{1}_{\Sigma^c}(x),$$

and

$$(2.13) \quad u(x) = c_1(x)\mathbf{1}_{\Sigma_1 \cap \Sigma_2^c \cap \Sigma_3^c}(x) + c_2(x)\mathbf{1}_{\Sigma_2}(x) + c_3(x)\mathbf{1}_{\Sigma_3}(x) + c_4(x)\mathbf{1}_{\Sigma_1^c}(x),$$

where $\Sigma_i \subseteq \Omega$ may be multiply connected, and c_i are smooth functions. The particular choice of four-phase solution (2.13) is motivated by the specific characteristics of typical images of CT scans (e.g., lungs) as illustrated in Figure 2.10. In the case where the func-

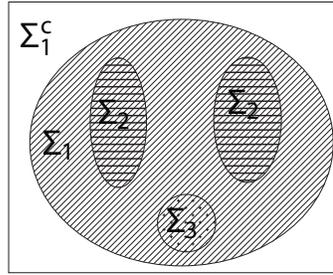


Figure 2.10: Illustration of the four-phase image model described in equation (2.13).

tions c_i are constant, equations (2.12) and (2.13) represent piecewise constant solutions.

Note that there are several ways of writing a four-phase solution, one obvious way being

$$u(x) = c_1\mathbf{1}_{\Sigma_1 \cap \Sigma_2}(x) + c_2\mathbf{1}_{\Sigma_1 \cap \Sigma_2^c} + c_3\mathbf{1}_{\Sigma_1^c \cap \Sigma_2} + c_4\mathbf{1}_{\Sigma_1^c \cap \Sigma_2^c}$$

which only uses two sets Σ_1 and Σ_2 . This four-phase model was used for example in [38].

For simplicity in the next two sections, we restrict our exposition to two-phase solutions. The generalization to four phases follows easily from the two-phase case.

2.4.2 Mumford-Shah based flows and their area preserving analogues

Within the setting introduced in the previous section, we now describe two related models. The first one is variational and a variant of the Mumford-Shah functional (thus using perimeter regularization). The second one inspires itself from the first one but replaces

the curvature motion term (obtained from perimeter regularization) by the area preserving curvature motion.

Mumford-Shah based flows

We consider observed images g as in (2.10), where A is a linear operator (e.g., partial Fourier or Radon transform). We will also look at the special case $A = I$, which is the denoising problem. In the general case, and using our model (2.12), we consider the following minimization:

$$(2.14) \quad \min_{\Sigma, c_1, c_2} \left\{ E(\Sigma, c_1, c_2) := \text{Per}(\Sigma) + \lambda \int_{\Omega'} (g(y) - (A [c_1(x)\mathbf{1}_\Sigma + c_2(x)\mathbf{1}_{\Sigma^c}])(y))^2 dy \right. \\ \left. + \mu \left(\int_{\Omega} |\nabla c_1|^2 dx + \int_{\Omega} |\nabla c_2|^2 dx \right) \right\},$$

where the operator A maps functions defined on Ω to functions defined on Ω' . In the limit $\mu \rightarrow \infty$, (2.14) reduces to the simpler minimization

$$(2.15) \quad \min_{\Sigma, c_1, c_2} \left\{ E(\Sigma, c_1, c_2) := \text{Per}(\Sigma) + \lambda \int_{\Omega'} (g(y) - (A [c_1\mathbf{1}_\Sigma + c_2\mathbf{1}_{\Sigma^c}])(y))^2 dy \right\},$$

where c_i are constant. We remark that the energy written in (2.14) is not the Mumford-Shah model (2.11) restricted to two-phase, piecewise smooth solutions. The difference lies in the regularization terms; in our case, the Dirichlet energy of the functions c_i are integrated over the whole domain Ω , whereas in the Mumford-Shah model, the Dirichlet energy of the functions c_i are integrated over the set of points $x \in \Omega$ on which the solution u takes the values $c_i(x)$. The latter was considered by Vese and Chan with $A = I$ as an improvement over the piecewise constant Mumford-Shah model [96]. However, their piecewise smooth Mumford-Shah model is not lower semi-continuous, and is thus likely to be ill-posed (see Appendix A). To regain lower semi-continuity, it is necessary to consider the lower semi-continuous envelope of their piecewise smooth model, which unfortunately turns out to be like the full Mumford-Shah model. The simplification made in the piecewise smooth

model is thus lost in its lower semi-continuous envelope. In contrast, our model (2.14) is well-posed. Moreover, for computational purposes, (2.14) is much easier to deal with, since unlike the Vese-Chan model, it avoids the complication of boundary conditions that need to be imposed along the moving contour.

In the next paragraph, we briefly present how to compute the minimizer of (2.14) and (2.15) using the level set method since the equations obtained from the level set formulation of (2.14) and (2.15) using gradient descent serve as a starting point for our proposed area preserving flows. We define $\partial\Sigma$ to be the interface between the sets Σ and Σ^c , and embed $\partial\Sigma$ as the zero level set of a smooth function $\phi(x, t)$, namely $\Gamma(t) = \{x \in \Omega : \phi(x, t) = 0\}$, with $\Sigma(t) = \{x \in \Omega : \phi(x, t) > 0\}$. By rewriting (2.14) in terms of ϕ , we obtain the minimization

(2.16)

$$\min_{\phi, c_1, c_2} \left\{ E(\phi, c_1, c_2) := \int_{\Omega} |\nabla H(\phi)| dx + \lambda \int_{\Omega'} \left(g(y) - A [c_1 H(\phi) + c_2 (1 - H(\phi))] (y) \right)^2 dy + \mu \left(\int_{\Omega} |\nabla c_1|^2 dx + \int_{\Omega} |\nabla c_2|^2 dx \right) \right\},$$

where H is the one-dimensional Heaviside function. Using gradient descent on E in (2.16), we deduce the following evolutions for ϕ and c_i :

$$(2.17) \quad \phi_t = |\nabla \phi| \left(\kappa + 2\lambda A^* \left(g - A [c_1 H(\phi) + c_2 (1 - H(\phi))] \right) (c_1 - c_2) \right),$$

and

$$(2.18a) \quad \frac{\partial c_1}{\partial t} = \mu \Delta c_1 + \lambda A^* \left[g - A (c_1 H(\phi) + c_2 (1 - H(\phi))) \right] H(\phi),$$

$$(2.18b) \quad \frac{\partial c_2}{\partial t} = \mu \Delta c_2 + \lambda A^* \left[g - A (c_1 H(\phi) + c_2 (1 - H(\phi))) \right] (1 - H(\phi)),$$

where $\kappa(x) = \nabla \cdot \left(\frac{\nabla \phi(x)}{|\nabla \phi|} \right)$ is the curvature of the interface Γ at the point $x \in \Gamma$, and A^* is the adjoint operator of A . In the piecewise constant case, the optimality conditions for the

constants c_i become

$$(2.19a) \quad c_1 = \frac{\int_{\Omega} A^* [g - A [c_2(1 - H(\phi))]] H(\phi) dx}{\int_{\Omega} A^* [AH(\phi)] H(\phi) dx},$$

$$(2.19b) \quad c_2 = \frac{\int_{\Omega} A^* [g - A [c_1 H(\phi)]] (1 - H(\phi)) dx}{\int_{\Omega} A^* [A(1 - H(\phi))] (1 - H(\phi)) dx}.$$

Equation (2.17) is still valid for piecewise constant solutions.

In the next section, we describe the area preserving segmentation flow and motivate its use using the Mumford-Shah based model we just described.

Area Preserving flows

The minimization of the Mumford-Shah based functional (2.14) seeks a piecewise smooth function approximating a given image by penalizing, among other things, the length of the boundaries between each of the regions contained in the image. In particular, this penalization ensures that no trivial segmentation is obtained. However, the length term penalty can sometimes lead to a shrinkage of the interface around the objects to be detected. In fact, in the absence of fidelity, the level set evolution seeking the minimizer of the Mumford-Shah based functional reduces to curvature motion. This is easy to see using equation (2.17) describing the level set evolution. This equation describes the motion of an interface $\partial\Sigma$ with normal velocity

$$(2.20) \quad v_N = \kappa + 2\lambda A^* (g - A [c_1 H(\phi) + c_2 (1 - H(\phi))]) (c_1 - c_2).$$

With no fidelity term (i.e., $\lambda = 0$), the motion reduces to the evolution of $\partial\Sigma$ with normal velocity $v_N = \kappa$, which is nothing else than motion by curvature. As noted in Chapter I, this motion shrinks curves to a point. To remove this shrinkage, we replace the curvature motion term by the area preserving curvature motion and propose the following flow:

$$(2.21) \quad \phi_t = |\nabla\phi| (\kappa - \bar{\kappa} + 2\lambda A^* (g - A [c_1 H(\phi) + c_2 (1 - H(\phi))]) (c_1 - c_2)).$$

Equation (2.21) can be described as the area preserving curvature motion combined with the flow obtained from the fidelity term in the Mumford-Shah based functional. Note that this flow applies to both piecewise constant and piecewise smooth images. In the complete area preserving update, the evolutions of c_i described in (2.18a) and (2.18b) for piecewise smooth solutions and the optimality conditions for the constants c_i given by equations (2.19a) and (2.19b) remain the same. An important point about these area preserving flows is that contrary to (2.17), it is not clear whether (2.21) is variational, in other words we do not know whether (2.21) is gradient descent for an energy in a specific metric.

Considering equation (2.21), we see that in absence of fidelity term, the evolution proceeds as a regular area preserving curvature motion. As described earlier, this motion preserves the total area of all the connected components during the evolution. The advantage of this default motion (i.e. when $\lambda = 0$) is that no shrinkage is induced by the model. However, even though the total area is preserved, this default motion will eventually merge all connected components into one component (this is the coarsening behavior observed in the previous application) with area equal to the initial total area. To remove this property (undesirable in this particular image processing application), we replace the usual area preserving flow by a component-wise area preserving flow that preserves the area of each of the connected components rather than their total area. Note that unlike the area preserving curvature motion of Section 2.3, we are not specifying the area of each region. The areas may change due to the fidelity term.

2.4.3 Algorithms

In this section, we apply the ideas developed in Section 2.2 for rapid and accurate computation of area preserving curvature flows for the Mumford-Shah based reconstruction models discussed in the previous sections.

Mumford-Shah based flows

We recall that in the general piecewise smooth case with operator, the Mumford-Shah based flow for the level set function ϕ simulates the motion of an interface defined through the level set function ϕ with normal velocity given by equation (2.20). To simulate this evolution using signed distance functions, we simply replace the level set function ϕ by the signed distance function to the interface (denoted d). At each time step in the complete algorithm, we alternate between the update of the functions c_i by solving the PDEs (2.18a) and (2.18b) for a length of time δt , and the update of the signed distance function d . For clarity in the exposition of the algorithm, we define the following update function:

$$[c_{1out}, c_{2out}] = \mathbf{Update_c}(c_1, c_2, d, g, \delta t),$$

1. Form the right-hand sides

$$\text{RHS}_1 = \text{FFT} \left\{ c_1 + \lambda \delta t A^* \left[g - A(c_1 H(d) + c_2(1 - H(d))) \right] H(d) \right\},$$

$$\text{RHS}_2 = \text{FFT} \left\{ c_2 + \lambda \delta t A^* \left[g - A(c_1 H(d) + c_2(1 - H(d))) \right] (1 - H(d)) \right\}.$$

2. Update c_1 and c_2

$$c_{1out} = \text{real} \left(\text{IFFT} \left(\frac{\text{RHS}_1}{I + \mu \delta t \Delta} \right) \right),$$

$$c_{2out} = \text{real} \left(\text{IFFT} \left(\frac{\text{RHS}_2}{I + \mu \delta t \Delta} \right) \right).$$

3. Return c_{1out} and c_{2out} .

The algorithm for the Mumford-Shah based flow is given below.

Algorithm 6. Mumford-Shah based flow. Given the initial set Σ_0 defined through its signed distance function $d_0(x)$, initial functions c_i^0 and a time step size $\delta t > 0$, generate the sets Σ_j via their signed distance function $d_j(x)$, and the functions c_i^j at subsequent discrete times $t = j(\delta t)$ by alternating the following three operations:

1. Update the functions c_1^j and c_2^j :

$$[c_1^j, c_2^j] = \mathbf{Update_c}(c_1^{j-1}, c_2^{j-1}, d_{j-1}, g, \delta t)$$

2. Form the function

$$\mathcal{L}(x) = (G_{\delta t} * d_{j-1})(x) + (\delta t)S_{j-\frac{1}{2}}(x),$$

where

$$(2.22) \quad S_{j-\frac{1}{2}} = 2\lambda A^* \left(g - A \left[c_1^j(x)H(d_{j-1}) + c_2^j(x)(1 - H(d_{j-1})) \right] \right) (c_1^j - c_2^j).$$

3. Construct the distance function d_j using

$$d_j(x) = \mathbf{Redist}(\mathcal{L}(x)).$$

Area preserving flows

At each time step in the complete algorithm for the area preserving flows, we alternate between the update of the functions c_i and the update of the signed distance function d . As discussed earlier, the functions c_i in the area preserving flows follow the same updates as in the Mumford-Shah based flow. The complete algorithm for area preserving flow builds on Algorithm 5 and uses the function ComponentwiseAP described in Section 2.2.

Algorithm 7. Area Preserving flow. Given the initial set Σ_0 defined through its signed distance function $d_0(x)$, initial functions c_i^0 and a time step size $\delta t > 0$, generate the sets Σ_j via their signed distance function $d_j(x)$, and the functions c_i^j at subsequent discrete times $t = j(\delta t)$ by alternating

1. Update the functions c_1^j and c_2^j :

$$[c_1^j, c_2^j] = \mathbf{Update_c}(c_1^{j-1}, c_2^{j-1}, d_{j-1}, g, \delta t)$$

2. Form the functions

$$(2.23) \quad \begin{aligned} \mathcal{L}(x) &= (G_{\delta t} * d_j)(x) + (\delta t)S_{j-\frac{1}{2}}(x) \\ \mathcal{L}_o(x) &= \mathbf{ComponentwiseAP}(\mathcal{L}, d_j, \delta t) \end{aligned}$$

where $S_{j-\frac{1}{2}}$ is given by equation (2.22).

3. Construct signed distance function d_j by

$$d_j(x) = \mathbf{Redist}(\mathcal{L}_o(x)).$$

Proposed implementation

In this section we describe a specific part of the implementation of Algorithms 6 and 7 that involves replacing step 2 with the solution of a linear parabolic PDE. This approach, introduced by Esedoğlu and Tsai in [38], aims at alleviating numerical instabilities created by the use of large δt , or large fidelity constant λ (from the additional term S). As described in the two previous sections, the second step of Algorithms 6 and 7 involves convolving the signed distance function d_j with a Gaussian kernel, and then translating the resulting level set by $(\delta t)S_{j-\frac{1}{2}}$ (and $-\bar{\kappa}_j(\delta t)$ in the area preserving case). When δt or λ are large, this translation can lead to numerical instabilities.

Following Esedoğlu and Tsai's idea [38], we replace the diffusion step of our algorithm

by the solution of the following initial value problem

$$(2.24) \quad \begin{cases} u_t = \Delta u + 2\tilde{\lambda}(T_b - (T_a + T_b)u), \\ u(x, 0) = d_\Sigma(x) + \frac{1}{2}, \end{cases}$$

where the terms T_a and T_b depend on the fidelity term; in particular, they only depend on the functions c_i , the given image and the parameters λ and μ . In particular, T_a and T_b take on different values, depending on which type of solution and which type of model we are using, namely piecewise smooth versus piecewise constant, two-phase versus four-phase, and operator ($A \neq I$) versus identity operator. Note that the initial condition of (2.24) is a shifted signed distance function to the interface $\partial\Sigma$. The shift, and in particular the value of the shifting constant will be made clear in the next calculations. This shift is necessary to obtain the correct fidelity term. We present the PDE method for general T_a and T_b .

We first consider the one-dimensional version of (2.24)

$$(2.25) \quad \begin{cases} \psi_t = \psi_{xx} + 2\tilde{\lambda}(T_b - (T_a + T_b)\psi) \text{ on } \mathbb{R} \times \mathbb{R}^+ \\ \psi(x, 0) = d_R(x) + \frac{1}{2} = \tilde{d}_R(x), \end{cases}$$

where $\tilde{d}_R(x)$ is the signed distance function to the interface $\{0\} \cup \{R\}$ for $R > 0$, i.e., $\{0\} \cup \{R\}$ is the zero level set of $\tilde{d}_R(x)$. Figure 2.11 shows the graph of the 1D signed distance function \tilde{d}_R . The solution of (2.25) is given by

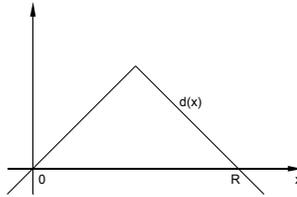


Figure 2.11: One dimensional signed distance function \tilde{d}_R to the interface $\{0\} \cup \{R\}$.

$$\psi(x, t) = e^{-2\tilde{\lambda}(T_a+T_b)t} \left\{ (G_t * \tilde{d}_R)(x) + \frac{T_b}{T_a + T_b} (e^{2\tilde{\lambda}(T_a+T_b)t} - 1) \right\},$$

where G_t is the one dimensional Gaussian kernel given in (1.19). The goal of the next calculations is to determine the location of the interface $\partial\Sigma$ after a time δt , where δt is assumed to be small. In particular, starting from the interface at $x = 0$ at time $t = 0$, we compute its new location $\ell(\delta t)$ at time δt . For this, we assume that since δt is small, the solution ψ is a shifted signed distance function to the new interface at δt . At $x = 0$ we have $(G_t * \tilde{d}_R)(0) = \frac{1}{2} + (G_t * d_R)(0) + O(\delta t^2) = \frac{1}{2} + O(\delta t^2)$ (see [36] for expansions of $(G_t * d_R)$ around 0). We therefore obtain

$$\begin{aligned}\psi(0, \delta t) &= \left(1 - 2\tilde{\lambda}(T_a + T_b)\delta t + O(\delta t^2)\right) \left\{ \frac{1}{2} + \frac{T_b}{T_a + T_b} \left(2\tilde{\lambda}(T_a + T_b)\delta t + O(\delta t^2)\right) \right\} \\ &= \frac{1}{2} + 2\tilde{\lambda}T_b\delta t - \tilde{\lambda}(T_a + T_b)\delta t + O(\delta t^2) \\ &= \frac{1}{2} + \tilde{\lambda}(T_b - T_a)\delta t + O(\delta t^2).\end{aligned}$$

Thus

$$(2.26) \quad \psi(0, \delta t) \approx \frac{1}{2} + \tilde{\lambda}(T_b - T_a)\delta t.$$

Now we look at the spatial derivative of ψ around $x = 0$

$$\partial_x \psi(x, \delta t) = e^{-2\tilde{\lambda}(T_a + T_b)\delta t} \frac{\partial(G_t * \tilde{d}_R)(x)}{\partial x},$$

with

$$\frac{\partial(G_t * \tilde{d}_R)(x)}{\partial x} = \frac{-1}{2\delta t \sqrt{4\pi\delta t}} \int_{-\infty}^{\infty} (x-y)e^{\frac{(x-y)^2}{4\delta t}} \tilde{d}_R(y) dy.$$

Since $\tilde{d}_R(y) = y + \frac{1}{2}O(\delta t^2)$ around $y = 0$, we conclude that

$$\begin{aligned}\frac{\partial(G_t * \tilde{d}_R)(0)}{\partial x} &= \frac{1}{2\delta t} \underbrace{\frac{1}{\sqrt{4\pi\delta t}} \int_{-\infty}^{\infty} y^2 e^{-\frac{y^2}{4\delta t}} dy}_{=2\delta t} + \frac{1}{4\delta t} \underbrace{\frac{1}{\sqrt{4\pi\delta t}} \int_{-\infty}^{\infty} ye^{-\frac{y^2}{4\delta t}} dy}_{=0} + O(\delta t) \\ &= 1 + O(\delta t).\end{aligned}$$

Thus

$$(2.27) \quad \partial_x \psi(0, \delta t) \approx e^{-2\tilde{\lambda}(T_a + T_b)\delta t} \approx 1.$$

If we take δt small enough, the solution ψ of (2.25) with the shifted signed distance function \tilde{d}_R taken as initial condition should still be close to the same shifted signed distance function. Thus, starting with an interface located at $x = 0$, we want to measure the location of the interface at time δt . The location of the interface at time δt is noted $\ell(\delta t)$, and by definition satisfies $\psi(\ell(\delta t), \delta t) = \frac{1}{2}$ (the interface at δt is situated on the $\frac{1}{2}$ level set of the signed distance function ψ). We now expand $\psi(\ell(\delta t), \delta t)$ around $x = 0$ to obtain

$$\psi(\ell(\delta t), \delta t) = \psi(0, \delta t) + \ell \partial_x \psi(0, \delta t) + O(|\ell|^2).$$

Since $\ell(\delta t)$ is defined such that $\psi(\ell(\delta t), \delta t) = \frac{1}{2}$, the interface located at $x = 0$ at $t = 0$ moves to the location

$$(2.28) \quad \ell(\delta t) \approx \tilde{\lambda}(T_a - T_b) \delta t,$$

at $t = \delta t$.

In two dimensions, the initial value problem (2.25) becomes

$$(2.29) \quad \begin{cases} u_t = \Delta u + 2\tilde{\lambda}(T_b - (T_a + T_b)u) \text{ on } \mathbb{R} \times \mathbb{R}^+ \\ u(x, 0) = d_\Sigma(x) + \frac{1}{2} = \tilde{d}_\Sigma(x), \end{cases}$$

In the same way, we consider the solution of (2.29) for a short time δt in a neighborhood of a point $x \in \partial\Sigma$. Following [38], we consider the ansatz

$$(2.30) \quad u(x, t) = \psi(\tilde{d}_{\Sigma(t)}(x), t)$$

as a candidate for the solution of (2.29). Using (2.30) we obtain

$$(2.31) \quad \begin{aligned} u_t(x, t) &= \psi_x(\tilde{d}_{\Sigma(t)}(x), t) \left(\frac{\partial \tilde{d}_{\Sigma(t)}(x)}{\partial t} \right) + \psi_t(\tilde{d}_{\Sigma(t)}(x), t) \\ &= \psi_x(\tilde{d}_{\Sigma(t)}(x), t) \left(\frac{\partial \tilde{d}_{\Sigma(t)}(x)}{\partial t} \right) + \psi_{xx}(\tilde{d}_{\Sigma(t)}(x), t) + 2\tilde{\lambda}(T_b - (T_a + T_b)\psi(x, t)). \end{aligned}$$

In addition, we have

$$(2.32) \quad \Delta u(x, t) = \psi_x(\tilde{d}_{\Sigma(t)}(x), t) \Delta \tilde{d}_{\Sigma(t)}(x) + \psi_{xx}(\tilde{d}_{\Sigma(t)}(x), t) |\nabla \tilde{d}_{\Sigma(t)}(x)|^2.$$

Since \tilde{d} is a signed distance function, it satisfies $|\nabla \tilde{d}_{\Sigma(t)}(x)|^2 = 1$. Moreover, since \tilde{d} is a shifted signed distance function to $\partial\Sigma$, we know that on the curve $\partial\Sigma$, $\Delta \tilde{d}_{\Sigma(t)}(x)$ is equal to the curvature. Combining (2.31) and (2.32), we obtain

(2.33)

$$u_t(x, t) = \Delta u(x, t) + 2\tilde{\lambda}(T_b - (T_a + T_b)u(x, t)) + \psi_x(\tilde{d}_{\Sigma(t)}(x), t) \left(\frac{\partial \tilde{d}_{\Sigma(t)}(x)}{\partial t} - \Delta \tilde{d}_{\Sigma(t)}(x) \right).$$

Hence, if the curve $\partial\Sigma(t)$ evolves under curvature motion described by

$$(2.34) \quad \frac{\partial \tilde{d}_{\Sigma(t)}(x)}{\partial t} = \Delta \tilde{d}_{\Sigma(t)}(x) = \kappa(\partial\Sigma(t)),$$

then ansatz (2.30) satisfies equation (2.29) on $\partial\Sigma(t)$. Now we consider the parameterized curve Γ evolving under the following motion

$$(2.35) \quad \begin{cases} \frac{\partial \Gamma(t)}{\partial t} = [\kappa(\Gamma(t)) + \tilde{\lambda}(T_b - T_a)] \vec{N}(\Gamma(t)) \\ \Gamma(0) = \partial F(0), \end{cases}$$

where the curve $\partial F(t)$ (the set $F(t)$ is a subset of \mathbb{R}^2) evolves under curvature motion given in (2.34). Note that the second term in the normal velocity under which the curve Γ is evolving, is the speed of the one-dimensional profile obtained in equation (2.28). Using the fact that $\partial F(t)$ moves under curvature motion, and using the result of equation (2.28) for the one-dimensional profile, we obtain

$$(2.36) \quad \left\{ x \in \mathbb{R}^2 : u(x, \delta t) = \frac{1}{2} \right\} \approx \Gamma(x, \delta t).$$

Thus, for small enough time δt , the half level set of the solution u of the initial value problem (2.29), evolves the interface $\partial\Sigma(t)$ from a given initial interface $\partial\Sigma$, with normal velocity $v_N = \kappa + \tilde{\lambda}(T_b - T_a)$. The half level set of u becomes the zero level set of \tilde{u} by the shift $\tilde{u}(x, t) = u(x, t) - \frac{1}{2}$. From equation (2.17) we can relate the parameter $\tilde{\lambda}$ with the fidelity parameter λ as

$$\tilde{\lambda} = 2\lambda.$$

with

$$(2.37) \quad \begin{aligned} T_a &= 0, \\ T_b &= A^* [g - A [c_1 H(\phi) + c_2 (1 - H(\phi))]] (c_1 - c_2). \end{aligned}$$

If $A = I$, we have $\tilde{\lambda} = \lambda$ and

$$(2.38) \quad \begin{aligned} T_a &= (c_1 - g)^2, \\ T_b &= (c_2 - g)^2. \end{aligned}$$

Note that (2.37) does not reduce to (2.38) when $A = I$. The result obtained in (2.38) uses a few simplifications that are not possible in the more general case with operator. The area preserving algorithm using the PDE implementation is given below. The Mumford-Shah based algorithm is obtained similarly.

Algorithm 8. Area Preserving flow with PDE implementation. Given the initial set Σ_0 defined through its signed distance function $d_0(x)$, initial functions c_i^0 and a time step size $\delta t > 0$, generate the sets Σ_j via their signed distance functions $d_j(x)$, and the functions c_i^j at subsequent discrete times $t = j(\delta t)$ by alternating the following steps:

1. Update the functions c_1^j and c_2^j :

$$[c_1^j, c_2^j] = \mathbf{Update_c}(c_1^{j-1}, c_2^{j-1}, d_{j-1}, g, \delta t)$$

2. Compute the solution $\mathcal{L}(x, \delta t)$ to the initial value problem (2.29):

$$\begin{cases} \mathcal{L}_t = \Delta \mathcal{L} + 2\tilde{\lambda}(T_b - (T_a + T_b)\mathcal{L}) \text{ on } \mathbb{R} \times \mathbb{R}^+ \\ \mathcal{L}(x, 0) = d_{\Sigma}(x) + \frac{1}{2} = \tilde{d}_{\Sigma}(x), \end{cases}$$

$$\text{with } T_a = 0 \text{ and } T_b = A^* \left[g - A \left[c_1^j H(d_{j-1}) + c_2^j (1 - H(d_{j-1})) \right] \right] (c_1^j - c_2^j).$$

3. Form \mathcal{L}_o using

$$\mathcal{L}_o(x) = \mathbf{ComponentwiseAP}(\mathcal{L}, d_j, \delta t)$$

4. Construct signed distance function d_j by

$$d_j(x) = \mathbf{Redist}(\mathcal{L}_o(x)).$$

Using the same technique, we can obtain the expressions of T_a and T_b for the four-phase model.

2.4.4 Comparisons and applications

In this section, we compare the performance of the Mumford-Shah based model with the performance of the area preserving flow on a simple theoretical example, and in numerical experiments.

A simple example

In this section we consider a very simple but illustrative example on which we compare the performance of the two-phase, piecewise constant Mumford-Shah based flow, with the performance of the two-phase, piecewise constant area preserving flow in the setting where $A = I$ in (2.21). In particular, the goal is to demonstrate that there exists a range of values for the parameter λ for which the Mumford-Shah based flow will introduce a shrinkage. On the other hand, on the same image, the area preserving flow will never introduce a shrinkage, regardless of the choice of parameter λ . We consider a simple example on which the area preserving flow is equivalent to the 2-means algorithm, namely we consider an image with a circular object. It is easy to see that if the image contains a circular object, then its boundary will be a circle and its curvature will be equal to its average curvature. Thus if we start the area preserving flow with a circle as initial condition, the circle will remain a circle during the entire evolution (by symmetry), and the term $\kappa - \bar{\kappa}$ in the normal velocity of the area preserving flow will therefore be zero. In this case, the normal velocity reduces to

$$v_N = \lambda \left((c_2 - g)^2 - (c_1 - g)^2 \right),$$

and is nothing else than the 2-means algorithm, looking for 2 regions in the image, with values c_1 in one and c_2 in the other. In this case, there is no geometric term. We can see already that the sign of v_N will determine in which direction the contour will move, regardless of the choice of λ (assuming $\lambda > 0$). In this particular example, the sign of the difference $\left((c_2 - g)^2 - (c_1 - g)^2 \right)$ for any $\lambda > 0$, will determine the direction of evolution. If $\lambda = 0$, the radius of the final contour will be the radius of the circle with the same area as the initial contour.

We consider the image domain Ω to be the ball of radius R centered at 0, denoted

$B(0, R)$, and f_0 to be the image defined in polar coordinates on Ω as

$$(2.39) \quad f_0(r, \theta) = \begin{cases} I_1 & \text{if } 0 \leq r < r_1, \\ a - \frac{r}{b} & \text{if } r_1 \leq r < r_2, \\ 0 & \text{if } r_2 \leq r \leq R, \end{cases}$$

where $a = \frac{r_2 I_1 - r_1 I_2}{r_2 - r_1}$, $b = \frac{r_2 - r_1}{I_1 - I_2}$, and r_1 , r_2 , I_1 and I_2 are constants to be adjusted. By construction f_0 is rotationally invariant and can be obtained by rotating the profile graphed in Figure 2.12. Figure 2.13 shows the image f_0 from two different perspectives: as a grayscale image and as a surface.

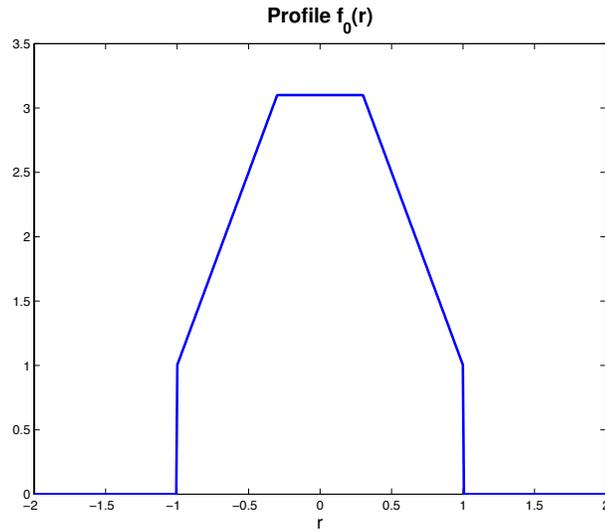


Figure 2.12: Profile $f_0(r) = f_0(r, \theta)$ for any $0 \leq \theta < 2\pi$.

The object we want to detect is the clear disk with external boundary being the circle of radius r_2 (largest circle) centered at zero. Thus we would like to have a model that finds this clear disk without introducing a shrinkage. We will show that for a range of values of the parameter λ , the minimizer of the Mumford-Shah based flow is a circle of radius $r_{min} < r_2$, thus demonstrating that the Mumford-Shah based flow does indeed introduce a shrinkage. On the other-hand, for any choice of fidelity parameter $\lambda > 0$, the area preserving flow will not introduce a shrinkage.

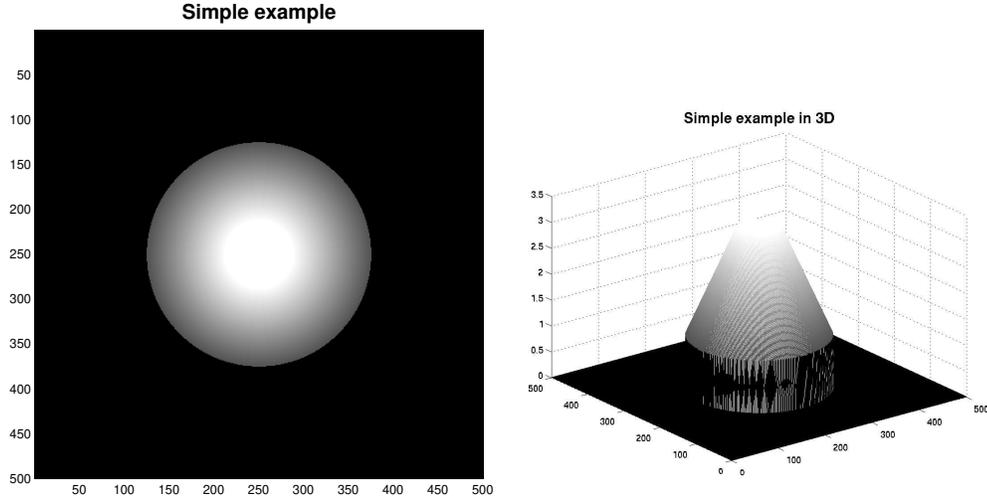


Figure 2.13: Left: gray scale display of image f_0 . Right: surface plot of image f_0 .

We now calculate the expression of the two-phase, piecewise constant Mumford-Shah based energy functional, denoted E_{MS_t} , in the case where the observed image is given by f_0 in (2.39). By symmetry, (since f_0 is rotationally invariant), we know that the minimizer Σ of E_{MS_t} will be a disk of radius r_{min} centered at 0, ($0 < r_{min} \leq R$). Thus, to determine r_{min} , we work out the expression of E_{MS_t} in the case where Σ is a disk of radius r centered at 0, and minimize E_{MS_t} with respect to r . In the piecewise constant case with $A = I$, the optimal constants c_1 and c_2 are given by

$$c_1 = \frac{\int_{\Sigma} g dx}{|\Sigma|}, \text{ and } c_2 = \frac{\int_{\Sigma^c} g dx}{|\Sigma^c|},$$

which are a particular case of equations (2.19a) and (2.19b). In the case where $g = f_0$ and Σ is a disk of radius r centered at 0, we obtain the following optimal constants c_1 and c_2 :

$$(2.40) \quad c_1(r) = \begin{cases} I_1 & \text{if } 0 \leq r < r_1, \\ \frac{1}{r^2} \left[I_1 r_1^2 + 2 \left(\frac{a}{2} (r^2 - r_1^2) - \frac{r^3 - r_1^3}{3b} \right) \right] & \text{if } r_1 \leq r < r_2, \\ \frac{1}{r^2} \left[I_1 r_1^2 + \frac{r_2 - r_1}{3} (r_2 (1 + 2I_2) + r_1 (2 + I_2)) \right] & \text{if } r_2 \leq R, \end{cases}$$

and

$$(2.41) \quad c_2(r) = \begin{cases} \frac{1}{R^2-r^2} \left[I_1 (r_1^2 - r^2) + 2 \left(\frac{a}{2} (r_2^2 - r_1^2) - \frac{r_2^3 - r_1^3}{3b} \right) \right] & \text{if } 0 \leq r < r_1, \\ \frac{1}{(R^2-r^2)} \left[2 \left(\frac{a}{2} (r_2^2 - r^2) - \frac{r_2^3 - r^3}{3b} \right) \right] & \text{if } r_1 \leq r < r_2, \\ 0 & \text{if } r_2 \leq R. \end{cases}$$

With expressions (2.40) and (2.41) for the constants c_1 and c_2 , we calculate the exact expression of $E_{MSI}(r)$, in the case where $\Sigma = B(0, r)$, and obtain

$$(2.42) \quad E_{MSI}(r) = \begin{cases} 2r + \lambda (c_2(r) - I_1)^2 (r_1^2 - r^2) + 2\lambda g_2(r_2, r_1) + \lambda c_2(r)^2 (R^2 - r_2^2) & \text{if } 0 \leq r < r_1, \\ 2r + \lambda (c_1(r) - I_1)^2 r_1^2 + 2\lambda [g_1(r, r_1) + g_2(r_2, r)] + \lambda c_2(r)^2 (R^2 - r_2^2) & \text{if } r_1 \leq r < r_2, \\ 2r + \lambda r_1^2 (c_1(r) - I_1)^2 + 2\lambda \left[\frac{r_2^4 - r^4}{4b^2} + \frac{2(c_1(r) - a)(r_2^3 - r_1^3)}{3b} + \frac{1}{2} (c_1(r) - a)^2 (r_2^2 - r_1^2) \right] & \text{if } r_2 \leq R. \end{cases}$$

where $g_i(r, p) = \frac{r^4 - p^4}{4b^2} + \frac{2(c_i(r) - a)(r^3 - p^3)}{3b} + \frac{1}{2} (c_i(r) - a)^2 (r^2 - p^2)$, for $i = 1, 2$.

Minimizing (2.42) with respect to r , gives us r_{min} , the optimal radius for the minimizing disk. Since (2.42) was too complicated to minimize exactly, we graphed the energy E_{MSI} for various choices of images g , namely different choices of r_1 , r_2 , I_1 and I_2 , and different choices of parameter λ , and found images g on which the minimizer r_{min} was strictly smaller than r_2 for a nontrivial range of λ values. Figure 2.14 was obtained for the particular choices of $r_1 = 0.3$, $r_2 = 1$, $I_1 = 3.1$ and $I_2 = 1$. The left-hand graph of Figure 2.14 shows the energy E_{MSI} for the following choice of parameter λ : $\lambda = 0, 0.39, 0.61, 0.99$. We see that for the first two choices of λ , the minimizer r_{min} is equal to zero, which means that the initial contour shrinks to zero and no object is detected. On the other hand, if $\lambda = 0.39$, $r_{min} = 0.71 < r_2 = 1$ and for $\lambda = 0.99$, $r_{min} = 0.84 < r_2 = 1$, which in both cases shows a shrinkage in the final contour. The right-hand side graph shows the evolution of the minimizer r_{min} of the Mumford-Shah based energy functional, and the radius found by the area preserving flow. Both graph are plotted for $0 \leq \lambda \leq 10$. We see that in the area preserving case, the minimizer r_{min} is always $r_2 = 1$ (except in the absence of fidelity, i.e.,

$\lambda = 0$ case). On the other hand, we clearly see that the minimizer of the Mumford-Shah based energy grows towards r_2 before reaching it for a certain value of λ (in this example, the critical value of λ is around 10). The advantage of the area preserving flow is that there is no adjustment of the parameter λ in order to avoid shrinkage, which makes it more robust.

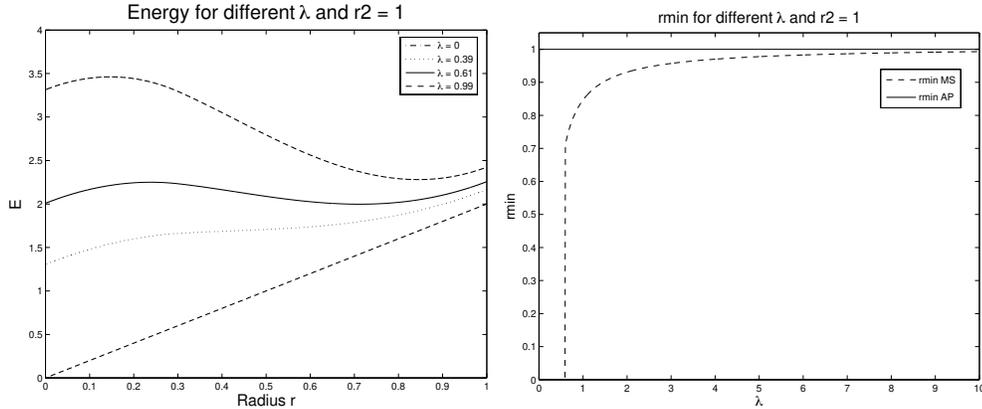


Figure 2.14: The left-hand graph shows the Mumford-Shah based energy E_{MS} for $\lambda = 0, 0.39, 0.61, 0.99$. For $\lambda < 0.6$, the minimizer r_{min} is just zero. For $\lambda = 0.61$, the minimizer r_{min} is $0.71 < r_2 = 1$ and for $\lambda = 0.99$, the minimizer r_{min} is $0.84 < r_2 = 1$. The right-hand graph shows the evolution of r_{min} as a function of λ , ($0 \leq \lambda \leq 10$), for the Mumford-Shah based flow (dashed line) and the area preserving flow (solid line).

Figure 2.15 shows the simple image f_0 superimposed with the final contours obtained from the Mumford-Shah based flow and the area preserving flow in the case $\lambda = 0.61$. The contours are graphed using the exact values of r_{min} obtained above. These are not the result of numerical experiments.

Numerical results

In this section, we focus on studying the effect of standard perimeter regularizations on image reconstructions given from measurements. In particular, we compare the performance of the Mumford-Shah based flow (using perimeter regularization) to the one of our area preserving flow on simple scenarios inspired from medical imaging applications. The results demonstrate that perimeter regularization does indeed introduce a shrinking

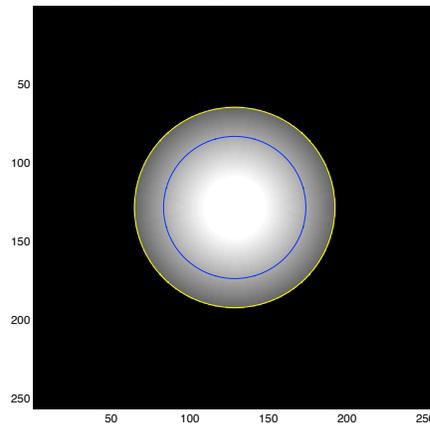
Simple example with MS type and AP contours for $\lambda = 0.61$ 

Figure 2.15: Simple image f_0 with the final contours graphed for $\lambda = 0.61$. The yellow circle is the final contour for the area preserving flow and the blue circle is the minimizer of the Mumford-Shah based energy functional. We see that for this particular value of λ , the Mumford-Shah based flow introduces quite a significant shrinkage.

bias. Figure 2.16 illustrates this point on a simple inverse problem. In this example, we reconstructed the image using only 11% of the Fourier coefficients and used noiseless data. We repeated the test with noisy data and reconstructed the image with 35% of the noisy Fourier coefficients. The noisy example is illustrated in Figure 2.18. Figures 2.17 and 2.19 show the original image superimposed with the contours of the objects found by the Mumford-Shah based flow and the area preserving flow. In these computations, we used the four-phase, piecewise smooth model.

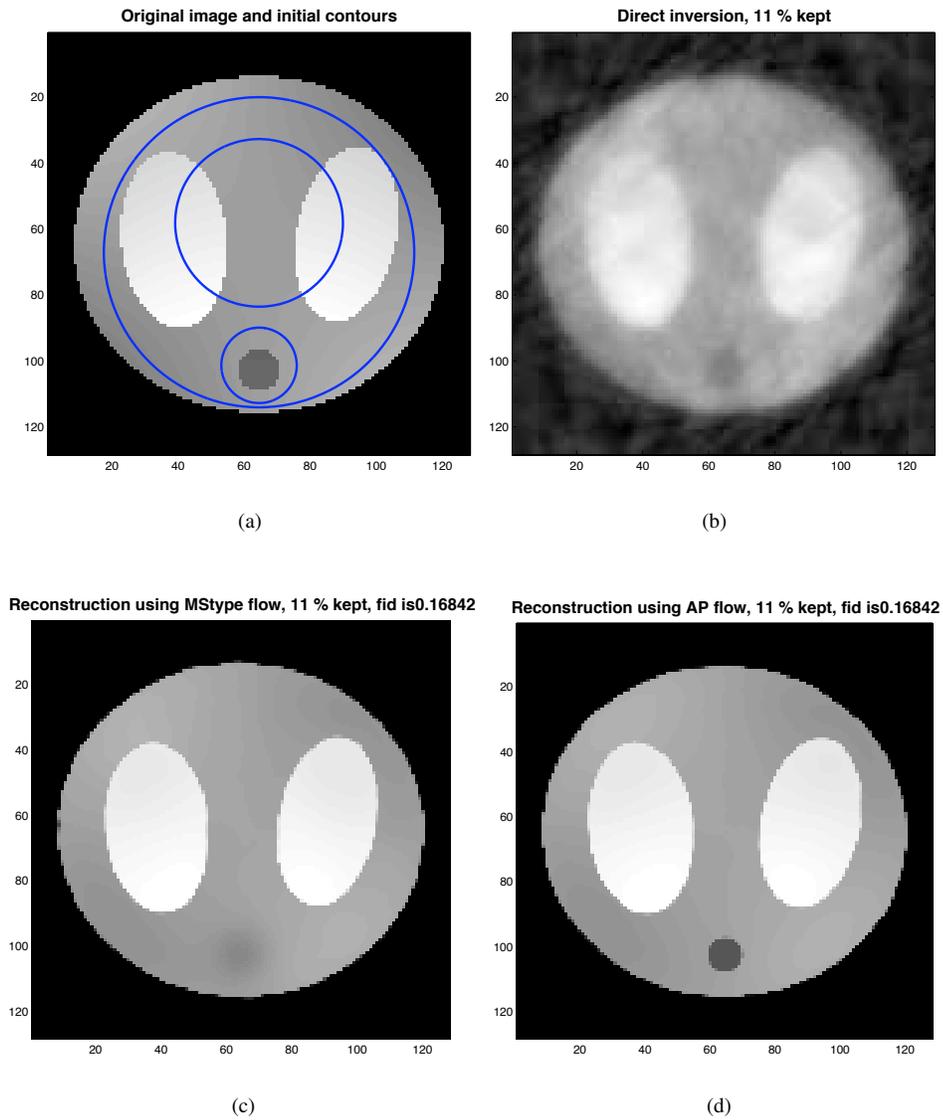


Figure 2.16: Synthetic image reconstructions comparing the loss of small scale features due to regularizations. 2.16(a): original image superimposed with the initial curves, 2.16(b): direct Fourier inversion of the observations (i.e. 11 % of the Fourier coefficients), 2.16(c): reconstruction obtained with the Mumford-Shah based flow (the boundaries of the small feature shrunk to zero), 2.16(d): reconstruction obtained using the area preserving flow.

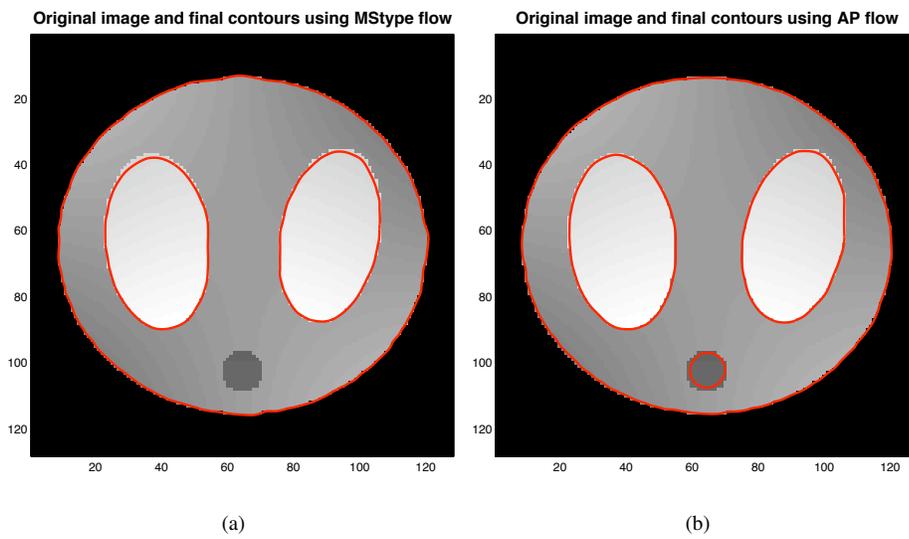


Figure 2.17: Contours found by the Mumford-Shah based flow and by the area preserving flow. Figure 2.17(a) shows the contours obtained by the Mumford-Shah based model and Figure 2.17(b) displays the contours obtained by the area preserving flow.

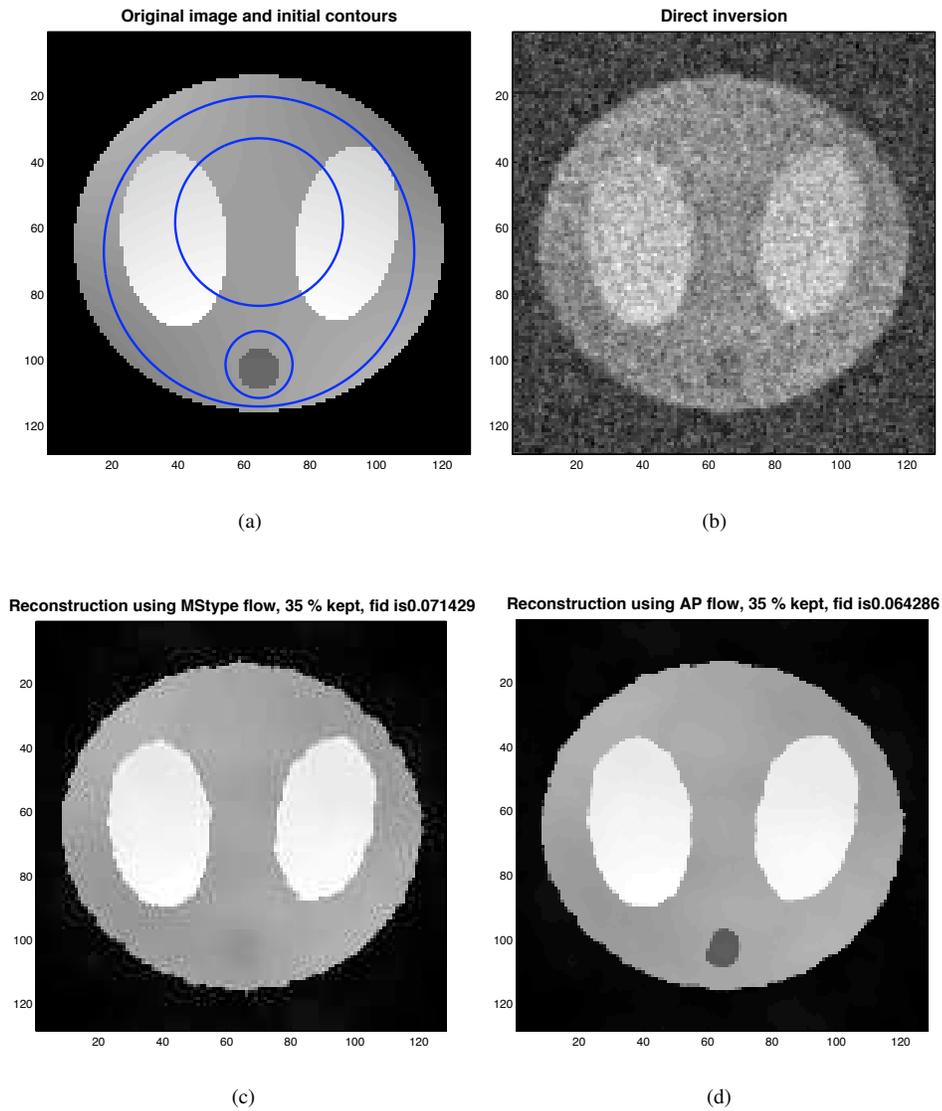


Figure 2.18: Synthetic image reconstructions comparing the loss of small scale features due to regularizations. 2.18(a): original image superimposed with the initial curves, 2.18(b): direct Fourier inversion of the observations (i.e. 35 % of the noisy Fourier coefficients), 2.18(c): reconstruction obtained with the Mumford-Shah based flow (the boundaries of the small feature shrunk to zero), 2.18(d): reconstruction obtained using the area preserving flow. In this example, the SNR of the observed datum g is 3.88 dB.

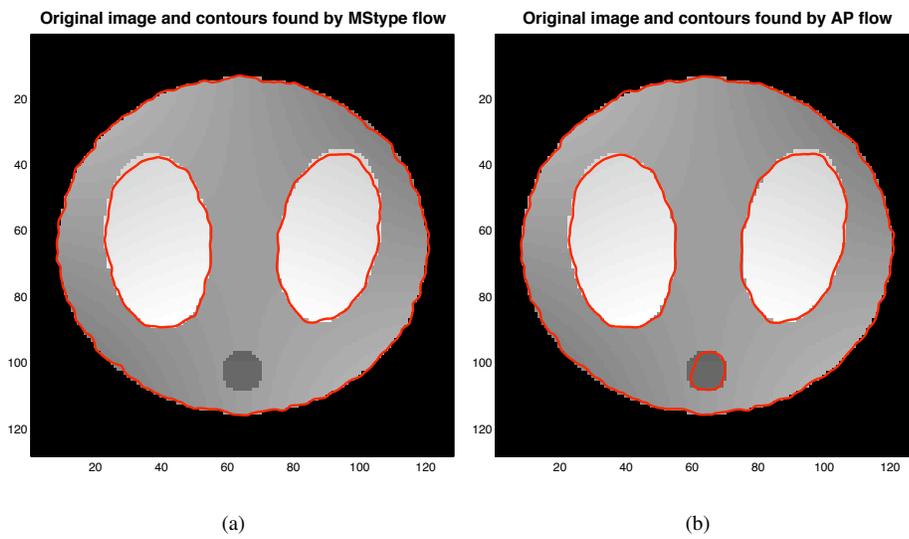


Figure 2.19: Contours found by the Mumford-Shah based flow and by the area preserving flow. Figure 2.19(a) shows the contours obtained by the Mumford-Shah based model and Figure 2.19(b) displays the contours obtained by the area preserving flow.

2.5 Conclusion

We described efficient and highly accurate algorithms for simulating general area preserving flows in the plane, with normal velocities of the form $v_N = \kappa - \bar{\kappa} + S$. Our schemes are based on a diffusion-generated motion approach using signed distance functions, thus making them more accurate than the standard threshold dynamics based schemes. Additionally, we proposed both first-order and second-order in time version of our algorithms and carried out tests to check their numerical convergence and accuracy. We presented two different applications of our new schemes. The first application demonstrated the ability of our algorithms to handle large scale computations due to its high accuracy and computational efficiency. In particular, our schemes were able to simulate the coarsening of large configurations of droplets under area preserving curvature motion. In addition, the results of our computations allowed us to perform some interesting statistical measurements on the configurations of droplets. The second application investigated the effect of perimeter regularization used in inverse problems on reconstructed images. We showed that the area preserving flow was able to detect small round objects with a lower fidelity constant than a Mumford-Shah based flow (i.e., using perimeter regularization), thus finding small round objects for a wider range of fidelity constants than the Mumford-Shah based model. This indicates that area preserving flows might be less sensitive to parameters than Mumford-Shah based flows.

The next chapter of this dissertation is the analytical part, and is devoted to a rigorous analysis of the coarsening rate of some high-order PDE models used in image processing applications.

CHAPTER III

Coarsening in High Order, Discrete, Ill-Posed Diffusion Equations

3.1 Introduction

As discussed in the introduction, the well-known model of Perona-Malik [76, 77] constitutes the first nonlinear equation proposed for image processing and computer vision applications. Their idea was to create an improved scale-space analysis that would gradually smooth the image away from the edges. We recall their model (1.7)

$$\begin{cases} u_t = \nabla \cdot (g(|\nabla u|) \nabla u) \\ u(x, 0) = f(x), \end{cases}$$

where g is a bounded, non-increasing positive function such that $\lim_{x \rightarrow \infty} g(x) = 0$ and $g(0) = 1$ allowing the spatial diffusion to become spatially adaptive. It turns out that such choice for g lead to an ill-posed model, which at the location of large gradients (i.e., edges) acts as a backward heat equation. Additionally, one significant practical caveat of this model is the *staircasing* artifact which produces “blocky” images, namely regions of moderately large gradient in the image develop flat regions separated by spurious edges. In fact, the staircasing instability of the Perona-Malik model is best explained in the one dimensional case where the solution u develops steps which merge together to create larger steps as the solution evolves in time. The result is an image that looks “blocky”. The illustration of the Perona-Malik evolution on an image was displayed in Chapter I, Figure 1.1.

You and Kaveh [100] introduced a nonlinear, fourth-order PDE as an improvement of the Perona-Malik model, particularly with regard to the staircasing artifact. In their model, the image gradually turns into a simpler and simpler piecewise *linear* image (as opposed to piecewise *constant*), which greatly abates the staircasing effect. Their fourth order PDE is

$$(3.1) \quad u_t = -\Delta (R(\Delta u)) \text{ subject to } u(x, 0) = f(x),$$

where $R(x) = xg(|x|)$ and g is the same nonlinear diffusivity as utilized in the Perona-Malik model. Like the Perona-Malik evolution, (3.1) generates coarser resolution images from a given one taken as initial condition, while preserving sharp contours and smoothing everything else. Numerical simulations with this PDE give similar results to Perona and Malik's in terms of noise removal and edge preservation but the staircasing artifact is almost completely absent. Figure 3.1 shows the evolution of the boat image according to the You and Kaveh equation.

However, being the fourth order analogue of the Perona-Malik model, the You-Kaveh model inherits the same ill-posedness issues as its progenitor. Some analysis of this PDE has been carried out by Greer and Bertozzi in [52], where they show that smooth traveling wave solutions of the one dimensional You-Kaveh PDE do not exist for sufficiently large jump heights. Their conclusion conjectures that the You-Kaveh equation has finite time singularities in u_{xx} , just as the Perona-Malik equation has finite time singularities in the slope u_x [51, 56].

Motivated by You and Kaveh's fourth order analogue of the Perona-Malik model, we consider the following family of $2n$ -th order nonlinear diffusion equations ($n \in \mathbb{N}^*$):

$$(3.2) \quad u_t = \begin{cases} -\Delta^{\frac{n}{2}}(R(\Delta^{\frac{n}{2}}u(x))) & \text{if } n \text{ is even,} \\ \nabla \cdot \Delta^{n-1}R(\nabla \Delta^{\frac{n-1}{2}}u(x)) & \text{if } n \text{ is odd.} \end{cases}$$

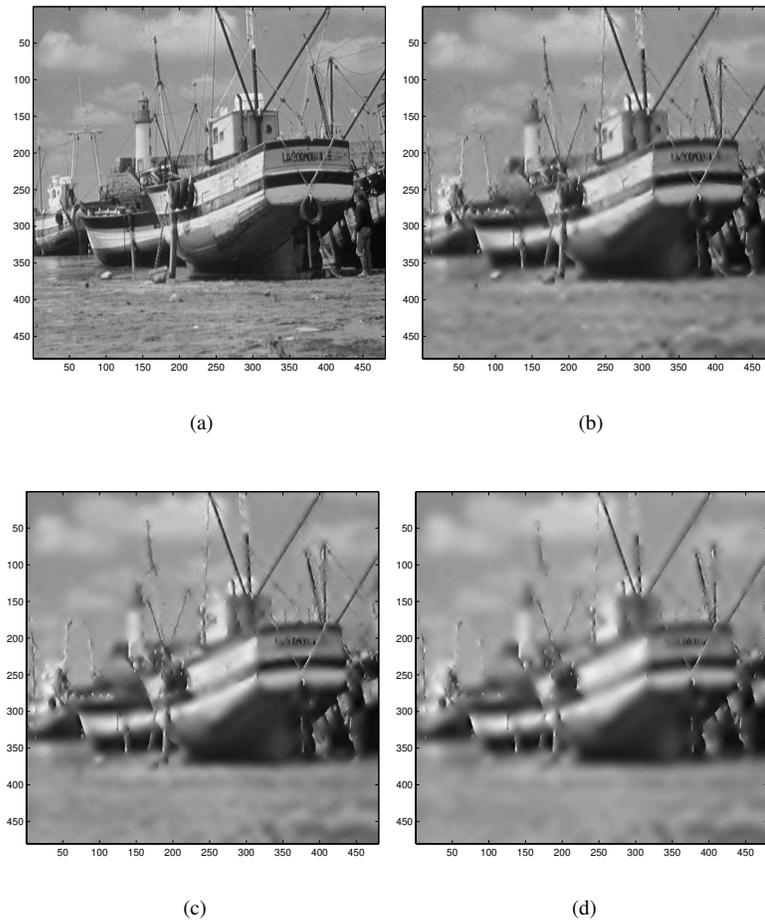


Figure 3.1: Simplification (or coarsening) of an image according the You and Kaveh equation. Figure 3.1(a) shows the original image, and Figures 3.1(b), 3.1(c) and 3.1(d) display the image at later times in the evolution.

For $n = 1$ equation (3.2) becomes the well-known Perona-Malik model, and for $n = 2$ we recover the You-Kaveh model. We prove rigorous upper bounds on the coarsening rate of (3.2) in any dimension $d \in \mathbb{N}^*$, for a large class of diffusivities R that include the original choices of Perona-Malik and You-Kaveh.

Upper bounds on the coarsening rate of second order, discrete, ill-posed diffusion equations (based on the Perona-Malik model) were obtained in [34] and improved in [37] for a large class of diffusivities. In this thesis we consider the same class of diffusivities as in [37] and prove analogous bounds for the $2n$ -th order equation (3.2) in any space dimension. However, our results and arguments differ from [34, 37] in several ways. First

of all, in the present thesis we work in the more general $2n$ -th order setting. Secondly, the rigorous bounds in [34, 37] are stated in terms of the decrease rate of the *energy* of the system (a somewhat abstract quantity), whereas the real quantity of interest in applications is the edge density in the solutions (equivalently, the number of “spikes” in its derivative). Indeed, the numerical experiments – unlike the rigorous results – presented in these papers are in terms of the spike density in the derivative of the solution, and even though the spike density and the energy can be related to each other through a heuristic argument, it is desirable to have a completely rigorous statement directly in terms of the spike density of the solution. In the present thesis, we replace the heuristic step by a rigorous argument and therefore obtain rigorous bounds for the spike density. Third, unlike in the second order case, the high order models we consider lack a maximum principle, which requires certain arguments to be modified. In particular, the construction of the test function needed to estimate the bounds in the low dimensional case is different from the construction used in [34, 37]. Our construction is simple and more general than the one proposed in these previous papers. Finally, it should be mentioned that the discrete PDEs considered in [34, 37] are only related to the Perona-Malik scheme in one space dimension; in higher dimensions, the correspondence between the image processing model and the PDEs considered in the afore-mentioned papers is lost. In this thesis, the fourth order PDEs considered and the theory developed for them corresponds to the relevant image processing model – the You-Kaveh model – in all space dimensions. For the general $2n$ -th order PDEs, the correspondence between the theory and the model exists in all space dimensions whenever n is even.

As in the second order case, the high order discrete-in-space equations exhibit an instability that very rapidly leads to the formation of spikes or discontinuities of width exactly one grid point, starting from generic initial data. These singular structures then interact

without changing their location: small ones get absorbed by larger ones, leading to fewer and larger structures and a corresponding increase in the length scale of the solution as measured by e.g. the average distance between the structures. We call the rate of increase of this quantity the coarsening rate.

In computer vision and image processing applications, the coarsening of the You-Kaveh flow can be observed as the noisy image gradually simplifies into a piecewise linear approximation by merging neighboring regions and preserving their edges. The rate at which this coarsening takes place is therefore crucial since it could provide a way to automatically stop the evolution at a given level of simplification in the image. Moreover, such an analysis would provide insight into the effect of different diffusivities on the dynamics. Our method is based on a recent technique developed by Kohn and Otto [58] for energy driven systems. This method uses the energy E of the system and a length scale quantity L in two inequalities: a dissipation inequality that involves $\frac{dL}{dt}$ and $\frac{dE}{dt}$, and an interpolation inequality between L and E . Combined with an ODE argument, these inequalities lead to a lower bound on the time-average of the energy which is equivalent to an upper bound on the coarsening rate. We present our results in any space dimension. For simplicity and clarity purposes, we present a detailed study of the fourth order case (corresponding to the You-Kaveh model from image processing), and then indicate how the results extend to higher order analogues.

3.2 Analysis of the You-Kaveh model

3.2.1 The equations

We consider the fourth order ill-posed nonlinear PDE proposed by You and Kaveh [100]

$$(3.3) \quad u_t = -\Delta(g(\Delta u)\Delta u),$$

and let $R(s) = sg(s)$, so that equation (3.3) becomes

$$(3.4) \quad u_t = -\Delta(R(\Delta u)).$$

If we let $v = \Delta u$ and take the Laplacian of equation (3.4), we obtain the following equation for v :

$$(3.5) \quad v_t = -\Delta^2(R(v)).$$

In the rest of this section, we will study the discrete version of equation (3.5) on $[0, 1]^d$ for a large class of diffusivities R , where $d \in \mathbb{N}^*$ is the spatial dimension.

3.2.2 The scheme

We work on a uniform discretization of the domain $[0, 1]^d$ and let $\delta = \frac{1}{N}$ denote the grid size in each of the coordinate directions. Consider the unit-spaced lattice $\mathbb{L} = \{0, 1, \dots, N-1\}^d$ and let $i \in \mathbb{L}$ be the vector $(i_1, \dots, i_j, \dots, i_d) \in \mathbb{L}$ corresponding to the point $(\frac{i_1}{N}, \dots, \frac{i_j}{N}, \dots, \frac{i_d}{N})$ in $[0, 1]^d$. Define the discrete Laplacian operator Δ_δ operating on the grid function v at the point $i \in \mathbb{L}$ to be

$$\Delta_\delta v_i := \sum_{j=1}^d D_{j,\delta}^+ D_{j,\delta}^- v_i,$$

where

$$D_{j,\delta}^+ v_i := \frac{v_{i+e_j} - v_i}{\delta} = \frac{v_{i_1, \dots, i_j+1, \dots, i_d} - v_{i_1, \dots, i_j, \dots, i_d}}{\delta},$$

and

$$D_{j,\delta}^- v_i := \frac{v_i - v_{i-e_j}}{\delta} = \frac{v_{i_1, \dots, i_j, \dots, i_d} - v_{i_1, \dots, i_j-1, \dots, i_d}}{\delta},$$

are forward and backward difference quotients respectively, in the j th coordinate direction, for $1 \leq j \leq d$. The ‘‘fourth order’’ discrete system that we consider is the following natural

spatial finite difference discretization of (3.5)

$$(3.6) \quad \frac{dv_i}{dt} = -\Delta_\delta^2(R(v_i)), \quad i \in \mathbb{L}.$$

This discrete in space, continuum in time version of (3.5) is equivalent to the discretization of the You-Kaveh model used in practice in image processing applications.

We impose periodic boundary conditions on (3.6), which means that we identify the lattice \mathbb{L} with \mathbb{Z}^d on which for any integer $0 \leq i_j \leq N - 1$ and $k_j \in \mathbb{Z}$, $0 \leq j \leq d$, the point $(i_1 + k_1N, \dots, i_d + k_dN)$ is identified with (i_1, \dots, i_d) . This periodicity ensures that the dynamics of (3.6) leave the total mass

$$(3.7) \quad \mu := \frac{1}{N^d} \sum_{i \in \mathbb{L}} v_i$$

unchanged. This system also has a non-increasing energy

$$(3.8) \quad E(v) = \frac{1}{N^d} \sum_{i \in \mathbb{L}} f(v_i),$$

where the density function f is defined as

$$(3.9) \quad f(x) = \int_0^x R(\xi) d\xi,$$

for $x \in \mathbb{R}$. For smooth and strictly convex energy densities f , solutions generated by the scheme (3.6) would be expected to converge (as $h \rightarrow 0^+$) to the solution of the parabolic PDE (3.5) on $[0, 1]^d$, with periodic boundary conditions. In this thesis, however, we consider *non-convex* even energy densities f that satisfy $f(x) > \eta|x|^\alpha$ for some constant $\eta > 0$, $\alpha \in [0, 1[$, and all $|x|$ large enough. Moreover, for $x \geq 0$, the functions f will be assumed to be strictly convex near 0, concave for large x , and have only one inflection point $b > 0$. By symmetry, $\pm b$ are the two inflection points of f on \mathbb{R} . Since (3.5) can be expanded as

$$v_t = -\Delta^2(R(v)) = -R'(v)\Delta^2 v + \underbrace{F(v, \nabla v, \nabla^2 v, \nabla^3 v)}_{\text{Lower Order Terms}},$$

we see that equation (3.5) becomes backward parabolic whenever $R'(v) < 0$, which occurs whenever $|v|$ is large enough. Consequently, there is no complete well-posedness theory for equation (3.5). Furthermore, we make the following assumptions on R :

1. $R : \mathbb{R} \rightarrow \mathbb{R}$ is smooth,
2. $R'(x) > 0$ on $[-b, b[$,
3. $R'(x) < 0$ on $] -\infty, -b[\cup]b, \infty[$, and
4. $R(0) \leq \lim_{x \rightarrow \pm\infty} R(x)$.

In image processing applications, a common choice for the diffusivity R is

$$(3.10) \quad R(x) = \frac{x}{1 + \left(\frac{x}{b}\right)^2},$$

which corresponds to the choice of g given in (1.8b). The corresponding density function f of (3.10) is the logarithmic density function

$$(3.11) \quad f(x) = \frac{b}{2} \ln \left(1 + \left(\frac{x}{b}\right)^2 \right), \quad b > 0.$$

However, other choices of R leading to backwards parabolic behavior are regularly used in applications. The practical implications of the choice of the diffusivity R on results has often been raised in the engineering literature (see e.g. [78]) and is still a current topic of research. As an illustration, Figure 3.2 shows the graph of f , R and R' , where the density function f is as in (3.11).

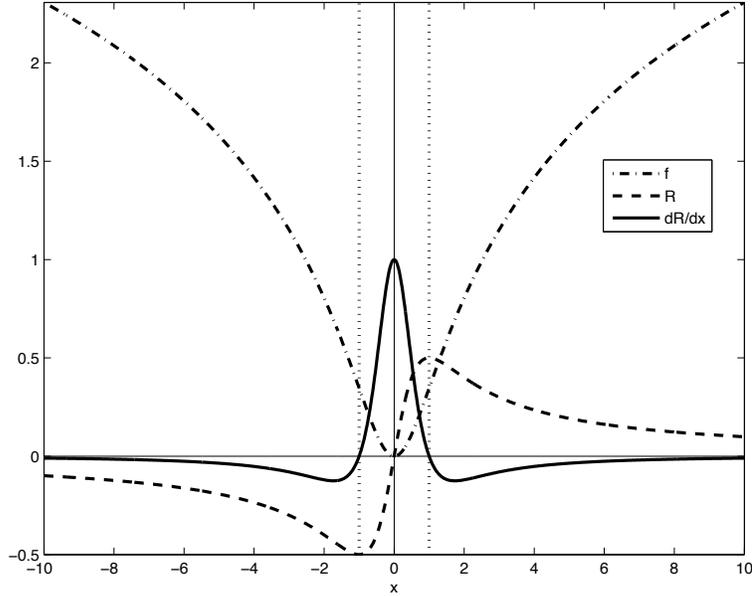


Figure 3.2: Functions f , R and R' with $R(x) = \frac{x}{1+x^2}$ and $f(x) = \frac{1}{2} \ln(1+x^2)$. The changes in concavity of the function f , which occur at ± 1 , are shown by the two dotted vertical lines.

3.2.3 Terminology and basic setup

The system (3.6) is gradient descent for the energy E defined in (3.8) with respect to the discrete H^{-2} norm. This norm can be expressed by duality as

$$(3.12) \quad \|v\|_{H^{-2}} := \left\{ \sup_{\phi} \frac{1}{N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \phi_i : \frac{1}{N^d} \sum_{i \in \mathbb{L}} (\Delta_{\delta}(\phi_i))^2 \leq 1 \right\}.$$

If the initial data of (3.6) satisfies $v_i \in]-b, b[$ so that it lies completely in the forward parabolic regime, the evolution proceeds as a typical parabolic smoothing, at least for small time. If on the other hand the initial data's mean value μ satisfies

$$|\mu| > b,$$

then part of the mass of the data sits in the ill-posed regime $] -\infty, -b[\cup]b, \infty[$ for all time, due to the conservation of mass. In this case, instabilities in the form of *spikes* rapidly develop from the initial data and start to interact, gradually decreasing in number

but growing in size. These spikes are located at the grid points where the solution exceeds the parabolicity threshold b . We therefore use the following terminology:

There is a *spike* at the j -th grid point if $|v_j| \geq b$.

Each spike is supported by one grid-point and remains at that grid location during the entire evolution of the equation. However, while the location of each spike is fixed, their height varies. Each spike either grows, or shrinks, and may eventually disappear as its mass is absorbed by others. Indeed, the concavity of f on $] - \infty, -b[\cup]b, \infty[$ encourages the accumulation of mass into fewer and higher spikes during the evolution. Smaller spikes thus get absorbed into larger ones until, generically, only one spike containing most of the mass remains; this is what we mean by coarsening of the system in this thesis. An example of this coarsening is shown in Figure 3.3. During the evolution, the average distance between spikes is observed to increase, and constitutes a natural length scale for the system.

The length scale L should be inversely proportional to the number of spikes K and therefore should behave something like

$$(3.13) \quad L \sim \frac{1}{K^\beta},$$

where $\beta > 0$ may depend on the spatial dimension. In addition, since f is an increasing function of $|v|$ for $v \in \mathbb{R}$, most of the energy is initially contained in the spikes, which suggests

$$(3.14) \quad E \sim \frac{K^\gamma}{N^d},$$

where $\gamma > 0$. Both γ and β are parameters to be identified. The interpolation inequality that we need relates the energy E and the length scale L in a one-sided version of

$$EL^{\frac{\gamma}{\beta}} \sim C(N),$$

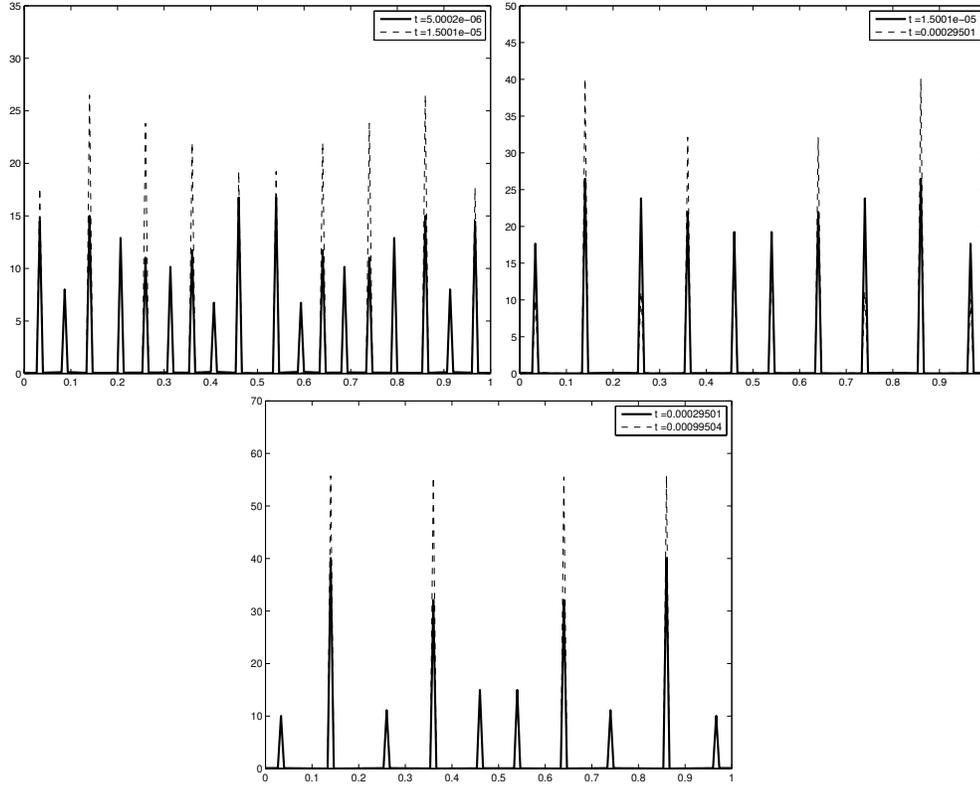


Figure 3.3: One dimensional evolution of (3.6) for $N = 150$ and initial solution $v_i(0) = \frac{3}{2} + 10^{-3} \sin(\pi x)$ with $x \in [0, 1]$. The solution quickly forms spikes whose number decreases as some of them shrink and disappear, while others grow taller to preserve the total mass.

where $C(N)$ is a constant depending only on N . So if we had a length scale L that behaved as in equation (3.13), we could then use the relation (3.14) to immediately have the interpolation inequality

$$EL^{\frac{\gamma}{\beta}} \sim \frac{1}{N^d}.$$

A natural choice for the length scale quantity L is therefore $L = \frac{1}{K^\beta}$, but the decay relation would then be hard to prove since it relates $\frac{dL}{dt}$ and $\frac{dE}{dt}$, and thus would involve differentiating $L = \frac{1}{K^\beta}$ which is a discrete function of time. For these reasons, we choose

$$(3.15) \quad L = \|v\|_{H^{-2}},$$

which is motivated by the fact that the system (3.6) is gradient descent for the energy E defined in (3.8) with respect to the discrete H^{-2} norm. This choice is also similar to length

scales used in [34, 58, 59, 74]. In particular, L as given in (3.15) will imply the following scaling between the length scale L and the number of spikes K

$$L \sim \begin{cases} \frac{1}{K^{\frac{1}{2}}} & \text{for } 1 \leq d \leq 3, \\ \frac{1}{\sqrt{K}} & \text{for } d \geq 4. \end{cases}$$

In the next section, we will study the stability of all the steady state solutions of the one dimensional system in the case of the logarithmic density function (3.11). The results show that solutions with more than two spikes, together with the constant solution equal to b or $-b$ (depending on the sign of μ), are unstable. The single spike solution (obtained when $|\mu| > b$) and the constant solution with values in $] -b, b[$ (obtained when $|\mu| < b$) are both stable. We note that these stability results can easily be generalized to the more general density functions f that are considered in this thesis.

3.2.4 Stability of stationary states

In the one dimensional case, the scheme becomes

$$(3.16) \quad \frac{dv_i}{dt} = -(D_\delta^+ D_\delta^-)^2 (R(v_i)), \quad 0 \leq i \leq N-1,$$

where

$$D_h^+ v_i = \frac{v_{i+1} - v_i}{\delta},$$

and

$$D_h^- v_i = \frac{v_i - v_{i-1}}{\delta},$$

are forward and backward difference quotients.

Steady states

To find the steady states of (3.16) we need to solve

$$\frac{dv_i}{dt} = 0, \quad \forall i \in [0, \dots, N-1],$$

i.e.,

$$(D_\delta^+ D_\delta^-)^2 (R(v_i)) = 0, \forall i \in [0, \dots, N-1].$$

Using the periodic boundary conditions we obtain

$$(3.17) \quad R(v_i) = C, \forall i \in [0, \dots, N-1],$$

where $C \in \mathbb{R}$ is a constant. Since the range of $s \mapsto R(s) = \frac{s}{1+(\frac{s}{b})^2}$ on \mathbb{R} is $[-\frac{b}{2}, \frac{b}{2}]$ we see that $-\frac{b}{2} \leq C \leq \frac{b}{2}$. We now consider three different cases:

1. $C = 0$

Then $v_i = 0 \forall i \in [0, \dots, N-1]$, which violates the conservation of mass (3.7) for $\mu \neq 0$. Thus if $\mu \neq 0$, $v = 0$ is not a feasible solution.

If $\mu = 0$ however, the solution $v = 0$ is asymptotically stable.

2. $C = \frac{b}{2}$ or $C = -\frac{b}{2}$

(a) If $C = \frac{b}{2}$, then $v_i = b, \forall i \in [0, \dots, N-1]$.

(b) If $C = -\frac{b}{2}$, then $v_i = -b, \forall i \in [0, \dots, N-1]$.

3. $0 < C < \frac{b}{2}$ or $-\frac{b}{2} < C < 0$.

(a) If $0 < C < \frac{b}{2}$, then (3.17) has two solutions that we call v_+ and v_- , with

$$0 < v_- < b < v_+ < \infty.$$

Also the condition $R(v_-) = R(v_+)$ gives

$$(3.18) \quad v_+ = \frac{b^2}{v_-}.$$

Now we define K ($1 \leq K \leq N-1$) to be the number of grid points that have the value v_+ . Conservation of mass (3.7) then implies

$$(3.19) \quad K v_+ + (N-K) v_- = N \mu.$$

We note that for $K = 0$, the steady state solution is just the constant solution with value $v_- = \mu$, which happens when $0 < \mu < b$. Similarly, when $K = N$, the steady state is the constant solution with value $v_+ = \mu$, which occurs when $\mu > b$. For $1 \leq K \leq N - 1$, we use equations (3.18) and (3.19) to solve for v_- and v_+ and obtain the following result:

Under the condition $\mu \geq \mu_{K,N}$ with $\mu_{K,N} = \frac{2b\sqrt{K(N-K)}}{N}$, we have the existence of two real solutions v_+ and v_- with value

$$(3.20) \quad \begin{cases} v_+ = \frac{N\mu + \sqrt{(N\mu)^2 - 4K(N-K)b^2}}{2K}, \\ v_- = \frac{N\mu - \sqrt{(N\mu)^2 - 4K(N-K)b^2}}{2(N-K)}. \end{cases}$$

(b) If $-\frac{b}{2} < C < 0$, (3.17) has two solutions that we call v_+ and v_- , with $-\infty < v_+ < -b < v_- < 0$. Using similar calculations we obtain the following:

For $1 \leq K \leq N - 1$, and under the condition $\mu \leq -\mu_{K,N}$, we have the existence of two real solutions v_+ and v_- with value

$$(3.21) \quad \begin{cases} v_+ = \frac{N\mu - \sqrt{(N\mu)^2 - 4K(N-K)b^2}}{2K}, \\ v_- = \frac{N\mu + \sqrt{(N\mu)^2 - 4K(N-K)b^2}}{2(N-K)}. \end{cases}$$

For $K = 0$, the steady state solution is the constant solution with value $v_- = \mu$, which happens in the case $-b < \mu < 0$, and for $K = N$, the steady state is the constant solution with value $v_+ = \mu$, which occurs when $\mu < -b$.

So the steady states of equation (3.16) are vectors v with K grid points with value v_+ and $(N - K)$ grid points with value v_- , for $1 \leq K \leq N - 1$. The condition on μ ensures

the existence of v_+ and v_- as real solutions. If $|\mu| < \mu_{K,N}$ for $1 \leq K \leq N-1$, there are no real solutions v_+ and v_- . The steady state solution with $K=0$ is the constant solution $v = \mu$, which occurs when $|\mu| < b$. Similarly, the steady state solution with $K=N$ is the constant solution $v = \mu$, in the case $|\mu| > b$. Note that $\mu_{K,N} \leq b$, for all $0 \leq K \leq N$. $\mu_{K,N}$ is only needed for $1 \leq K \leq N-1$, but it is also defined for $K=0$ and $K=N$. In particular, $\mu_{0,N} = \mu_{N,N} = 0$.

Lemma III.1. *Suppose $|\mu| > \mu_{1,N}$ and let v_- be as in (3.20) or (3.21) for $K=1$, namely*

$$v_- = \frac{N\mu - \sqrt{(N\mu)^2 - 4(N-1)b^2}}{2(N-1)} \quad \text{or} \quad v_- = \frac{N\mu + \sqrt{(N\mu)^2 - 4(N-1)b^2}}{2(N-1)}.$$

Then

$$(N-1)v_-^2 < b^2.$$

Proof: For either v_- above, we obtain after simplification

$$(N-1)v_-^2 = \frac{2}{\frac{4(N-1)}{(N\mu)^2}} \left(1 - \sqrt{1 - \frac{4(N-1)b^2}{(N\mu)^2}} \right) - b^2,$$

with $0 < \frac{4(N-1)}{(N\mu)^2} < \frac{1}{b^2}$. Now we consider the function $k(x) = \frac{1}{x} \left(1 - \sqrt{1 - b^2 x} \right)$ where $x \in]0, \frac{1}{b^2}[$. Since the range of the function k is $] \frac{b^2}{2}, b^2[$, $\forall x \in]0, \frac{1}{b^2}[$, we obtain

$$2k\left(\frac{4(N-1)}{(N\mu)^2}\right) - b^2 = (N-1)v_-^2 < b^2,$$

which proves the lemma. \square

We note that if $K=1$ and $|\mu| = \mu_{1,N}$ then $(N-1)v_-^2 = b^2$, leading to

$$(3.22) \quad v_+ = b\sqrt{N-1} \quad \text{and} \quad v_- = \frac{b}{\sqrt{N-1}}.$$

Stability analysis

In this section, we study the stability of the steady states using energy based arguments that easily generalize to higher dimensions. In [98], Witelski *et al.* studied the stability

of the steady states of a similar system using linearization. Their analysis was done in the one dimensional case only.

Proposition III.2. *Equilibrium solutions with $K \geq 2$ are unstable.*

For the proof, we refer to the one of Proposition 4 in Witelski *et al.* [98].

Proposition III.3. *For $|\mu| < b$, the equilibrium solution with $K = 0$ (i.e., $v_i = v_- = \mu$, $\forall i \in [0, \dots, N - 1]$) is stable.*

Proof: We prove this result by considering the energy functional (3.8)

$$E(v) = \frac{1}{N} \sum_{i=0}^{N-1} f(v_i),$$

where f is as in (3.9), and checking whether any perturbation of the constant steady state v_- can result in a decrease in energy. We thus consider the minimization problem

$$(3.23) \quad \min_v E(v) \text{ such that } \sum_{i=0}^{N-1} (v_i - v_-)^2 \leq \epsilon \text{ and } \frac{1}{N} \sum_{i=0}^{N-1} v_i = v_- = \mu,$$

for $\epsilon > 0$. Suppose the minimum of the energy is reached at a certain v^m which is not constant, i.e., $\exists i_1$ and $i_2 \in [0, \dots, N - 1]$ such that $v_{i_1}^m$ is the smallest value, $v_{i_2}^m$ the largest value, and $v_{i_1}^m \neq v_{i_2}^m$.

Now define \tilde{v}^m as

$$\tilde{v}_i^m = \begin{cases} v_i^m & \forall i \in [0, \dots, N - 1] \setminus \{i_1, i_2\} \\ v_{i_2}^m & \text{if } i = i_1 \\ v_{i_1}^m & \text{if } i = i_2. \end{cases}$$

Trivially $E(v^m) = E(\tilde{v}^m)$. Now we choose ϵ small enough so that none of the values of v^m reside outside the interval $] - b, b[$, namely

$$|v_i^m| < b, \forall i \in [0, \dots, N - 1].$$

Since f is strictly convex on $] - b, b[$, we have

$$f\left(\frac{v_i^m + \tilde{v}_i^m}{2}\right) = \frac{1}{2} (f(v_i^m) + f(\tilde{v}_i^m)), \quad \forall i \in [0, \dots, N-1] \setminus \{i_1, i_2\},$$

$$f\left(\frac{v_{i_1}^m + \tilde{v}_{i_1}^m}{2}\right) = f\left(\frac{v_{i_1}^m + v_{i_2}^m}{2}\right) < \frac{1}{2} (f(v_{i_1}^m) + f(v_{i_2}^m)),$$

and similarly for $i = i_2$. We therefore obtain

$$E\left(\frac{v^m + \tilde{v}^m}{2}\right) < \frac{1}{2} (E(v^m) + E(\tilde{v}^m)) = E(v^m),$$

contradicting the assumption that v^m achieves the minimum of the energy. The constant solution is therefore the only one that minimizes the energy and thus is asymptotically stable. \square

Proposition III.4. *Assume $N \geq 3$. For $\mu = b$, the constant equilibrium solution $v \equiv b$ is unstable. Similarly, for $\mu = -b$ the constant equilibrium solution $v \equiv -b$ is unstable.*

Proof: We prove the lemma for the constant solution $v \equiv b$. The proof for $v \equiv -b$ is analogous. We choose an integer $M \in [2, N-1]$ and perturb the equilibrium solution in the following way: we increase one grid point from b to $b + \epsilon$ and decrease M other grid points from b to $b - \epsilon/M$ where $\epsilon > 0$ will be determined later. The remaining grid points have value b . We define v^0 to be the constant solution equal to b and by v^ϵ the perturbed solution. We note that $\|v^\epsilon - v^0\|_2 = \sqrt{\frac{M+1}{NM}}\epsilon$.

Now we will show that v^ϵ decreases the energy, i.e.,

$$(3.24) \quad \forall M \in [2, N-1], \exists \epsilon > 0 \text{ such that } E(v^\epsilon) < E(v^0).$$

Calculating the energy for both v^0 and v^ϵ we obtain

$$E(v^0) = f(b),$$

and

$$E(v^\epsilon) = \frac{1}{N} \left((N - M - 1)f(b) + Mf\left(b - \frac{\epsilon}{M}\right) + f(b + \epsilon) \right).$$

Subtracting $E(v^0)$ from $E(v^\epsilon)$, we obtain

$$E(v^\epsilon) - E(v^0) = \frac{1}{N} \left(\left(Mf\left(b - \frac{\epsilon}{M}\right) + f(b + \epsilon) \right) - (M + 1)f(b) \right) = \frac{\zeta(\epsilon)}{N},$$

where $\zeta(\epsilon) = \left(Mf\left(b - \frac{\epsilon}{M}\right) + f(b + \epsilon) \right) - (M + 1)f(b)$.

We now show that $\exists \epsilon > 0$ such that $\zeta(\epsilon) < 0$. Replacing f by its expression we obtain

$$\zeta(\epsilon) = \frac{b}{2} \ln \left(\left(1 - \frac{\epsilon}{bM} + \frac{1}{2} \left(\frac{\epsilon}{bM} \right)^2 \right)^M \left(1 + \frac{\epsilon}{b} + \frac{1}{2} \left(\frac{\epsilon}{b} \right)^2 \right) \right) = \frac{b}{2} \ln \left(\varsigma \left(\frac{\epsilon}{b} \right) \right),$$

where $\varsigma(x) = \left(1 - \frac{x}{M} + \frac{1}{2} \left(\frac{x}{M} \right)^2 \right)^M \left(1 + x + \frac{x^2}{2} \right)$ for $x > 0$. ς satisfies the following properties on \mathbb{R}^+ :

$$\varsigma(x) < 1 \text{ for } x \in]0, M - 1[.$$

This therefore implies that

$$\exists \epsilon \in]0, b(M - 1)[, \text{ such that } \varsigma \left(\frac{\epsilon}{b} \right) < 1,$$

and thus shows (3.24). \square

Proposition III.5. *If $|\mu| > \mu_{1,N}$, then the equilibrium solution with $K = 1$ is asymptotically stable. If $|\mu| = \mu_{1,K}$, then the equilibrium solution with $K = 1$ is unstable.*

Proof: We consider μ positive but the proof for μ negative is completely similar.

Let $\mu \geq \mu_{1,N}$. We define v^0 to be the equilibrium solution and i_0 the grid point at which v^0 takes the value v_+ . Recall that since v^0 is the equilibrium solution for $K = 1$ and $\mu \geq \mu_{1,N}$, $v_i^0 = v_-$ for all $i \neq i_0$. Consider the minimization problem

$$\min_v E(v) \text{ such that } \sum_{i=0}^{N-1} (v_i - v_i^0)^2 \leq \epsilon \text{ and } \sum_{i=0}^{N-1} v_i = N\mu,$$

with $\epsilon > 0$.

We first consider all perturbations that only affect the $(N - 1)$ grid points with value v_- . In this case, we can use the analysis done for $K = 0$ since the mass of the $(N - 1)$ perturbed grid points is preserved. We therefore know that the minimum of the energy is reached when the $(N - 1)$ grid points have the same constant value v_- .

Now we consider all the perturbations that also perturb the grid point at i_0 . We define $\frac{\delta_m}{N} \neq 0$ to be the change of mass of the $(N - 1)$ remaining grid points. The value of the grid point at i_0 is now $v_+ + \delta_m$. Since the value of this grid point is determined for each δ_m , we can look at all the perturbations that change the mass of the $(N - 1)$ remaining grid points by $\frac{\delta_m}{N}$. Here again, we can apply the analysis done for $K = 0$ with the condition on the mass in equation (3.23) replaced by

$$\frac{1}{N} \sum_{i \neq i_0} v_i = \frac{N - 1}{N} v_- - \frac{\delta_m}{N},$$

since we choose ϵ small enough so that the perturbed $(N - 1)$ grid points stay strictly between $-b$ and b . We can therefore conclude that the energy is minimized if the $(N - 1)$ remaining grid points have a constant value $v_- - \frac{\delta_m}{N-1}$. Consequently the solution that minimizes the energy with $v_{i_0} = v_+ + \delta_m$ is

$$v_i^p = \begin{cases} v_+ + \delta_m & \text{if } i = i_0 \\ v_- - \frac{\delta_m}{N-1} & \text{if } i \neq i_0. \end{cases}$$

Now we consider the energy associated to the solution v^p

$$E(v^p) = \frac{1}{N} \left(f(v_+ + \delta_m) + (N - 1) f\left(v_- - \frac{\delta_m}{N-1}\right) \right).$$

Differentiating with respect to δ_m and evaluating at $\delta_m = 0$ gives

$$\left. \frac{dE(v^p)}{d\delta_m} \right|_{\delta_m=0} = \frac{1}{N} (R(v_+) - R(v_-)) = 0,$$

since $R(v_-) = R(v_+)$. Differentiating again with respect to δ_m and evaluating at $\delta_m = 0$ we obtain

$$\left. \frac{d^2E(v^p)}{d\delta_m^2} \right|_{\delta_m=0} = \frac{1}{N} \left(\frac{1}{N-1} - \frac{v_-^2}{b^2} \right) R'(v_-).$$

- Now suppose $\mu > \mu_{1,N}$.

Using Lemma III.1 we know that $\frac{1}{N-1} - \frac{v_-^2}{b^2} > 0$, which shows that

$$\left. \frac{d^2E(v^p)}{d\delta_m^2} \right|_{\delta_m=0} > 0.$$

Thus the solution v^0 is asymptotically stable.

- Now suppose $\mu = \mu_{1,N}$.

In this case, as noted at the end of Lemma III.1, we have

$$(N-1)v_-^2 = b^2,$$

which leads to

$$\left. \frac{d^2E(v^p)}{d\delta_m^2} \right|_{\delta_m=0} = \frac{1}{N} \left(\frac{1}{N-1} - \frac{v_-^2}{b^2} \right) R'(v_-) = 0.$$

Continuing to differentiate, we look at the third derivative of the energy with respect to δ_m and obtain

$$\left. \frac{d^3E(v^p)}{d\delta_m^3} \right|_{\delta_m=0} = \frac{1}{N} \left(R''(v_+) - \frac{1}{(N-1)^2} R''(v_-) \right).$$

Using the expressions for v_+ and v_- in (3.22) we obtain

$$\left. \frac{d^3E(v^p)}{d\delta_m^3} \right|_{\delta_m=0} = \frac{2\sqrt{N-1}N-2}{bN^3} \frac{N-2}{N-1} = \frac{2(N-2)}{bN^3\sqrt{N-1}},$$

which is strictly positive for $N \geq 3$. The steady state v^0 is therefore a saddle point in the energy landscape and thus is unstable. \square

Remark. The stability analysis carried out for the one dimensional system can easily be generalized to the d dimensional system, for any $d \in \mathbb{N}^*$, since it is only based on energy arguments which do not depend on the dimension.

3.3 Upper bounds on the coarsening rate of You-Kaveh type models

Following [37], we introduce the model density functions

$$(3.25) \quad F_\alpha(x) := \begin{cases} 0 & \text{if } 0 \leq |x| \leq b \\ |x - b|^\alpha & \text{if } |x| > b, \end{cases}$$

where $b > 0$ and $\alpha \in [0, 1[$. Our results are obtained for general density functions $f \geq \eta F_\alpha$ for some constant $\eta > 0$. Without loss of generality, we work with positive initial data, i.e., $v_i(0) \geq 0$, for all $i \in \mathbb{L}$.

For the statement of Theorem III.6, we introduce the following functions:

$$\sigma = \sigma(d) = \begin{cases} 2^{-\frac{4d(1-\alpha)}{4+d(1-\alpha)}} \left(\frac{d(1-\alpha)+4}{4-d(1-\alpha)} \right)^{\frac{4-d(1-\alpha)}{d(1-\alpha)+4}} & \text{if } 1 \leq d \leq 3, \\ 2^{-\frac{4(1-\alpha)}{2-\alpha}} \left(\frac{2-\alpha}{\alpha} \right)^{\frac{\alpha}{2-\alpha}} & \text{if } d \geq 4, \end{cases}$$

and

$$\rho = \rho(d) = \begin{cases} 2^{\frac{d(1-\alpha)}{2}} \left(\frac{d(1-\alpha)+4}{4-d(1-\alpha)} \right)^{\frac{4-d(1-\alpha)}{2d(1-\alpha)}} & \text{if } 1 \leq d \leq 3, \\ 2^{2(1-\alpha)} \left(\frac{2-\alpha}{\alpha} \right)^{\frac{\alpha}{2(1-\alpha)}} & \text{if } d \geq 4. \end{cases}$$

Theorem III.6. *Let $d \in \mathbb{N}^*$ and f be the energy density function such that $f \geq \eta F_\alpha$ for some $\eta > 0$, where F_α is defined in (3.25). Let E be as in (3.8), where v is the solution of (3.6). Assume $\mu > b$ and*

$$E(0) < \frac{1}{12} \frac{\eta(\mu - b)^{2-\alpha}}{\mu^{2(1-\alpha)}}.$$

Then there exist universal constants $C_l < \infty$ and $C_h < \infty$, such that, if we let T_ be as*

$$(3.26) \quad T_* = T_*(d) = \begin{cases} \frac{\rho}{C_l^{1-\alpha}} N^{d(1-\alpha)} L(0)^{\frac{4+d(1-\alpha)}{2}} & \text{if } 1 \leq d \leq 3, \\ \frac{\rho}{C_h^{1-\alpha}} N^{4(1-\alpha)} L(0)^{2(2-\alpha)} & \text{if } d \geq 4, \end{cases}$$

we have

- If $1 \leq d \leq 3$,

$$\frac{1}{T} \int_0^T E^2 dt \geq \sigma C_l^{\frac{8(1-\alpha)}{4+d(1-\alpha)}} \left[(N^4 T)^{-\frac{d(1-\alpha)}{d(1-\alpha)+4}} \right]^2,$$

provided $T \geq T_*$.

- If $d \geq 4$,

$$\frac{1}{T} \int_0^T E^2 dt \geq \sigma C_h^{\frac{2(1-\alpha)}{2-\alpha}} \left[(N^4 T)^{-\frac{1-\alpha}{2-\alpha}} \right]^2,$$

provided $T \geq T_*$.

The constants C_l and C_h are given by the interpolation inequalities (3.28) and (3.29) respectively.

3.3.1 Decay relation

In this section, we establish the first ingredient for applying Kohn and Otto's technique to our problem, namely the decay relation, also called the dissipation inequality.

Lemma III.7. $\left| \frac{dL}{dt} \right| \leq \left(-\frac{dE}{dt} \right)^{\frac{1}{2}}.$

Proof: Since the system (3.6) is gradient descent for the energy E with respect to the discrete H^{-2} norm, we can write

$$v_t = -\nabla_v E,$$

where the gradient is defined with respect to the discrete H^{-2} norm. Differentiating E with respect to t , we get

$$\frac{dE}{dt} = \langle \nabla_v E, v_t \rangle_{H^{-2}} = -\|v_t\|_{H^{-2}}^2.$$

Now differentiating $L^2 = \|v\|_{H^{-2}}^2$ with respect to t , we obtain

$$\left| 2L \frac{dL}{dt} \right| = 2 |\langle v_t, v \rangle_{H^{-2}}| \leq 2 \|v_t\|_{H^{-2}} \|v\|_{H^{-2}} = 2L \|v_t\|_{H^{-2}},$$

using the Cauchy-Schwartz inequality. We therefore obtain

$$\left| \frac{dL}{dt} \right| \leq \|v_t\|_{H^{-2}} = \left(-\frac{dE}{dt} \right)^{\frac{1}{2}},$$

thus proving the lemma. \square

3.3.2 Interpolation inequality

In this section, we establish the second ingredient for applying Kohn and Otto's technique to our problem: the interpolation inequality.

Lemma III.8. *Let the length scale L be defined as in (3.15) and let $\mu > b$. Assume $f \geq \eta F_\alpha$ for some constant $\eta > 0$ and $\alpha \in [0, 1[$. Let the energy E be defined as in (3.8), where v is the solution of (3.6), and assume*

$$(3.27) \quad E(0) < \frac{1}{12} \frac{\eta(\mu - b)^{2-\alpha}}{\mu^{2(1-\alpha)}}.$$

Then

- If $1 \leq d \leq 3$

$$(3.28) \quad E^{\frac{1}{1-\alpha}} L^{\frac{d}{2}} \geq \frac{C_l}{N^d},$$

for some $C_l > 0$ depending only on μ , b and α .

- If $d \geq 4$

$$(3.29) \quad E^{\frac{1}{1-\alpha}} L^2 \geq \frac{C_h}{N^4},$$

for some $C_h > 0$ depending only on μ , b and α .

We write E_{F_α} , the energy associated to the energy density function F_α defined in (3.25). Note that it suffices to consider the case $f = F_\alpha$ since if an interpolation inequality $E_{F_\alpha} L^\beta \geq \theta > 0$ holds, and $f \geq \eta F_\alpha$, then $EL^\beta \geq \eta\theta$. Similarly, the condition $E < C\eta$ implies $E_{F_\alpha} < C$. We therefore prove the interpolation inequality for the case $f = F_\alpha$. In the proof, we write E for E_{F_α} . As in [37], we prove the interpolation inequality in two steps and start by showing that the typical *positive* spike height is greater than $E^{\frac{-1}{1-\alpha}}$ in Lemma III.9. We note that Lemma III.9 is similar to Lemma 4 in [37], except that here we only consider the large *positive* spikes. If Lemma III.9 was the direct extension of Lemma 4 in [37], we would have considered all the large spikes (i.e., both negative and positive).

Lemma III.9. *Define the typical positive spike height h as*

$$(3.30) \quad h := \left(\frac{\mu - b}{3} \right)^{\frac{1}{1-\alpha}} E^{\frac{-1}{1-\alpha}}.$$

Define also the two sets \mathbb{S}_{lp} and \mathbb{S}_{lb} consisting of the large positive spikes, and points where the solution v_i is less than b (i.e. negative spikes and non spikes) respectively:

$$(3.31) \quad \begin{aligned} \mathbb{S}_{lp} &:= \{i \in \mathbb{L} : v_i - b > h\}, \\ \text{and } \mathbb{S}_{lb} &:= \{i \in \mathbb{L} : v_i \leq b\}. \end{aligned}$$

Then

$$\frac{1}{N^d} \sum_{i: b < v_i \leq b+h} (v_i - b) \leq \frac{\mu - b}{3},$$

and thus

$$\frac{1}{N^d} \sum_{i \in \mathbb{S}_{lp} \cup \mathbb{S}_{lb}} (v_i - b) \geq \frac{2}{3}(\mu - b).$$

Proof:

$$\begin{aligned}
\sum_{i:b < v_i \leq b+h} (v_i - b) &= \sum_{i:b < v_i \leq b+h} (v_i - b)^\alpha (v_i - b)^{1-\alpha} \\
&\leq \max_{i:b < v_i \leq b+h} (v_i - b)^{1-\alpha} \sum_{i:b < v_i \leq b+h} (v_i - b)^\alpha \\
&\leq h^{1-\alpha} \sum_{i:b \leq v_i \leq b+h} v_i^\alpha \\
&\leq N^d h^{1-\alpha} E = \frac{N^d (\mu - b)}{3}.
\end{aligned}$$

Thus using the conservation of mass

$$\frac{1}{N^d} \sum_{i \in \mathbb{S}_{lp} \cup \mathbb{S}_{lb}} (v_i - b) + \frac{1}{N^d} \sum_{i:b < v_i \leq b+h} (v_i - b) = \mu - b,$$

we obtain

$$\begin{aligned}
\frac{1}{N^d} \sum_{i \in \mathbb{S}_{lp} \cup \mathbb{S}_{lb}} (v_i - b) &= (\mu - b) - \frac{1}{N^d} \sum_{i:b < v_i \leq b+h} (v_i - b) \\
&\geq \frac{2}{3}(\mu - b).
\end{aligned}$$

This finishes the proof. \square

In the proof of the interpolation inequality, we need to show that the proportion of large positive spikes is bounded from above by $\frac{\mu}{h}$. For clarity purposes, we put this result in Lemma III.10. We note that in the second order case (see [37]), since the data v is positive for all times, Lemma III.10 is simply the Chebyshev inequality. For higher order cases however, the result is slightly more complicated to prove due to the loss of positivity in the data.

Lemma III.10. *Let the typical positive spike height h be defined as in (3.30) and let the set of large positive spikes \mathbb{S}_{lp} be defined as in (3.31). We define $|\mathbb{S}_{lp}|$ to be the number of large positive spikes. Then*

$$\frac{|\mathbb{S}_{lp}|}{N^d} \leq \frac{\mu}{h}.$$

Proof:

$$\begin{aligned} E &= \frac{1}{N^d} \sum_{i:|v_i| \geq b} |v_i|^\alpha \\ &\geq \frac{1}{N^d} \sum_{i:|v_i| \geq b+h} v_i^\alpha \\ &\geq \frac{|\mathbb{S}_{lp}|}{N^d} (b+h)^\alpha \geq \frac{|\mathbb{S}_{lp}|}{N^d} h^\alpha. \end{aligned}$$

Using the definition of h in (3.30), we obtain

$$h^{\frac{1}{1-\alpha}} \leq \left(\frac{\mu - b}{3} \right)^{\frac{1}{1-\alpha}} \left(\frac{N^d}{|\mathbb{S}_{lp}|} \right)^{\frac{1}{1-\alpha}},$$

and thus

$$|\mathbb{S}_{lp}| h \leq \left(\frac{\mu - b}{3} \right) N^d \leq \frac{\mu}{3} N^d \leq N^d \mu. \quad \square$$

Proof of Lemma III.8:

To estimate $L = \|v\|_{H^{-2}}$, we make use of its duality definition given in (3.12). In particular, recall from (3.12)

$$(3.32) \quad L \geq \frac{\frac{1}{N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \varphi_i}{\|\varphi\|_{H^2}},$$

where $\|\varphi\|_{H^2}^2 = \frac{1}{N^d} \sum_{i \in \mathbb{L}} (\Delta_\delta \varphi_i)^2$. We also note that the smallness condition (3.27) on E implies

$$(3.33) \quad h > 4^{\frac{1}{1-\alpha}} \frac{\mu^2}{\mu - b}.$$

Case 1: Let $1 \leq d \leq 3$.

In this case, we use a test function φ adapted to the large positive spikes. To construct this test function we consider a function $\mathcal{F} \in C_c^\infty(\mathbb{R}^d, \mathbb{R}^+)$, with support contained in $[-\frac{1}{2}, \frac{1}{2}]^d$ and L^1 norm equal to 1, e.g.

$$\mathcal{F}(x) = \begin{cases} C_0 e^{\frac{-1}{\frac{1}{4}-|x|^2}} & \text{if } |x| < \frac{1}{2} \\ 0 & \text{if } |x| \geq \frac{1}{2}, \end{cases}$$

where the constant C_0 is chosen such that $\int_{\mathbb{R}^d} \mathcal{F}(x) dx = 1$. We also let

$$(3.34) \quad k = \left\lceil \left[\left(\frac{\mu - b}{4\mu^2} h + 1 \right) \right]^{\frac{1}{d}} \right\rceil,$$

where h is the typical positive spike height defined in (3.30). The integer k is related to the radius (in number of grid points) of the cube on which the rescaled version of the function \mathcal{F} will be supported.

For $p = (p_1, \dots, p_d) \in \mathbb{S}_{lp}$ and $j \in \{1, 2, \dots, d\}$, we define the set S_j^m as

$$S_j^m := \{q \in \{0, 1, \dots, N-1\} : |q - p_j|_N \leq m\},$$

where $|r - s|_N = \min_{\ell \in \mathbb{N}} |r - s + \ell N|$ is the distance modulo N between the integers r and s in $\{0, 1, \dots, N-1\}$. Now for $m \in \mathbb{N}^*$ and p in \mathbb{S}_{lp} , we define the cube $\mathcal{Q}_m(p)$ as

$$(3.35) \quad \mathcal{Q}_m(p) := S_1^m \times \dots \times S_d^m.$$

In other words, $\mathcal{Q}_m(p)$ is the cube of radius m (in number of grid points) centered at the large positive spike located at $p \in \mathbb{S}_{lp}$. The cube $\mathcal{Q}_m(p)$ thus contains $(2m+1)^d$ grid points.

We now assume k is divisible by 4 (if not, the proof follows similarly with $\frac{k}{4}$ and $\frac{k}{2}$ replaced by $\lfloor \frac{k}{4} \rfloor$ and $2 \lfloor \frac{k}{4} \rfloor$ respectively), and define the two sets Σ_1 and Σ_2 as

$$\Sigma_1 = \bigcup_{p \in \mathbb{S}_{lp}} \mathcal{Q}_{\frac{k}{4}}(p),$$

and

$$\Sigma_2 = \bigcup_{p \in \mathbb{S}_{lp}} Q_{\frac{k}{2}}(p).$$

Σ_1 is the union of all the cubes of radius $\frac{k}{4}$ centered at the large positive spikes, and Σ_2 is the union of all the cubes of radius $\frac{k}{2}$ centered at the large positive spikes. Σ_2 is therefore a dilated version of Σ_1 , with dilation factor 2.

We now consider the restriction of the function \mathcal{F} on its support $\left[-\frac{1}{2}, \frac{1}{2}\right]^d$, and rescale it by a factor of $\frac{2N}{k}$ on $\left[-\frac{1}{2}, \frac{1}{2}\right]^d$. The rescaled version $\mathcal{F}\left(\frac{2N}{k}x\right)$ defined on $\left[-\frac{1}{2}, \frac{1}{2}\right]^d$ is then extended to \mathbb{R}^d by periodicity with period 1. We define $\overline{\mathcal{F}}\left(\frac{2N}{k}x\right)$ to be the extended function. We now define the vector \mathcal{F} on \mathbb{Z}^d as

$$\begin{aligned} \mathcal{F}_i &= \left(\frac{2N}{k}\right)^d \int_{\frac{i}{N} - \frac{1}{2N}}^{\frac{i}{N} + \frac{1}{2N}} \overline{\mathcal{F}}\left(\frac{2N}{k}x\right) dx, \\ &= N^d \int_{\frac{i}{\frac{k}{2}} - \frac{1}{2\frac{k}{2}}}^{\frac{i}{\frac{k}{2}} + \frac{1}{2\frac{k}{2}}} \overline{\mathcal{F}}(x) dx. \end{aligned}$$

The normalization factor $\left(\frac{2N}{k}\right)^d$ ensures that

$$\frac{1}{N^d} \sum_{i \in \mathbb{L}} \mathcal{F}_i = \|\mathcal{F}\|_{L^1(\mathbb{R}^d, \mathbb{R}^+)} = 1.$$

By construction, the support of the restricted vector \mathcal{F} to the lattice \mathbb{L} is contained in the cube $Q_{\frac{k}{4}}(0)$. Moreover since k^d is proportional to h , which is an increasing quantity, we see that the size of the support of the vector \mathcal{F} also increases as the system coarsens.

The test function φ is now defined on the lattice \mathbb{L} , as the discrete circular convolution between the vector \mathcal{F} and the characteristic function of the set Σ_1 :

$$(3.36) \quad \varphi_i = \frac{1}{N^d} \sum_{j \in \mathbb{L}} (\chi_{\Sigma_1})_j \mathcal{F}_{i-j}, \text{ for } i \in \mathbb{L}.$$

Claim:

$$(3.37) \quad \varphi_i = \begin{cases} 1 & \text{if } i \in \mathbb{S}_{lp} \\ 0 & \text{if } i \in (\Sigma_2)^\circ \\ a_i & \text{else, where } 0 \leq a_i \leq 1. \end{cases}$$

For clarity purposes, the proof of the claim is provided at the end of the section. Figures 3.4 and 3.5 illustrate the construction of the test function φ in the one dimensional case.

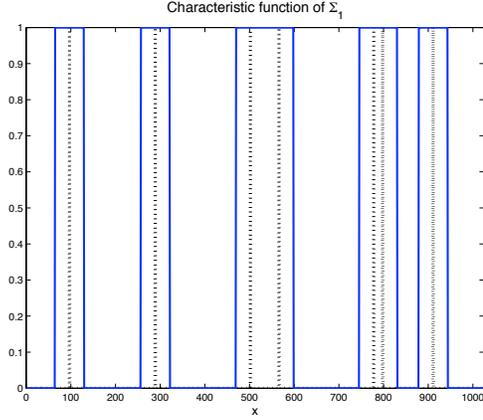


Figure 3.4: Characteristic function of the set Σ_1 . The dashed lines represent the locations of the large positive spikes.

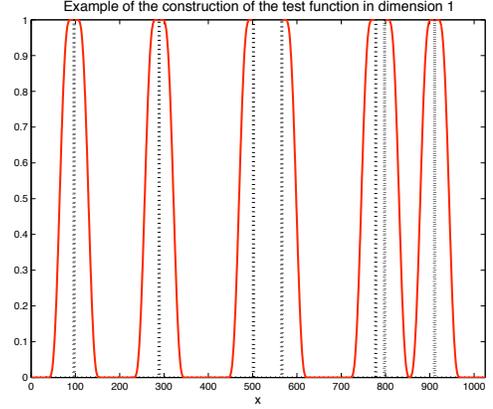


Figure 3.5: The dashed lines represent the locations of the large positive spikes.

Remark 1. As in [37], the test function φ is chosen to be as flat as possible, but localized enough for the term (3.38) to dominate (3.39) in the estimation of the length scale L .

We first bound the numerator of (3.32) from below and estimate

$$\begin{aligned}
 (3.38) \quad \frac{1}{Nd} \sum_{i \in \mathbb{L}} v_i \varphi_i &\geq \frac{1}{Nd} \sum_{i \in \mathbb{L}} (v_i - b) \varphi_i \\
 &= \frac{1}{Nd} \sum_{i \in \mathbb{S}_{lp}} (v_i - b) + \frac{1}{Nd} \sum_{i: b < v_i \leq b+h} (v_i - b) \varphi_i + \frac{1}{Nd} \sum_{i \in \mathbb{S}_{lb}} (v_i - b) \varphi_i \\
 &\geq \frac{1}{Nd} \sum_{i \in \mathbb{S}_{lp} \cup \mathbb{S}_{lb}} (v_i - b) \\
 &\geq \frac{2}{3} (\mu - b) \text{ (using Lemma III.9)}.
 \end{aligned}$$

Remark 2. The properties of φ given in (3.37) show that the support of the test function φ is a subset of Σ_2 with φ vanishing at the boundaries of Σ_2 . We can therefore write

$$\varphi_i = (\chi_{\Sigma_2})_i \left(\frac{1}{Nd} \sum_{j \in \mathcal{Q}_k(i)} (\chi_{\Sigma_1})_j \mathcal{F}_{i-j} \right) \leq (\chi_{\Sigma_2})_i, \quad i \in \mathbb{L}.$$

We therefore estimate

$$\begin{aligned}
(3.39) \quad \frac{\mu}{N^d} \sum_{i \in \mathbb{L}} \varphi_i &\leq \frac{\mu}{N^d} \sum_{i \in \mathbb{L}} (\chi_{\Sigma_2})_i \\
&\leq \frac{\mu}{N^d} \sum_{p \in \mathbb{S}_{lp}} \sum_{i \in Q_{\frac{k}{2}}(p)} 1 \\
&= \frac{\mu |\mathbb{S}_{lp}|}{N^d} k^d \\
&\leq \frac{\mu^2}{h} k^d \quad (\text{by Lemma III.10}) \\
&\leq \frac{\mu - b}{4} + \frac{\mu^2}{h} \left(\text{using } k^d \leq \frac{\mu - b}{4\mu^2} h + 1 \text{ from (3.34)} \right) \\
&\leq \frac{\mu - b}{4} + \left(\frac{1}{4} \right)^{\frac{1}{1-\alpha}} (\mu - b) \left(\text{using } h > 4^{\frac{1}{1-\alpha}} \frac{\mu^2}{(\mu - b)} \text{ from (3.33)} \right) \\
&\leq \frac{\mu - b}{2},
\end{aligned}$$

which combined with (3.38) gives

$$(3.40) \quad \frac{1}{N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \varphi_i \geq \frac{\mu - b}{6}.$$

We now estimate the H^2 norm of φ and first obtain a pointwise upper bound on $|\Delta_\delta \varphi_i|$:

$$\begin{aligned}
|\Delta_\delta \varphi_i| &= \left| (\chi_{\Sigma_2})_i \left(\frac{1}{N^d} \sum_{j \in Q_{\frac{k}{4}}(i)} (\chi_{\Sigma_1})_j \Delta_\delta \mathcal{F}_{i-j} \right) \right| \\
&\leq (\chi_{\Sigma_2})_i \frac{1}{N^d} \sum_{j \in Q_{\frac{k}{4}}(i)} |(\chi_{\Sigma_1})_j \Delta_\delta \mathcal{F}_{i-j}| \\
&= (\chi_{\Sigma_2})_i \left(\frac{2N}{k} \right)^2 \frac{1}{N^d} \sum_{j \in Q_{\frac{k}{4}}(i)} |(\chi_{\Sigma_1})_j \Delta_{\frac{\delta}{k}} \mathcal{F}_{i-j}| \\
&\leq (\chi_{\Sigma_2})_i \left(\frac{2N}{k} \right)^2 \frac{1}{N^d} \sum_{j \in Q_{\frac{k}{4}}(i)} |\Delta_{\frac{\delta}{k}} \mathcal{F}_{i-j}| \\
&= (\chi_{\Sigma_2})_i \left(\frac{2N}{k} \right)^2 \frac{1}{N^d} \underbrace{\sum_{j \in Q_{\frac{k}{4}}(0)} |\Delta_{\frac{\delta}{k}} \mathcal{F}_j|}_{\leq \|\Delta \mathcal{F}\|_{L^1(\mathbb{R}^d, \mathbb{R}^+)}} \\
&\leq (\chi_{\Sigma_2})_i \left(\frac{2N}{k} \right)^2 \|\Delta \mathcal{F}\|_{L^1(\mathbb{R}^d, \mathbb{R}^+)}.
\end{aligned}$$

For clarity, we let $C_1 = 4\|\Delta\mathcal{F}\|_{L^1(\mathbb{R}^d, \mathbb{R}^+)}$ and estimate the H^2 norm of φ :

$$\begin{aligned}
\|\varphi\|_{H^2}^2 &\leq \frac{1}{N^d} \sum_{i \in \mathbb{L}} (\Delta_\delta \varphi_i)^2 \\
&\leq C_1^2 \left(\frac{N}{k}\right)^4 \frac{1}{N^d} \sum_{i \in \mathbb{L}} (\chi_{\Sigma_2})_i \\
&\leq C_1^2 \left(\frac{N}{k}\right)^4 \frac{1}{N^d} \sum_{p \in \mathbb{S}_{lp}} \sum_{i \in \mathcal{Q}_{\frac{k}{2}}(p)} 1 \left(\text{since } \Sigma_2 = \bigcup_{p \in \mathbb{S}_{lp}} \mathcal{Q}_{\frac{k}{2}}(p) \right) \\
&= C_1^2 \left(\frac{N}{k}\right)^4 \frac{|\mathbb{S}_{lp}|}{N^d} k^d.
\end{aligned}$$

Thus

$$\begin{aligned}
\|\varphi\|_{H^2}^2 &\leq C_1^2 N^4 k^{d-4} \frac{|\mathbb{S}_{lp}|}{N^d} \\
&\leq \frac{\mu C_1^2}{h} N^4 k^{d-4} \quad (\text{by Lemma III.10}) \\
&\leq \frac{\mu C_1^2}{h} N^4 \left(\frac{4\mu^2}{(\mu-b)h} \right)^{\frac{4}{d}-1} \left(\text{using } k^d \geq \frac{\mu-b}{4\mu^2} h \text{ from (3.34) since } 1 \leq d < 4 \right) \\
&\leq \mu C_1^2 \left(\frac{4\mu^2}{\mu-b} \right)^{\frac{4}{d}-1} \frac{N^4}{h^{\frac{4}{d}}}.
\end{aligned}$$

The final estimate is

$$(3.41) \quad \|\varphi\|_{H^2} \leq \frac{\beta N^2}{h^{\frac{2}{d}}},$$

$$\text{with } \beta = C_1 \sqrt{\mu \left(\frac{4\mu^2}{\mu-b} \right)^{\frac{4}{d}-1}}.$$

Now using the definition of L , we have

$$\begin{aligned}
L = \|v\|_{H^{-2}} &\geq \frac{1}{N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \frac{\varphi_i}{\|\varphi\|_{H^2}} \\
&\geq \frac{h^{\frac{2}{d}}}{6\beta N^2} (\mu - b) \quad (\text{using (3.41) and (3.40)}) \\
&= \frac{1}{6N^2} \frac{E^{\frac{-2}{d(1-\alpha)}} (\mu - b)^{\frac{2}{d(1-\alpha)}+1}}{3^{\frac{2}{d(1-\alpha)}} \beta} \quad (\text{using the definition of } h \text{ in (3.30)}) \\
&= \frac{\lambda E^{\frac{-2}{d(1-\alpha)}}}{N^2},
\end{aligned}$$

where $\lambda = \frac{1}{6} \frac{(\mu-b)^{\frac{2}{d(1-\alpha)}+1}}{3^{\frac{2}{d(1-\alpha)}} \beta}$. Now, letting $C_l = \lambda^{\frac{d}{2}}$, we can write the above inequality as

$$(3.42) \quad E^{\frac{1}{1-\alpha}} L^{\frac{d}{2}} \geq \frac{C_l}{N^d},$$

which proves the first part of Lemma III.8.

Remark 3: The particular profile of the test function φ does not play much of a role in the determination of the coarsening rate. What matters is how the support of the smooth function \mathcal{F} scales with k , and through k , h . In fact, any function in the space $C_c^\infty(\mathbb{R}^d, \mathbb{R}^+)$ will give the same coarsening rate. We also note that this construction differs from the constructions given in [34] and [37]. In particular, it makes more transparent how the spatial dimension d and the order of the equation n completely determine the coarsening rate.

Case 2: Let $d \geq 4$.

In this case we define the test function φ on the lattice \mathbb{L} as the characteristic function of the set of large positive spikes \mathbb{S}_{lp} , namely $\varphi = \chi_{\mathbb{S}_{lp}}$. We remark that contrary to the low dimensional case ($1 \leq d \leq 3$), the test function is chosen to be supported on single spikes.

We first estimate its H^2 norm

$$\begin{aligned}\|\varphi\|_{H^2}^2 &\leq \frac{8|\mathbb{S}_{I_p}|d}{N^{d-4}} \\ &\leq \frac{8dN^4\mu}{h} \text{ (using Lemma III.10),}\end{aligned}$$

and obtain

$$(3.43) \quad \|\varphi\|_{H^2} \leq 2N^2 \sqrt{\frac{2d\mu}{h}}.$$

Now we estimate

$$\begin{aligned}\frac{1}{N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \varphi_i &= \frac{1}{N^d} \sum_{i \in \mathbb{S}_{I_p}} (v_i - \mu) \\ &\geq \frac{1}{N^d} \sum_{i \in \mathbb{S}_{I_p}} (v_i - b) - \mu \frac{|\mathbb{S}_{I_p}|}{N^d} \\ &\geq \frac{1}{N^d} \sum_{i \in \mathbb{S}_{I_p} \cup \mathbb{S}_{I_b}} (v_i - b) - \frac{\mu^2}{h} \text{ (using Lemma III.10)} \\ &\geq \frac{2}{3}(\mu - b) - \frac{\mu^2}{h} \text{ (using Lemma III.9)} \\ &\geq \frac{2}{3}(\mu - b) - \frac{\mu - b}{4} \left(\text{using } h > 4^{\frac{1}{1-\alpha}} \frac{\mu^2}{(\mu - b)} \text{ from (3.33)} \right) \\ &\geq \frac{5}{12}(\mu - b).\end{aligned}$$

Therefore

$$(3.44) \quad \frac{1}{N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \varphi_i \geq \frac{5}{12}(\mu - b).$$

Combining estimates (3.43) and (3.44), we obtain

$$\begin{aligned}L = \|v\|_{H^{-2}} &\geq \frac{1}{\|\varphi\|_{H^2} N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \varphi_i \\ &\geq \frac{5(\mu - b)}{24 \sqrt{2d} N^2} \sqrt{\frac{h}{\mu}} \\ &= \frac{5(\mu - b)^{\frac{3-2\alpha}{2(1-\alpha)}} E^{\frac{-1}{2(1-\alpha)}}}{24 \sqrt{6d\mu} N^2} \text{ (using the definition of } h \text{ in (3.30)).}\end{aligned}$$

Letting $C_h = \frac{25(\mu-b)^{\frac{3-2\alpha}{1-\alpha}}}{3456d\mu}$, we can write the above inequality as

$$(3.45) \quad E^{\frac{1}{1-\alpha}} L^2 \geq \frac{C_h}{N^4},$$

which proves the second part of Lemma III.8. \square

Combining the decay relation with the interpolation inequality, and using Kohn and Otto's ODE argument [58], we obtain the result of Theorem III.6.

Proof of the claim:

By construction, $0 \leq \varphi_i \leq 1$ for all $i \in \mathbb{L}$. We have $\varphi_i \geq 0$ because it is the sum of positive quantities, and $\varphi_i \leq 1$ because the vector \mathcal{F} is normalized to sum up to 1. Also we note that φ_i can be written as

$$\varphi_i = \frac{1}{N^d} \sum_{j \in \mathcal{Q}_{\frac{k}{4}}(i)} (\chi_{\Sigma_1})_j \mathcal{F}_{i-j},$$

since the support of the restricted vector \mathcal{F} to the lattice \mathbb{L} is contained in the cube $\mathcal{Q}_{\frac{k}{4}}(0)$.

- Now suppose $i \in \mathbb{S}_{lp}$. Since $\Sigma_1 = \bigcup_{p \in \mathbb{S}_{lp}} \mathcal{Q}_{\frac{k}{4}}(p)$, we know that $\mathcal{Q}_{\frac{k}{4}}(i) \subset \Sigma_1$, which implies $(\chi_{\Sigma_1})_j = 1$ for $j \in \mathcal{Q}_{\frac{k}{4}}(i)$. Thus φ_i simplifies to

$$\varphi_i = \frac{1}{N^d} \sum_{j \in \mathcal{Q}_{\frac{k}{4}}(i)} \mathcal{F}_{i-j} = \frac{1}{N^d} \sum_{j \in \mathcal{Q}_{\frac{k}{4}}(0)} \mathcal{F}_j = \frac{1}{N^d} \sum_{j \in \{-\frac{N}{2}, -\frac{N}{2}+1, \dots, \frac{N}{2}\}^d} \mathcal{F}_j = 1.$$

- Now suppose $i \in (\Sigma_2)^\circ$. Since Σ_2 is a dilated version by 2 of Σ_1 , we have by construction

$$\text{For all } i \in (\Sigma_2)^\circ \text{ and } j \in \Sigma_1, |i_q - j_q|_N \geq \frac{k}{4}, \text{ for all } 1 \leq q \leq d.$$

Therefore, if $i \in (\Sigma_2)^\circ$, we have $\mathcal{Q}_{\frac{k}{4}}(i) \subset (\Sigma_1)^c$ which implies $(\chi_{\Sigma_1})_j = 0$ for $j \in \mathcal{Q}_{\frac{k}{4}}(i)$.

Thus

$$\varphi_i = \frac{1}{N^d} \sum_{j \in \mathcal{Q}_{\frac{k}{4}}(i)} (\chi_{\Sigma_1})_j \mathcal{F}_{i-j} = 0. \quad \square$$

3.3.3 Rigorous result in terms of the spike density

In this section, we establish the rigorous result of Theorem III.6 in terms of the spike density. In other words, we show that for a certain interval of time, namely until the spike density gets low, the time-average of the spike density is bounded from below by a function of time that decays with the rate indicated by the one obtained in Theorem III.6.

We introduce the following notation. Let $K(t)$ be the number of spikes at time t , and let T_* be as in (3.26). Define also $I(T)$ to be

$$(3.46) \quad I(T) = \int_0^T \left(\frac{K(t)}{N^d} \right)^{2(1-\alpha)} dt.$$

Corollary III.11. *Let $\gamma \in]0, 1[$. Assume there exists a constant $\theta > 0$ such that $f(x) \leq \theta|x|^\alpha$ for all $|x|$ large enough, and assume also that μ is large enough such that $f(\mu) \leq \theta\mu^\alpha$. If $I(T_*) > \gamma$, then define T^* to be the first time at which I equals γ , i.e.,*

$$T^* = \min\{t \geq T_* : I(t) = \gamma\}.$$

Then for $T \in [T_*, T^*]$ and under the assumptions of Theorem III.6, there exist constants $C_1 < \infty$ and $C_2 < \infty$, such that

- If $1 \leq d \leq 3$,

$$I(T) \geq C_1 \left[(N^4 T)^{-\frac{d}{d(1-\alpha)+4}} \right]^{2(1-\alpha)}.$$

- If $d \geq 4$,

$$I(T) \geq C_2 \left[(N^4 T)^{-\frac{1}{2-\alpha}} \right]^{2(1-\alpha)}.$$

Proof: Let \mathbb{S} be the set of spikes and suppose $I(T_*) > \gamma$, which by definition implies

$I(T) \geq \gamma$ for $T \in [T_*, T^*]$. We define the energy of the spikes E_S to be

$$E_S := \frac{1}{N^d} \sum_{i \in \mathbb{S}} f(v_i),$$

and the energy of the non-spikes E_{NS} to be

$$E_{NS} := \frac{1}{N^d} \sum_{i \in \mathbb{L} \setminus \mathbb{S}} f(v_i),$$

where f is the energy density function. Then $E = E_{NS} + E_S$. We also define the mass of

the spikes to be $\mu_S := \frac{1}{N^d} \sum_{i \in \mathbb{S}} v_i$ and the mass of the non-spikes to be $\mu_{NS} := \frac{1}{N^d} \sum_{i \in \mathbb{L} \setminus \mathbb{S}} v_i$.

We easily see that

$$E_{NS} = \frac{1}{N^d} \sum_{i: |v_i| < b} f(v_i) \leq f(b).$$

For E_S , we have

$$E_S \leq \max \left\{ E_S : \frac{1}{N^d} \sum_{i \in \mathbb{S}} v_i = \mu_S \right\}.$$

Using the concavity of the function f for $|x| > b$, we can show that the maximum of the energy E_S is reached when all the spikes have the same height $|v_i| = \frac{N^d \mu_S}{K}$, assuming their mass μ_S is fixed. Moreover, since the heights of the non-spikes is always greater than $-b$, we can bound their mass μ_{NS} from below as

$$\mu_{NS} \geq -\left(1 - \frac{K}{N^d}\right)b,$$

and deduce that

$$\frac{N^d \mu_S}{K} \leq \frac{N^d}{K} (\mu + b) - b.$$

Therefore

$$\begin{aligned} E &\leq f(b) + \frac{K}{N^d} f\left(\frac{N^d}{K} (\mu + b) - b\right) \\ &\leq f(b) + \theta \frac{K}{N^d} \left(\frac{N^d}{K} (\mu + b) - b\right)^\alpha \left(\text{since } \frac{N^d}{K} (\mu + b) - b > \mu\right). \\ &\leq f(b) + \theta \frac{K}{N^d} \left(2 \frac{N^d}{K} \mu\right)^\alpha \left(\text{since } b + \mu < 2\mu\right) \\ &= f(b) + (2\mu)^\alpha \theta \left(\frac{K}{N^d}\right)^{1-\alpha}. \end{aligned}$$

Squaring both sides, we obtain

$$E^2 \leq 2 \left(f(b)^2 + (2\mu)^{2\alpha} \theta^2 \left(\frac{K}{N^d} \right)^{2(1-\alpha)} \right),$$

whose time-average satisfies

$$I(T) = \int_0^T \left(\frac{K}{N^d} \right)^{2(1-\alpha)} dt \geq \frac{1}{\lambda_1(\theta)\mu^{2\alpha}} \left(\int_0^T E^2 dt - 2f(b)^2 \right),$$

where $\lambda_1(\theta) = 2^{2\alpha+1}\theta^2$. If $T \in [T_*, T^*]$, we have by assumption that $I(T) \geq \gamma$, which implies

$$\frac{2f(b)^2}{\lambda_1(\theta)\mu^{2\alpha}} \leq \lambda_2(\gamma)I(T),$$

with $\lambda_2(\gamma) = \frac{2f(b)^2}{\gamma\lambda_1(\theta)\mu^{2\alpha}}$. Moreover, if $T \in [T_*, T^*]$, we can apply Theorem III.6 and therefore obtain

$$I(T) \geq \begin{cases} \frac{\sigma C_l^{\frac{8(1-\alpha)}{4+d(1-\alpha)}}}{(1+\lambda_2(\gamma))\lambda_1(\theta)\mu^{2\alpha}} \left[(N^4 T)^{-\frac{d(1-\alpha)}{d(1-\alpha)+4}} \right]^2 & \text{if } 1 \leq d \leq 3, \\ \frac{\sigma C_h^{\frac{2(1-\alpha)}{2-\alpha}}}{(1+\lambda_2(\gamma))\lambda_1(\theta)\mu^{2\alpha}} \left[(N^4 T)^{-\frac{1-\alpha}{2-\alpha}} \right]^2 & \text{if } d \geq 4. \end{cases}$$

This finishes the proof. \square

Remark. Since $I(T^*) = \gamma$, we can bound T^* from below by

$$T^* \geq \frac{1}{N^4} \begin{cases} T_l^0 = \left(\frac{\sigma C_l^{\frac{8(1-\alpha)}{4+d(1-\alpha)}}}{2(4^\alpha \mu^{2\alpha} \theta^2 \gamma + f(b)^2)} \right)^{\frac{d(1-\alpha)+4}{2d(1-\alpha)}} & \text{if } 1 \leq d \leq 3, \\ T_h^0 = \left(\frac{\sigma C_h^{\frac{2(1-\alpha)}{2-\alpha}}}{2(4^\alpha \mu^{2\alpha} \theta^2 \gamma + f(b)^2)} \right)^{\frac{2-\alpha}{2(1-\alpha)}} & \text{if } d \geq 4. \end{cases}$$

For μ large, C_l (3.42) scales like $\mu^{\frac{2\alpha+d(1-\alpha)}{2(1-\alpha)}}$ and C_h (3.45) scales like $\mu^{\frac{2-\alpha}{1-\alpha}}$, which implies

$$T_l^0 \sim \left(\frac{\mu^{\frac{4(2\alpha+d(1-\alpha))}{4+d(1-\alpha)}}}{\gamma\mu^{2\alpha} + f(b)^2} \right)^{\frac{d(1-\alpha)+4}{2d(1-\alpha)}} \quad \text{and} \quad T_r^0 \sim \left(\frac{\mu^2}{\gamma\mu^{2\alpha} + f(b)^2} \right)^{\frac{2-\alpha}{2(1-\alpha)}}.$$

Since μ is fixed, we can choose $\gamma > 0$ such that $\gamma\mu^{2\alpha} = O(1)$ and obtain

$$T_l^0 \sim \mu^{\frac{2(2\alpha+d(1-\alpha))}{d(1-\alpha)}} \quad \text{and} \quad T_r^0 \sim \mu^{\frac{2-\alpha}{1-\alpha}}.$$

Since both exponents $\frac{2(2\alpha+d(1-\alpha))}{d(1-\alpha)}$ and $\frac{2-\alpha}{1-\alpha}$ are greater or equal to 2 for any $\alpha \in [0, 1]$, we see that T_l^0 and T_r^0 can be very large. Thus T^* can be very large, which implies that Corollary III.11 can be valid for a very long time.

3.4 Upper bounds on coarsening for the $2n$ -th order equations

3.4.1 Equations and scheme

More generally, we study the following $2n$ -th order equation

$$(3.47) \quad v_t = (-1)^{n+1} \Delta^n (R(v)),$$

for $n \in \mathbb{N}^*$. We note that for n even, equation (3.47) is closely related to equation (3.2)

$$u_t = -\Delta^{\frac{n}{2}} (R(\Delta^{\frac{n}{2}} u(x))).$$

Taking the Laplacian raised to $\frac{n}{2}$ of both sides of equation (3.2) and setting $v = \Delta^{\frac{n}{2}} u$, we see that v satisfies (3.47) if u satisfies (3.2). For n odd, the connection is only true in the one dimensional case. To see that let's recall the one dimensional equation (3.2)

$$(3.48) \quad u_t = \partial_x^n R(\partial_x^n u(x)).$$

Taking n partial derivatives in x of both sides of equation (3.48) and setting $v = \partial_x^n u$, we see that v satisfies the one dimensional equation (3.47) if u satisfies (3.48).

We now discretize (3.47) on the same grid $[0, 1]^d$, and consider the system v_i satisfying the system of ODEs

$$(3.49) \quad \frac{dv_i}{dt} = (-1)^{n+1} (\Delta_\delta)^n (R(v_i)), \quad i \in \mathbb{L}.$$

We recall that we use periodic boundary conditions. Like (3.6), the system (3.49) is gradient descent for the energy E defined in (3.8) with respect to the discrete H^{-n} norm

$$\|v\|_{H^{-n}} := \begin{cases} \left\{ \sup_\phi \frac{1}{N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \phi_i : \frac{1}{N^d} \sum_{i \in \mathbb{L}} \left((\Delta_\delta)^{\frac{n}{2}} \phi_i \right)^2 \leq 1 \right\} & \text{if } n \text{ is even,} \\ \left\{ \sup_\phi \frac{1}{N^d} \sum_{i \in \mathbb{L}} (v_i - \mu) \phi_i : \frac{1}{N^d} \sum_{i \in \mathbb{L}} \left| \left(\nabla_\delta \Delta_\delta^{\frac{n-1}{2}} \right) \phi_i \right|^2 \leq 1 \right\} & \text{if } n \text{ is odd.} \end{cases}$$

Like (3.6), if the initial data has some mass in the ill-posed regime, the evolution of (3.49) gives rise to spikes whose coarsening behavior is very similar to the coarsening of (3.6).

Thus we define our length scale quantity L as

$$(3.50) \quad L := \|v\|_{H^{-n}}.$$

When chosen this way, we see that L scales with the number of spikes K as

$$L \sim \begin{cases} \frac{1}{K^{\frac{n}{d}}} & \text{for } 1 \leq d \leq 2n - 1, \\ \frac{1}{\sqrt{K}} & \text{for } d \geq 2n. \end{cases}$$

3.4.2 Main result

We introduce the following functions

$$\tilde{\sigma} = \tilde{\sigma}(d) = \begin{cases} 2^{-\frac{4d(1-\alpha)}{2n+d(1-\alpha)}} \left(\frac{d(1-\alpha)+2n}{2n-d(1-\alpha)} \right)^{\frac{2n-d(1-\alpha)}{d(1-\alpha)+2n}} & \text{if } 1 \leq d \leq 2n - 1, \\ 2^{-\frac{4(1-\alpha)}{2-\alpha}} \left(\frac{2-\alpha}{\alpha} \right)^{\frac{\alpha}{2-\alpha}} & \text{if } d \geq 2n, \end{cases}$$

and

$$\tilde{\rho} = \tilde{\rho}(d) = \begin{cases} 2^{\frac{d(1-\alpha)}{n}} \left(\frac{d(1-\alpha)+2n}{2n-d(1-\alpha)} \right)^{\frac{2n-d(1-\alpha)}{2d(1-\alpha)}} & \text{if } 1 \leq d \leq 2n - 1, \\ 2^{2(1-\alpha)} \left(\frac{2-\alpha}{\alpha} \right)^{\frac{\alpha}{2(1-\alpha)}} & \text{if } d \geq 2n. \end{cases}$$

Theorem III.12. *Let $d \in \mathbb{N}^*$ and let f be the energy density function such that $f \geq \eta F_\alpha$ for some $\eta > 0$, where F_α is defined in (3.25). Let E be as in (3.8) where v is the solution of (3.49). Assume $\mu > b$ and*

$$E(0) < \frac{1}{12} \frac{\eta(\mu - b)^{2-\alpha}}{\mu^{2(1-\alpha)}}.$$

Then there exist universal constants $\tilde{C}_l < \infty$ and $\tilde{C}_h < \infty$, such that, if we let \tilde{T}_ be as*

$$(3.51) \quad \tilde{T}_* = \tilde{T}_*(d) = \begin{cases} \frac{\tilde{\rho}}{\tilde{C}_l^{1-\alpha}} N^{d(1-\alpha)} L(0)^{\frac{2n+d(1-\alpha)}{n}} & \text{if } 1 \leq d \leq 2n - 1, \\ \frac{\tilde{\rho}}{\tilde{C}_h^{1-\alpha}} N^{2n(1-\alpha)} L(0)^{2(2-\alpha)} & \text{if } d \geq 2n, \end{cases}$$

we have

- *If $1 \leq d \leq 2n - 1$,*

$$\frac{1}{T} \int_0^T E^2 dt \geq \tilde{\sigma} \tilde{C}_l^{\frac{4n(1-\alpha)}{2n+d(1-\alpha)}} \left[(N^{2n} T)^{-\frac{d(1-\alpha)}{d(1-\alpha)+2n}} \right]^2,$$

provided $T \geq \tilde{T}_$.*

- If $d \geq 2n$,

$$\frac{1}{T} \int_0^T E^2 dt \geq \tilde{\sigma} \tilde{C}_h^{\frac{2(1-\alpha)}{2-\alpha}} \left[(N^{2n} T)^{-\frac{1-\alpha}{2-\alpha}} \right]^2,$$

provided $T \geq \tilde{T}_*$.

The constants \tilde{C}_l and \tilde{C}_h are given by the interpolation inequalities (3.52) and (3.53) respectively.

3.4.3 Decay relation and interpolation inequality

To prove Theorem III.12, we use the same argument as in the fourth order case, and establish a decay relation and an interpolation inequality. The decay relation remains the same as in the fourth order case since it only relies on the gradient descent property of the scheme. The difference lies in the interpolation inequality.

Lemma III.13. *Let the length scale L be defined as in (3.50) and let $\mu > b$. Assume $f \geq \eta F_\alpha$ for some constant $\eta > 0$ and $\alpha \in [0, 1[$. Let the energy E be defined as in (3.8) where v is the solution of (3.49), and assume*

$$E(0) < \frac{1}{12} \frac{\eta(\mu - b)^{2-\alpha}}{\mu^{2(1-\alpha)}}.$$

Then

- If $1 \leq d \leq 2n - 1$

$$(3.52) \quad E^{\frac{1}{1-\alpha}} L^{\frac{d}{n}} \geq \frac{\tilde{C}_l}{N^d},$$

for some $\tilde{C}_l > 0$ depending only on μ, b, n and α .

- If $d \geq 2n$

$$(3.53) \quad E^{\frac{1}{1-\alpha}} L^2 \geq \frac{\tilde{C}_h}{N^{2n}},$$

for some $\tilde{C}_h > 0$ depending only on μ , b , n and α .

Outline of the proof: This proof very closely follows the proof of Lemma III.8. In particular, we keep all the definitions introduced there, like the positive spike height h and the number of grid points k . Lemmas III.9 and III.10 therefore remain true. We also note that the only difference between the general case and the fourth order case is in the definition of L . In particular, the choice of test functions remains the same as in the fourth order case. Thus, estimates (3.40) and (3.44) still hold.

- Let $1 \leq d \leq 2n - 1$.

We define the test function φ as in (3.36) since the function \mathcal{F} is infinitely differentiable, and estimate the H^n norm of φ as in the fourth order case. In particular, $|\Delta_\delta \varphi_i|$ is replaced by $\left| \left(\Delta_\delta^{\frac{n}{2}} \right) \varphi_i \right|$ if n is even or $\left| \left(\nabla_\delta \Delta_\delta^{\frac{n-1}{2}} \right) \varphi_i \right|$ if n is odd. The final estimate is

$$(3.54) \quad \|\varphi\|_{H^n}^2 \leq \frac{\tilde{\beta} N^n}{h^{\frac{2n}{d}}},$$

where h is defined in (3.30) and $\tilde{\beta}$ is a constant depending only on μ , b , n and \mathcal{F} .

Combining (3.54) with (3.40), we obtain

$$L = \|v\|_{H^{-n}} \geq \frac{\tilde{\lambda} E^{-\frac{n}{d(1-\alpha)}}}{N^{2n}},$$

where $\tilde{\lambda} = \frac{1}{6} \frac{(\mu-b)^{\frac{2}{d(1-\alpha)}+1}}{3^{\frac{2}{d(1-\alpha)}\tilde{\beta}}}$. Letting $\tilde{C}_l = \tilde{\lambda}^{\frac{d}{n}}$, we obtain

$$E^{\frac{1}{1-\alpha}} L^{\frac{d}{n}} \geq \frac{\tilde{C}_l}{N^d}.$$

- Let $d \geq 2n$.

We consider the discrete function $\varphi = \chi_{\mathbb{S}_{lp}}$ as in the fourth order case. An estimate on its H^n norm gives

$$\|\varphi\|_{H^n}^2 \leq \frac{\tau(n)|\mathbb{S}_{lp}|d}{N^{d-2n}},$$

where $\tau(n)$ is a constant depending only on the order of derivatives. It can be worked out exactly but its exact value does not affect the bound on the coarsening rate. Using the result of Lemma III.10 we obtain

$$(3.55) \quad \|\varphi\|_{H^n}^2 \leq \frac{\tau(n)dN^{2n}\mu}{h}.$$

Combining estimate (3.44) with (3.55) we show that

$$L = \|v\|_{H^{-n}} \geq \frac{5(\mu - b)^{\frac{3-2\alpha}{2(1-\alpha)}} E^{-\frac{1}{2(1-\alpha)}}}{12N^n \sqrt{3\tau d\mu}}.$$

Letting $\tilde{C}_h = \frac{25(\mu-b)^{\frac{3-2\alpha}{1-\alpha}}}{432\tau d\mu}$, we can write the above inequality as

$$E^{\frac{1}{1-\alpha}} L^2 \geq \frac{\tilde{C}_h}{N^{2n}}.$$

This finishes the proof of Lemma III.13. \square

We now establish the analogue of Corollary III.11 and let \tilde{T}_* be as in (3.51).

Corollary III.14. *Let $I(T)$ be as in (3.46) and define \tilde{T}^* as in Corollary III.11, with T_* replaced by \tilde{T}_* . Then under the assumptions of Theorem III.12 and Corollary III.11, for $T \in [\tilde{T}_*, \tilde{T}^*]$, there exist constants $\tilde{C}_1 < \infty$ and $\tilde{C}_2 < \infty$ such that*

- If $1 \leq d \leq 2n - 1$,

$$I(T) \geq \tilde{C}_1 \left[(N^{2n}T)^{-\frac{d}{d(1-\alpha)+2n}} \right]^{2(1-\alpha)}.$$

- If $d \geq 2n$,

$$I(T) \geq \tilde{C}_2 \left[(N^{2n}T)^{-\frac{1}{2-\alpha}} \right]^{2(1-\alpha)}.$$

The proof is analogous to the one of Corollary III.11.

3.5 Numerical evidence and discussion

In this section, we present numerical experiments for the fourth order (i.e., You-Kaveh) and sixth order equation corroborating the rigorous statement of Corollary III.14, and discuss some of the implications of this result in image processing applications.

3.5.1 Numerical results

We show the actual coarsening rate of (3.47) by discretizing the system of ODEs (3.49) in time and solving it numerically as in [34, 37]. The graphs below show that the coarsening rate depends on N as our bounds indicate. We also note that our analysis does not include the early time dynamics since our results only hold for later times. We let $G(v_i^n) = -\Delta_h^n (R(v_i^n))$ and use the forward Euler method in time

$$(3.56) \quad \frac{v_i^{n+1} - v_i^n}{\delta t} = G(v_i^n).$$

As in [37], we choose the following density function f

$$(3.57) \quad f(x) = (1 + x^2)^{\frac{\alpha}{2}},$$

with $\alpha = \frac{1}{2}$. As an initial condition, we use a perturbation of $v_i = 2$. Using the result of Corollary III.11 and Corollary III.14, we have

- In the fourth order case

$$\int_0^T \left(\frac{K}{N^d}\right)^{2-2\alpha} dt \geq C \begin{cases} \left((N^4 T)^{\frac{-d}{d(1-\alpha)+4}}\right)^{2-2\alpha} & \text{for } 1 \leq d \leq 3 \\ \left((N^4 T)^{\frac{-1}{2-\alpha}}\right)^{2-2\alpha} & \text{for } d \geq 4, \end{cases}$$

for some constant $C < \infty$.

- In the sixth order case

$$\int_0^T \left(\frac{K}{N^d}\right)^{2-2\alpha} dt \geq C \begin{cases} \left((N^6 T)^{\frac{-d}{d(1-\alpha)+6}}\right)^{2-2\alpha} & \text{for } 1 \leq d \leq 5 \\ \left((N^6 T)^{\frac{-1}{2-\alpha}}\right)^{2-2\alpha} & \text{for } d \geq 6, \end{cases}$$

for some constant $C < \infty$.

In the figures below we show loglog plots of the spike density $\frac{K}{N^d}$ of the solution of (3.56) versus $N^{2n}T$, with f defined in (3.57), for $n = 2$ and $n = 3$, in support of the rigorous result stated in Corollary III.14. The rigorous bounds from Theorem III.12 and Corollary III.14 also lead to the following heuristic scalings for the length scale L

$$N^n L \sim \begin{cases} (N^{2n}T)^{\frac{n}{d(1-\alpha)+2n}} & \text{if } 1 \leq d \leq 2n - 1, \\ (N^{2n}T)^{\frac{1}{2(2-\alpha)}} & \text{if } d \geq 2n. \end{cases}$$

We provide plots for the length scale L to corroborate the above scalings. For the fourth order equation, our numerical results are presented in dimensions two, three, four and five to illustrate the fact that the coarsening rate does indeed become constant after dimension four. Figures 3.6, 3.8, 3.10, and 3.12 show the decay of the spike density $\frac{K}{N^d}$ versus $N^{2n}T$, superimposed with the theoretical coarsening rate $(N^{2n}T)^{-\frac{d}{d(1-\alpha)+2n}}$ in dimensions two, three, four and five respectively. Figures 3.7, 3.9, 3.11, and 3.13 display similar plots for the length scale L . The last two figures illustrate our numerical results for the sixth order equation in dimension two. Figure 3.14 shows the plot for the spike density $\frac{K}{N^d}$ and Figure 3.15 the plot for the length scale L .

The computational results presented below show very good agreement with the theoretical bounds of Sections 3.3 and 3.4 and their implications described above. In particular, although the rigorous results presented in this thesis are one-sided bounds, as in previous applications of the Kohn and Otto's technique, they seem to reflect the typically observed behavior of the dynamics. Indeed, after a brief initial period of rapid change, in our experiments both the spike density $\frac{K}{N^d}$ and the length scale quantity L settle into a rate that is remarkably close to the one-sided bounds. Nevertheless, a slight deviation from the bounds is present in the behavior of $\frac{K}{N^d}$ at dimensions $d \geq 4$ for the fourth order (i.e. You-Kaveh) equation (see Figures 3.10 and 3.12), whereas no such deviation can be discerned

in our experiments in the behavior of L (see Figures 3.11 and 3.13). A similar slight deviation was observed in [34] for the second order (i.e. Perona-Malik) equation at dimension $d = 2$. Based on these cases, it may be reasonable to suspect that in general, for the $2n$ -th order equation, the deviation appears at dimension $d = 2n$, which is when the interpolation inequality of Lemma III.13 switches from one form to the other.

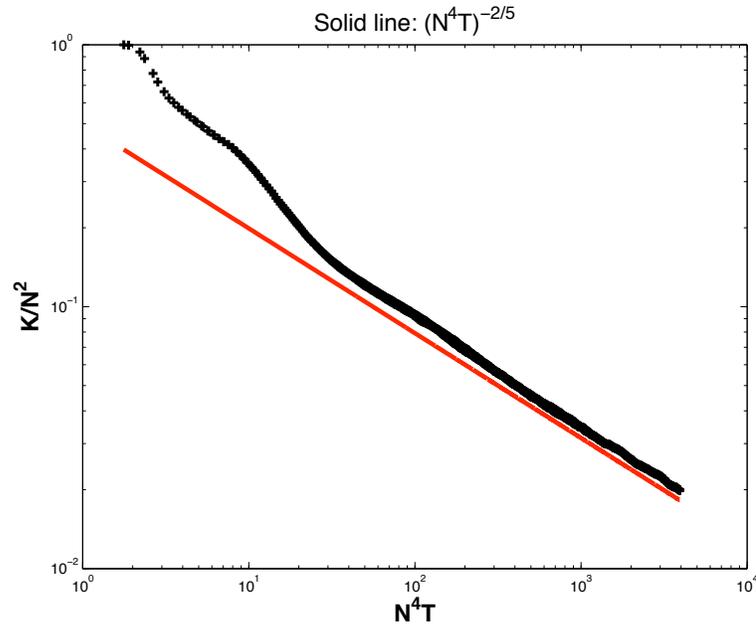


Figure 3.6: Experiments done with the two-dimensional You-Kaveh equation (3.6) with $f(x) = (1 + x^2)^{1/4}$. The spike density $\frac{K}{N^2}$ is represented by the dashed lines for $N = 175$ and $N = 200$. The coarsening rate $(N^4 T)^{-2/5}$ indicated by the bound obtained in Theorem III.6 is represented by the solid line. After an initial period, the spikes appear to coarsen at the predicted rate indicating that the bound seems close to optimal.

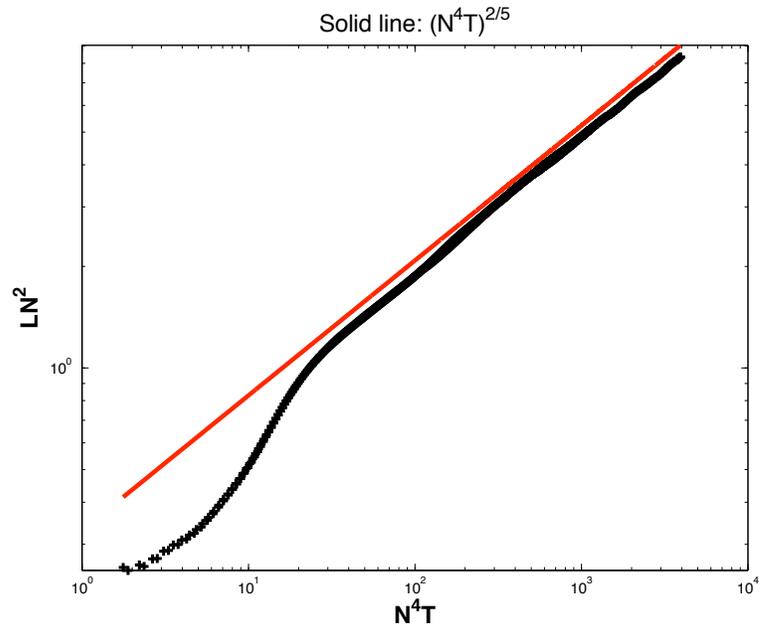


Figure 3.7: Experiments done with the two-dimensional You-Kaveh equation (3.6) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The rate $(N^4T)^{\frac{2}{5}}$ is represented by the solid line. The quantity LN^2 , represented by the dashed lines for $N = 175$ and $N = 200$, appears to coarsen at the rate $(N^4T)^{\frac{2}{5}}$ after an initial period of time. This shows that after a transient initial time, the length scale L behaves like $L \sim \frac{1}{k}$, validating our choice (3.15) as a length scale measure.

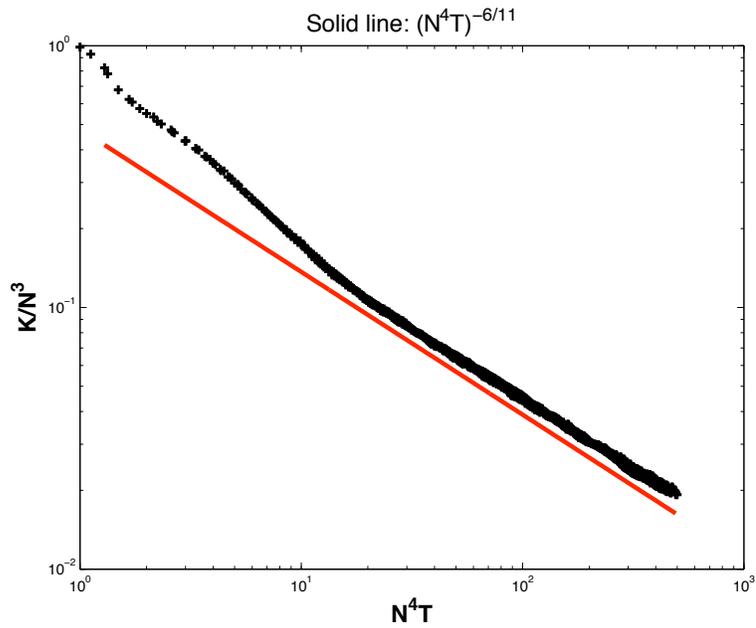


Figure 3.8: Experiments done with the three-dimensional You-Kaveh equation (3.6) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The spike density $\frac{K}{N^3}$ is represented by the dashed lines for $N = 20$ and $N = 25$. The coarsening rate $(N^4T)^{-\frac{6}{11}}$ indicated by the bound obtained in Theorem III.6 is represented by the solid line. After an initial period, the spikes appear to coarsen at the predicted rate indicating that the bound seems close to optimal.

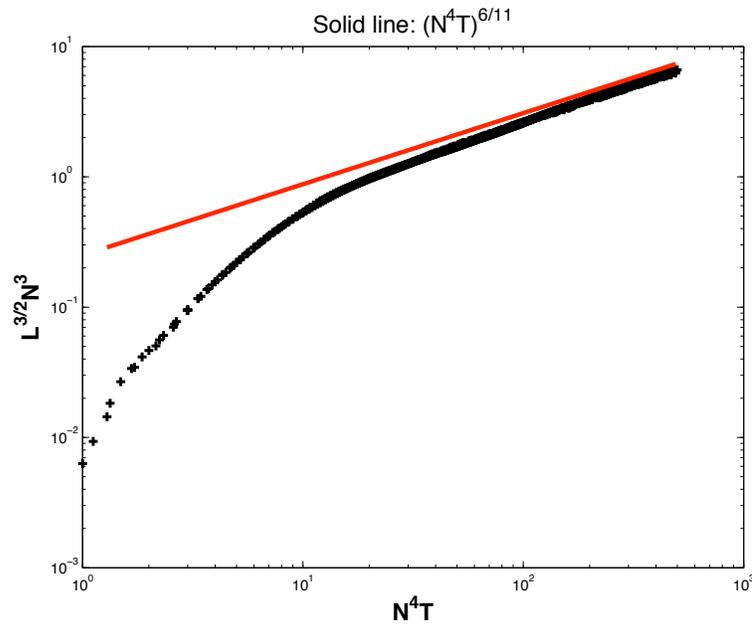


Figure 3.9: Experiments done with the three-dimensional You-Kaveh equation (3.6) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The rate $(N^4T)^{\frac{6}{11}}$ is represented by the solid line. The quantity $L^{\frac{3}{2}}N^3$, represented by the dashed lines for $N = 20$ and $N = 25$, appears to coarsen at the rate $(N^4T)^{\frac{6}{11}}$ after an initial period of time. This shows that after a transient initial time, the length scale L behaves like $L \sim \frac{1}{K^{\frac{3}{5}}}$, validating our choice (3.15) as a length scale measure.

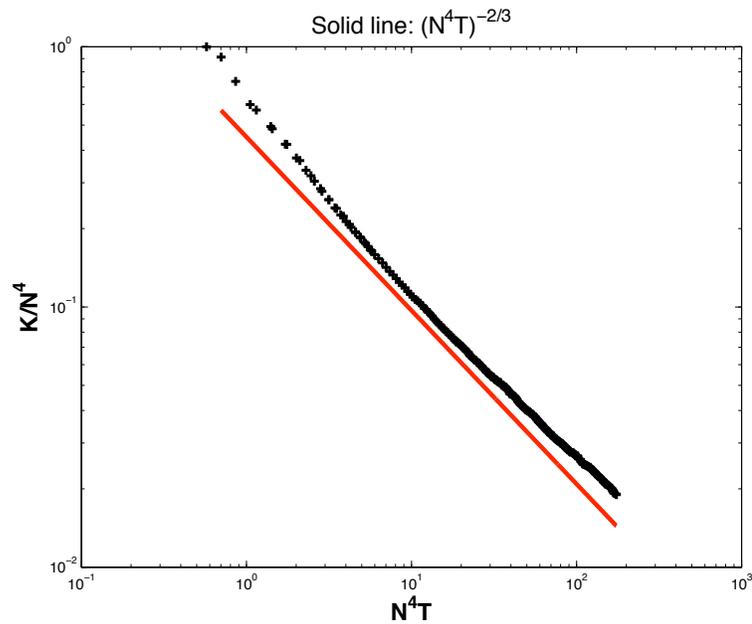


Figure 3.10: Experiments done with the four-dimensional You-Kaveh equation (3.6) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The spike density $\frac{K}{N^4}$ is represented by the dashed lines for $N = 10$ and $N = 15$. The coarsening rate $(N^4T)^{-\frac{2}{3}}$ indicated by the bound obtained in Theorem III.6 is represented by the solid line. After an initial period, the spikes appear to coarsen at the predicted rate indicating that the bound seems close to optimal.

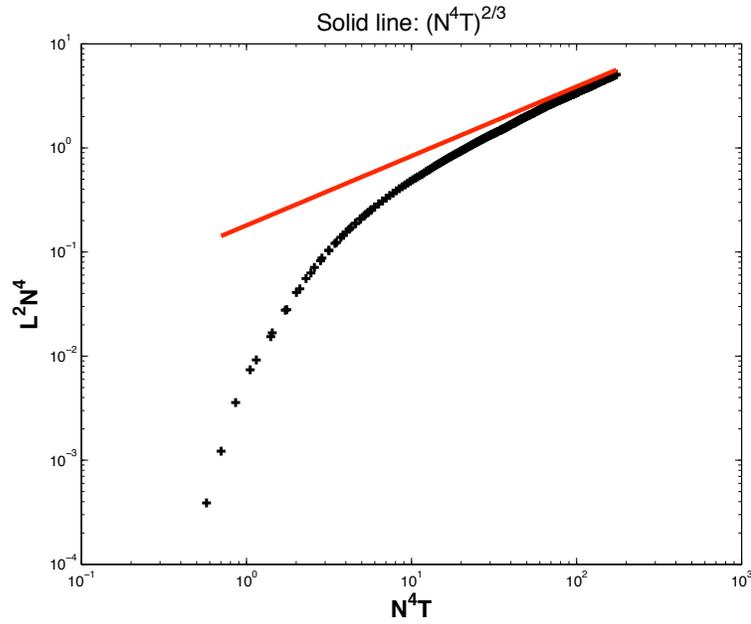


Figure 3.11: Experiments done with the four-dimensional You-Kaveh equation (3.6) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The rate $(N^4 T)^{\frac{2}{3}}$ is represented by the solid line. The quantity $L^2 N^4$, represented by the dashed lines for $N = 10$ and $N = 15$, appears to coarsen at the rate $(N^4 T)^{\frac{2}{3}}$ after an initial period of time. This shows that after a transient initial time, the length scale L behaves like $L \sim \frac{1}{\sqrt{k}}$, validating the choice (3.15) as a length scale measure.

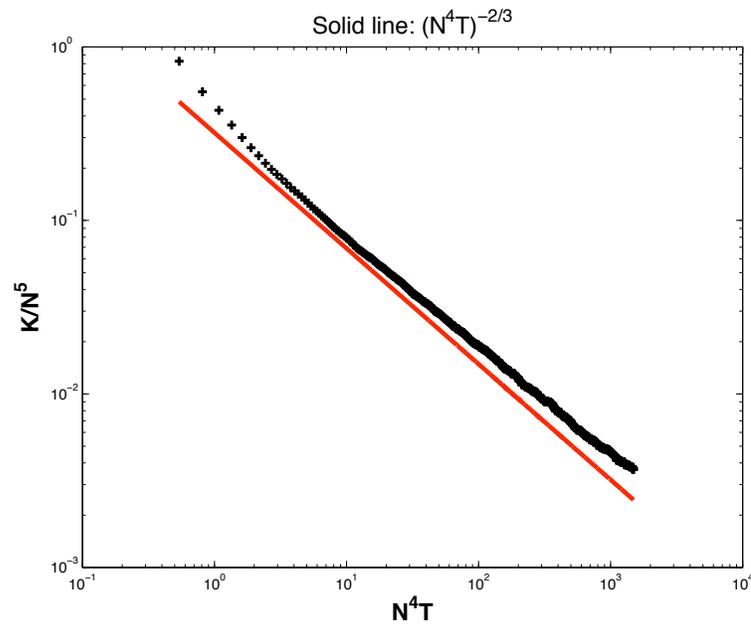


Figure 3.12: Experiments done with the five-dimensional You-Kaveh equation (3.6) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The spike density $\frac{K}{N^5}$ is represented by the dashed lines for $N = 7$. The coarsening rate $(N^4 T)^{-\frac{2}{3}}$ indicated by the bound obtained in Theorem III.6 is represented by the solid line. After an initial period, the spikes appear to coarsen at the predicted rate indicating that the bound seems close to optimal and corroborating the fact that the coarsening rate becomes constant after dimension four.

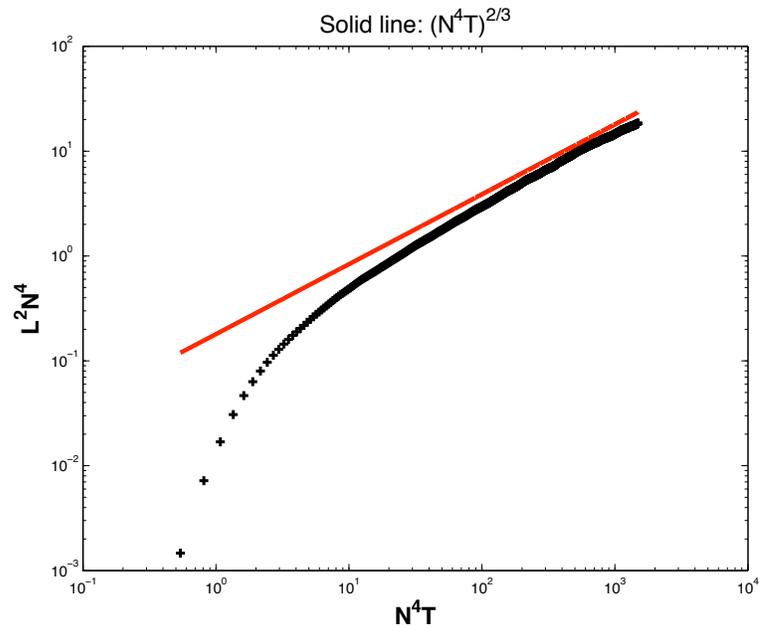


Figure 3.13: Experiments done with the five-dimensional You-Kaveh equation (3.6) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The rate $(N^4 T)^{\frac{2}{3}}$ is represented by the solid line. The quantity $L^2 N^4$, represented by the dashed lines for $N = 7$, appears to coarsen at the rate $(N^4 T)^{\frac{2}{3}}$ after an initial period of time. This shows that after a transient initial time, the length scale L behaves like $L \sim \frac{1}{\sqrt{K}}$, as in the four dimensional case.

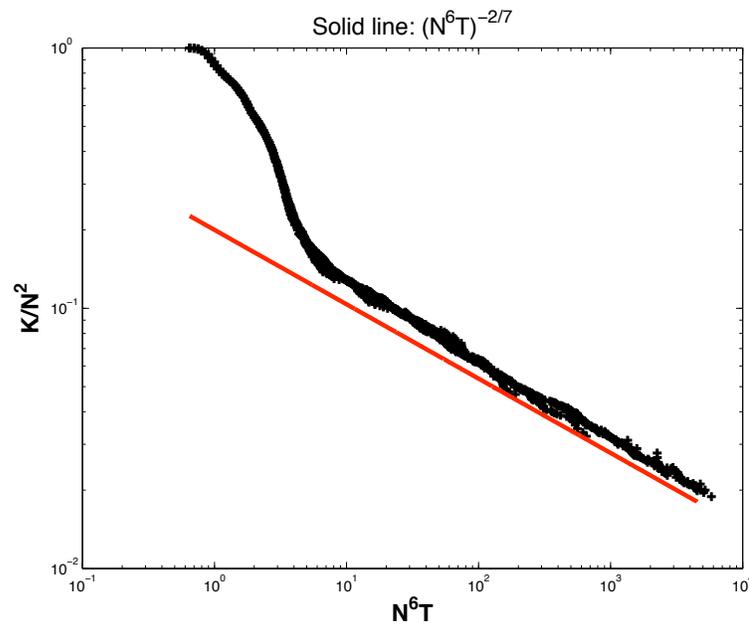


Figure 3.14: Experiments done with the two-dimensional sixth-order equation (3.49) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The spike density $\frac{K}{N^2}$ is represented by the dashed lines for $N = 30$, $N = 50$ and $N = 75$. The coarsening rate $(N^6 T)^{-\frac{2}{7}}$ indicated by the bound obtained in Theorem III.12 is represented by the solid line. After an initial period, the spikes appear to coarsen at the predicted rate indicating that the bound seems close to optimal.

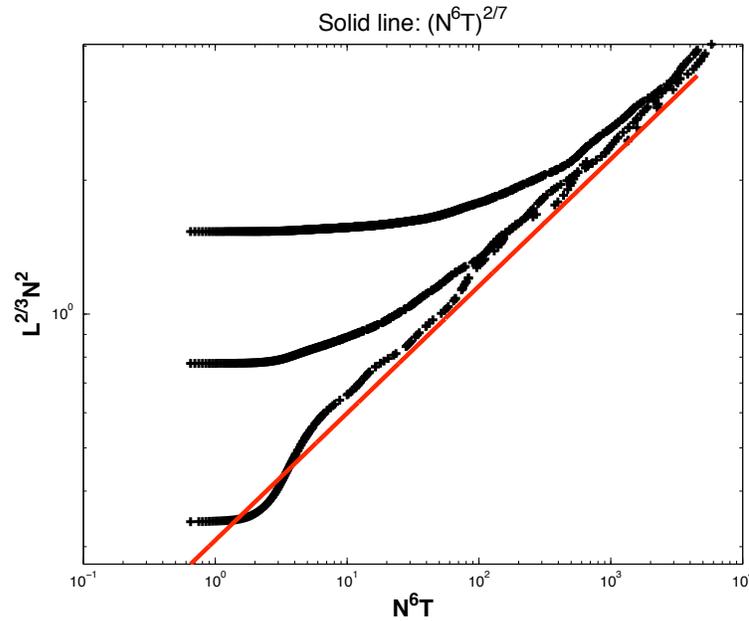


Figure 3.15: Experiments done with the two-dimensional sixth order equation (3.49) with $f(x) = (1 + x^2)^{\frac{1}{4}}$. The rate $(N^6 T)^{\frac{2}{7}}$ is represented by the solid line. The quantity $L^{\frac{2}{3}} N^2$, represented by the dashed lines for $N = 30$, $N = 50$ and $N = 75$, appears to coarsen at the rate $(N^6 T)^{\frac{2}{7}}$ after an initial period of time. This shows that after a transient initial time, the length scale L behaves like $L \sim \frac{1}{K^{\frac{1}{3}}}$, validating the choice (3.50) as a length scale measure.

3.5.2 Discussion

In practical applications to image processing of the schemes considered in this thesis (e.g., the You-Kaveh model (3.6)), a central question is the appropriate choice of parameters. In particular, given an image to be simplified (i.e., coarsened), it is often critical to know when to stop the time evolution of the processing equation. Since the complexity of an image is generally measured in terms of its “edge content”, Corollary III.14 presented in Section 3.4 may be interpreted to furnish a partial answer to this question. Although the results of this thesis cannot provide a universal, absolute value for the time at which a desired level of simplification in the image will be reached, they can be used to infer a scaling between the evolution time and the level of simplification: indeed, inverting the rigorous

statement for the spike density given in Corollary III.14, we arrive at the following

$$(3.58) \quad \begin{cases} T \sim \left(\frac{K}{N^d}\right)^{-\frac{d(1-\alpha)+2n}{d}} & \text{if } 1 \leq d \leq 2n - 1, \\ T \sim \left(\frac{K}{N^d}\right)^{-(2-\alpha)} & \text{if } d \geq 2n, \end{cases}$$

where T is the time of evolution and $\frac{K}{N^d}$ the spike density. An interesting facet of formula (3.58) is its dependence on n , the order of the equation up to a factor of two. We see that in low dimensions, namely $1 \leq d \leq 2n - 1$ (two, three and four being the most common in image processing), the coarsening rate (in terms of the exponent) slows down as the order of the equation increases. In particular, we see that in dimensions one, two and three, the fourth order You-Kaveh evolution leads to a slower simplification of the image than the second order Perona-Malik equation, potentially requiring a longer integration time. This observation concerns the continuous in time versions of the schemes, and is thus independent of the choice of time-stepping method used for the fully discrete system. It is an additional factor that needs to be taken into account when assessing the computational complexity of the various models, together with the usual stability restriction on the time step size that occurs in explicit schemes (i.e., the CFL condition which gets worse as the order of the equation increases). At this junction, it is worth mentioning the observation in [34] that implicit schemes for these ill-posed evolutions do not seem to yield the expected improvements in complexity: although implicit schemes can indeed be unconditionally stable, thus allowing for larger time steps, decreasing the number of spikes by a certain factor requires roughly the same number of time steps as an explicit scheme, regardless of the step size. A reasonable explanation for this observation is that large time steps cannot capture accurately enough the evolutions of spikes (a highly non smooth solution) and thus introduce drastic errors that alter the discrete simulations. There is therefore no real gain in processing speed by using implicit time-stepping methods for these equations. Nevertheless, despite their additional, inherent computational complexity, these You-Kaveh type

models find use in image processing due to the improved quality of their results. There is indeed a trade-off between computational time and quality of the processed image, but the final choice of which element is more important (i.e., computational speed versus quality of the results) is in the end up to the practitioner and dependent on the application.

CHAPTER IV

Conclusion and Future Directions

4.1 Conclusion

The results obtained in this dissertation describe computational and analytical research work in partial differential equations in image processing applications. In the algorithmic part of this dissertation, we present new algorithms for general area preserving motions in the plane. These algorithms provide efficient and highly accurate computational tools for generating area preserving geometric motions and can be applied in many situations and many applications, including (but not restricted to) image processing and material science. The second part of this dissertation provides rigorous bounds on the coarsening rates of nonlinear diffusion equations used to simplify (also denoise) images in computer vision applications. These bounds are crucial for understanding the coarsening rate of these nonlinear equations, and are a step towards the automatic determination of the stopping time of these evolutions, given a desired level of simplification in the resulting image. In addition to its practical impact, this result provides a novel proof for the interpolation inequality. The proof is modified from its precursor (second-order case in [34, 37]), and involves a new construction of the test functions. This novel construction is more general and provides insight on how the order of the equation and the dimension of the space come into play in the coarsening rate.

4.2 Future Directions

In this section, we discuss future directions of research that extend the work accomplished in this dissertation.

4.2.1 Volume preserving flows in higher dimensions

A natural extension of our area preserving algorithms described in Chapter II is to volume preserving flows in dimensions three and higher. This generalization should be relatively straightforward and would provide efficient and accurate algorithms for area preserving flows in high dimensions.

4.2.2 Fourth order flow for image segmentation and inverse problems

It would be interesting to investigate a variational alternative to the area preserving flow, which might be preferred to remove any shrinking bias. For that purpose, consider a Mumford-Shah based regularization model in which the usual perimeter term in the Mumford-Shah energy functional is replaced by

$$(4.1) \quad J = \int_{\partial\Sigma} |\kappa| d\sigma.$$

Like the area preserving flow, this term removes the natural shrinking bias of the Mumford-Shah model. In particular, $J = 2\pi$ for any convex curve. The minimization of this new energy functional leads to a fourth order evolution that can be easily computed using the new diffusion generated motion algorithms [35].

4.2.3 Second order flow for image segmentation and inverse problems

An alternative to replacing the perimeter term in the Mumford-Shah energy functional by J in (4.1) as discussed above, would be to modify the perimeter term in the two dimen-

sional Mumford-shah functional as

$$Q = \frac{\text{Per}\Sigma}{\sqrt{\text{Area}(\Sigma)}}.$$

In three dimensions, this term would become

$$Q = \frac{\text{Surf}\Sigma}{(\text{Vol}(\Sigma))^{\frac{2}{3}}}.$$

This regularization is an isoperimetric ratio for the set Σ and is therefore scaling invariant and minimum for circles or spheres. Thus, like the area preserving flow, this term removes the shrinking bias of the Mumford-Shah model. Gradient descent on the complete energy functional leads to a second order evolution that can also be easily computed using the new diffusion generated motion algorithms with signed distance functions.

4.2.4 Generalizations to all images

Working with piecewise smooth images as described in Section 2.4 of Chapter II is actually quite restrictive. An extension to general images, considers an image f , with gray level $f(x)$ at location x . If we assume a binary expansion of the gray level with 8 digits (from 0 to 255), we can describe f as

$$f(x) = \sum_{j=0}^7 2^j \mathbf{1}_{\Sigma_j}(x),$$

where Σ_j is the set of points x , where the binary expansion of $f(x)$ has a 1 in the j -th bit. Such generalization can be applied to our area preserving flows, as well as to the two variational models discussed above.

4.2.5 Coarsening in ill-posed diffusion equations for asymptotically constant energy densities

In this dissertation, we have obtained rigorous upper bounds on the coarsening rate of a family of discrete ill-posed diffusion equations. These bounds are valid for a class of energy densities f , that in essence grow like a power law $t \mapsto t^\alpha$, with $\alpha \in [0, 1[$. One

unsolved problem is the case of asymptotically constant energy densities that corresponds to a very fast decay of the function R in (3.2). It would be very interesting to investigate this case on the Perona-Malik scheme studied in [34, 37]

$$(4.2) \quad \frac{dv_i}{dt} = \Delta_\delta (R(v_i)) \quad i \in \mathbb{L} = \{0, 1, \dots, N-1\}^d,$$

where $\delta = \frac{1}{N}$. In this situation, the usual Kohn-Otto framework cannot be applied since the dynamics is driven by the deviation of the energy from its asymptotic constant. However, one of our ideas is to obtain a modified version of the Kohn-Otto argument using previous results on energy densities with power law growth at infinity. The approach would be to consider a modified energy density \tilde{f} with power law growth at infinity, the energy of which \tilde{E} is also decreased by (4.2). Using previous arguments (see [37]), we can relate \tilde{E} to the usual length scale L in an interpolation inequality. The more interesting question is now to prove a suitably modified dissipation inequality between $\frac{dL}{dt}$ and $\frac{d\tilde{E}}{dt}$, which combined with the interpolation inequality previously obtained, would lead to upper bounds on the coarsening rate of (4.2) for asymptotically constant energy densities. Once obtained for the second order equation (4.2), this analysis should be relatively easy to extend to high order diffusion equations using the work described in Chapter III (see also [60]).

4.2.6 Convergence of an algorithm for motions with normal velocities of the form $f(\kappa)$

Here we consider the convergence of the new diffusion generated motion algorithms introduced by Esedoğlu *et al.* [35], the extension of which to area preserving curvature flow was described in Chapter II. It would be interesting to look at the version of this algorithm that generates motions by $v_N = f(\kappa)$, where f is Lipschitz with constant L_f :

Algorithm: Given the initial set Σ_0 through its distance function $d_0(x)$ and a time step $\delta t > 0$, generate the sets Σ_j via their distance function $d_j(x)$ at the subsequent discrete times $t = j(\delta t)$ by alternating:

1. With any $M \geq L_f$, form $\mathcal{L}(x) = d_j + \delta t f \left(\frac{1}{M\delta t} \{G_{M\delta t} * d_j - d_j\} \right)$,
2. Construct the distance function d_{j+1} using $d_{j+1}(x) = \mathbf{Redist}(\mathcal{L}(x))$.

A proof of convergence of a similar algorithm for just standard mean curvature motion was recently given by Chambolle *et al.* in [22]. Since the algorithm described above is monotone, we anticipate that an analogous approach may be applicable for these more general motions.

The ultimate goal is to obtain a proof of convergence for our area preserving diffusion generated motion algorithms. However, due to the non monotonicity of these area preserving schemes, we feel that a more immediate goal should be geared towards obtaining a convergence proof for the diffusion generated motion algorithms generating motions with $v_N = f(\kappa)$ described above. From this proof, we expect to acquire some insights towards obtaining a convergence proof for our area preserving algorithms.

APPENDIX

APPENDIX A

A note on the piecewise smooth model of Vese and Chan

We consider the two phase piecewise smooth model of Vese and Chan [96]:

(A.1)

$$E(\Sigma, c_1, c_2) := \text{Per}(\Sigma) + \lambda \left(\int_{\Sigma} (g - c_1)^2 dx + \int_{\Sigma^c} (g - c_2)^2 dx \right) + \mu \left(\int_{\Sigma} |\nabla c_1|^2 dx + \int_{\Sigma^c} |\nabla c_2|^2 dx \right),$$

where c_1 and c_2 are smooth functions on the image domain Ω (e.g., $\Omega = [-5, 5]^2$). We show that (A.1) can remain uniformly bounded even as the segmentation converges to a function that is not “piecewise smooth”, i.e., one that would have infinite energy under any reasonable interpretation of (A.1). In particular, we exhibit that the lower semi-continuous envelope of (A.1) would allow cracks, and thus presumably reduce to a model as general as the original full Mumford-Shah, instead of providing a simplification.

We find an image g , a set Σ , functions c_1 and c_2 and sequences Σ_k , c_1^k and c_2^k converging respectively to Σ , c_1 and c_2 such that

$$E(\Sigma, c_1, c_2) > \liminf_{k \rightarrow \infty} E(\Sigma_k, c_1^k, c_2^k).$$

We consider the observed image g to be

$$(A.2) \quad g(z) = \Im \left(\sqrt{z-1} \sqrt{z+1} \right),$$

where $z \in \mathbb{C}$. The function g is smooth everywhere except at the branch cut $] -1, 1[$ where it is discontinuous, as shown in Figure A.1. Thus, its Dirichlet energy on the whole domain

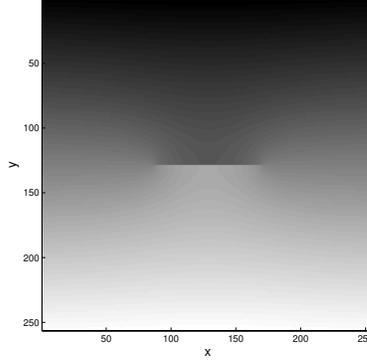


Figure A.1: Gray scale image of g described by equation (A.2)

Ω is infinite. We denote by Σ the branch cut $] - 1, 1[$ and consider the following sequence of sets Σ_k :

$$\Sigma_k := \left\{ x : |x - y| < \frac{1}{k} \text{ for all } y \in \Sigma \right\}.$$

By construction, the sequence Σ_k converges to Σ as $k \rightarrow \infty$ in L^1 . We now define the sequences

$$c_1^k(x) = 0 \text{ for all } x \in \Sigma_k \text{ and } c_2^k(x) = g(x) \text{ for all } x \in \Sigma_k^c,$$

and form

$$(A.3) \quad g_k(x) = c_1^k(x)\mathbf{1}_\Sigma(x) + c_2^k(x)\mathbf{1}_{\Sigma^c}(x).$$

Note that both c_1^k and c_2^k are smooth on their respective domain of definition. By construction we have

$$\lim_{k \rightarrow \infty} \int_{\Omega} (g_k(x) - g(x))^2 dx = 0,$$

since in the limit, g and g_k agree everywhere except on the set Σ of measure zero. The sequence g_k thus converges to g in the L^2 sense. The energy (A.1) of the limiting functional

g is thus

$$\begin{aligned} E(\Sigma, 0, g) &= \underbrace{\text{Per}(\Sigma) + \lambda \int_{\Sigma} g(x)^2 dx}_{=0 \text{ since } \Sigma \text{ has measure zero}} + \mu \int_{\Sigma} |\nabla g|^2 dx \\ &= \mu \int_{\Sigma} |\nabla g|^2 dx. \end{aligned}$$

Since g is discontinuous across the crack $] -1, 1[$, $\int_{\Sigma} |\nabla g|^2 dx$ should be taken as infinite and so should $E(\Sigma, 0, g)$.

Now, let's evaluate (A.1) on g_k :

$$\begin{aligned} E(\Sigma_k, c_1^k, c_2^k) &= \text{Per}(\Sigma_k) + \lambda \int_{\Sigma_k} g(x)^2 dx + \mu \int_{\Sigma_k^c} |\nabla g|^2 dx \\ &= 2 + \frac{2\pi}{k} + \lambda \int_{\Sigma_k} g(x)^2 dx + \mu \int_{\Sigma_k^c} |\nabla g|^2 dx \quad \left(\text{by construction } \text{Per}(\Sigma_k) = 2 + \frac{2\pi}{k} \right) \\ &\leq 2 + \frac{2\pi}{k} + \lambda \int_{\Omega} g(x)^2 dx + \mu \int_{\Sigma^c} |\nabla g|^2 dx. \end{aligned}$$

Since g is smooth on Σ^c and bounded on Ω , we know that both terms $\int_{\Omega} g(x)^2 dx$ and $\int_{\Sigma^c} |\nabla g|^2 dx$ are finite, namely there exists a constant $C < \infty$ such that

$$E(\Sigma_k, c_1^k, c_2^k) < C \text{ for all } k \in \mathbb{N}^*.$$

Thus

$$\liminf_{k \rightarrow \infty} E(\Sigma_k, c_1^k, c_2^k) < C < E(\Sigma, 0, g) = E(\lim_{k \rightarrow \infty} \Sigma_k, \lim_{k \rightarrow \infty} c_1^k, \lim_{k \rightarrow \infty} c_2^k),$$

which shows that (A.1), when relaxed, would assign finite energy to an image with an open curve as its discontinuity set.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing. *Arch. Rational Mech. Anal.*, 123:199–257, 1993.
- [2] H. Amann. Time-delayed Perona-Malik type problems. *Acta math. Univ. Comenianae*, LXXVI:15–38, 2007.
- [3] L. Ambrosio and V. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ -convergence. *Comm. Pure Appl. Math.*, 43:999–1036, 1990.
- [4] L. Bar, B. Berkels, G. Sapiro, and M. Rumpf. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *Proc. International Conference on Computer Vision (ICCV '07)*, Brazil, October 2007.
- [5] L. Bar, N. Sochen, and N. Kiryati. Semi-blind image restoration via mumford-shah regularization. *IEEE Trans. on Image Processing*, 15(2):483–493, 2006.
- [6] G. Barles and C. Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM Journal on Numerical Analysis*, 32(2):484–500, 1995.
- [7] A. Belahmidi and A. Chambolle. Time-delay regularization of anisotropic diffusion. *ESAIM Mathematical Modelling and Numerical Analysis*, 39(2):231–251, 2005.
- [8] G. Bellettini and G. Fusco. A regularized Perona-Malik functional: some aspects of the gradient dynamics. In H. Broer, J. Mahwin, A. Vanderbauwhede, and S. V. Lunel, editors, *EQUADIFF 2003 Proceedings of the International Conference on Differential Equations*, Hasselt, Belgium, July 2003.
- [9] G. Bellettini, M. Novaga, and E. Paolini. Global solutions to the gradient flow equation of a nonconvex functional. *SIAM J. Math. Anal.*, 37(5):1657–1687, 2006.
- [10] A. Braides. *Approximation of free-discontinuity problems*, volume 1694. Springer-Verlag, 1998.
- [11] A. Braides and G. Dal Maso. Non-local approximation of the Mumford-Shah functional. *Calc. Va. Partial Differ. Equ.*, 5(4):293–322, 1997.
- [12] L. Bronsard and B. Stoth. Volume preserving mean curvature flow as a limit of a nonlocal ginzburg-landau equation. *SIAM J. on Mathematical Analysis*, 28(4):769–807, 1997.
- [13] L. Bronsard and B. Wetton. A numerical method for tracking curve networks moving with curvature motion. *J. Comput. Phys.*, 120(1):66–87, 1993.
- [14] J. Canny. Finding edges and lines in images. *Tech. Report 720, Artificial Intelligence Laboratory, MIT, Boston, MA*, 1983.
- [15] I. Capuzzo Dolcetta, S. Finzi Vita, and R. March. Area preserving curve shortening flows: from phase separation to image processing. *Interfaces and Free Boundaries*, 4:325–343, 2002.
- [16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the 5th International Conference on Computer Vision*, pages 694–699, Boston, MA, June 1995. IEEE Computer Society Press.

- [17] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *The International Journal of Computer Vision*, 22(1):61–79, 1997.
- [18] F. Catté, P.L.Lions, J. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 29:182–193, 1992.
- [19] A. Chambolle. Image segmentation by variational methods: Mumford and Shah functional and the discrete approximation. *SIAM Journal of Applied Mathematics*, 55(3):827–863, 1995.
- [20] A. Chambolle and G. Dal Maso. Discrete approximations of the mumford-shah functional in dimension two. Technical Report no. 9820, Université Paris Dauphine, Ceremade, 1998.
- [21] A. Chambolle, R. DeVore, N. Lee, and B. Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7(3):319–334, 1998.
- [22] A. Chambolle and M. Novaga. Approximation of the anisotropic mean curvature flow. *Mathematical Models and Methods in Applied Sciences*, 17(6):833–844, 2007.
- [23] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.
- [24] L.-T. Cheng and Y.-H. Tsai. Redistancing by flow time dependent Eikonal equation. *J. Comput. Phys.*, 227(2):4002–4017, 2008.
- [25] S. Dai. On the shortening rate of collections of plane convex curves by the area-preserving mean curvature flow. *SIAM J. Math. Anal.*, 42(1):323–333, 2010.
- [26] E. De Giorgi and L. Ambrosio. Un nuovo tipo di funzionale del calcolo delle variazioni. *Att. Accad. Naz. Lincei, Rend. Cl. Sci. Fis. Mat. Nat.*, 82:199–210, 1988.
- [27] E. De Giorgi and T. Franzoni. Su un tipo di convergenza variazionale. *Atti Accad. Naz. Lincei, Rend. Cl. Sci. Fis. Mat. Natur.*, 58:842–850, 1975.
- [28] M. C. Delfour and J.-P. Zolesio. Shapes and geometries. analysis, differential calculus and optimization. *Advances in Design and Control, SIAM*, 2001.
- [29] R. DeVore, B. Jawerth, and B. Lucier. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory, Special Issue Wavelet Transforms Multires. Anal.*, 38:719–746, 1992.
- [30] D. Donoho. De-noising by soft thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.
- [31] D. Donoho. Nonlinear solution of linear inverse problems by wavelet-vaguelet decomposition. *Applied and Computational Harmonic Analysis*, 2:101–126, 1995.
- [32] D. Donoho and I. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of American Statistical Association*, 90:1200–1224, 1995.
- [33] S. Esedoğlu. An analysis of the Perona-Malik scheme. *Comm. Pure Appl. Math.*, 54:1442–1487, 2001.
- [34] S. Esedoğlu and J. Greer. Upper bounds of the coarsening rate of discrete ill-posed nonlinear diffusion equations. *Comm. Pure Appl. Math.*, 62(1):57–81, 2009.
- [35] S. Esedoğlu, S. Ruuth, and R. Tsai. Diffusion generated motion using signed distance functions. *Journal of Computational Physics*, 229(4):1017–1042, 2010.
- [36] S. Esedoğlu, S. J. Ruuth, and R. Tsai. Threshold dynamics for high order geometric motions. *Interfaces and Free Boundaries*, 10(3):263–282, 2008.

- [37] S. Esedoğlu and D. Slepčev. Refined upper bounds of the coarsening rate of discrete ill-posed nonlinear diffusion equations. *Nonlinearity*, 21(1):2759–2776, 2008.
- [38] S. Esedoğlu and Y.-H. R. Tsai. Threshold dynamics for the piecewise constant Mumford-Shah functional. *Journal of Computational Physics*, 211:367–384, 2006.
- [39] L. Evans. Convergence of an algorithm for mean curvature motion. *Indiana Univ. Math. J.*, 42:553–557, 1993.
- [40] H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93:418–491, 1959.
- [41] M. Gage. Curve shortening makes convex curves circular. *Invent. Math.*, 76:357–364, 1984.
- [42] M. Gage. On an area-preserving evolution equation for plane curves. *Contemp. Math.*, 51:51–62, 1986.
- [43] M. Gage and R. Hamilton. The heat equation shrinking convex plane curves. *J. Differential Geom.*, 23(1):69–96, 1986.
- [44] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [45] M. Ghisi and M. Gobbino. A class of local classical solutions for the one-dimensional Perona-Malik equation. *Preprint*, 2006.
- [46] M. Ghisi and M. Gobbino. Gradient estimates for the Perona-Malik equation. *Math. Ann.*, 337(3):557–590, 2007.
- [47] G. Gimel'farb. *Image Textures and Gibbs Random Fields*. Kluwer Academic Publishers, 1999.
- [48] M. Gobbino. Finite difference approximation of the Mumford-Shah functional. *Comm. on Pure and Applied Mathematics*, 51(2):197–228, 1998.
- [49] M. Gobbino. Entire solutions of the one-dimensional Perona-Malik equation. *Comm. Partial Differential Equation*, 32(4-6):719–743, 2007.
- [50] D. Golovaty. The volume preserving motion by mean curvature as an asymptotic limit of reaction-diffusion equations. *Q. of Appl. Math.*, LV:243–298, 1997.
- [51] J. Goodman, A. Kurganov, and P. Rosenau. Breakdown in Burgers-type equations with saturating dissipation fluxes. *Nonlinearity*, 12:247–268, 1999.
- [52] J. Greer and A. Bertozzi. Traveling wave solutions of fourth order PDEs for image processing. *SIAM J. Math. Anal.*, 36(1):38–68, 2004.
- [53] E. Hoetzl and W. Ring. An active contour approach for a mumford-shah model in x-ray tomography. *ISVC*, 2:1021–1030, 2009.
- [54] G. Huisken. Flow by mean curvature of convex surfaces into spheres. *J. Differential Geom.*, 20(1):237–266, 1984.
- [55] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. In *First International Conference on Computer Vision*, pages 259–268, London, June 1987.
- [56] B. Kawohl and N. Kutev. Maximum and comparison principle for one-dimensional anisotropic diffusion. *Math. Ann.*, 311:107–123, 1998.
- [57] S. Kichenassamy. The Perona-Malik paradox. *SIAM J. Appl. Math.*, 57(5):1328–1342, 1997.
- [58] R. Kohn and F. Otto. Upper bounds on coarsening rates. *Comm. Math. Phys.*, 229(3):375–395, 2002.

- [59] R. Kohn and X. Yan. Upper bound on the coarsening rate for an epitaxial growth model. *Comm. Pure Appl. Math.*, 56(11):1549–1564, 2003.
- [60] C. Kublik. Coarsening in high order, discrete, ill-posed diffusion equations. *To appear in Communications in Mathematical Sciences*.
- [61] S. Li. *Markov Random Field Modeling in Computer Vision*. Springer Verlag, 1995.
- [62] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.
- [63] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Roy. Soc. London Ser. B*, pages 187–217, 1980.
- [64] P. Mascarenhas. Diffusion generated motion by mean curvature. Technical report, UCLA, July 1992. CAM Report 92-33.
- [65] B. Merriman, J. K. Bence, and S. Osher. Diffusion generated motion by mean curvature. Technical report, UCLA, April 1992. CAM Report 92-18.
- [66] B. Merriman, J. K. Bence, and S. Osher. Diffusion generated motion by mean curvature. In AMS, editor, *Proceedings of the Computational Crystal Growers Workshop*, pages 73–83, Providence, RI, 1992.
- [67] B. Merriman, J. K. Bence, and S. Osher. Motion of multiple junctions: a level set approach. *Journal of Computational Physics*, 112(2):334–363, 1994.
- [68] L. Modica and S. Mortola. Un esempio di Γ -convergenza. *Boll. Un. Mat. Ital.*, 5(14-B):285–299, 1977.
- [69] W. W. Mullins. Two-dimensional motion of idealized grain boundaries. *J. Appl. Phys.*, 27:900–904, 1956.
- [70] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 62:577–685, 1989.
- [71] M. Nitzberg and T. Shiota. Nonlinear image filtering with edge and corner enhancement. *IEEE Trans. Pattern Anal. Machine Intell.*, 14:629–639, 1992.
- [72] S. Osher and R. Fedkiw. Level set methods: An overview and some recent results. *J. Comp. Phys.*, 169:463–502, 2001.
- [73] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comp. Phys.*, 79:12–49, 1988.
- [74] F. Otto, T. Rump, and D. Slepčev. Coarsening rates for a droplet model: Rigorous upper bounds. *SIAM J. Math. Anal.*, 32(12):503–529, 2006.
- [75] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang. A pde-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.
- [76] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. Technical report, Dept. of EECS, U.C. Berkeley, 1987.
- [77] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern. Anal. Machine Intell.*, 12:629–639, 1990.
- [78] P. Perona, J. Malik, and T. Shiota. *Geometry-Driven Diffusion in Computer Vision*, chapter Anisotropic diffusion, pages 73–92. Kluwer, 1995.
- [79] R. Ramlau and W. Ring. A Mumford-Shah level set approach for the inversion and segmentation of x-ray tomography data. *J. Comp. Phys.*, 221(2):539–557, 2007.

- [80] J. Rubinstein and P. Sternberg. Nonlocal reaction-diffusion equations and nucleation. *IMA J. Appl. Math.*, 48:248–264, 1992.
- [81] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [82] G. Russo and P. Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163:51–67, 2000.
- [83] S. J. Ruuth. A diffusion generated approach to multiphase motion. *J. Comput. Phys.*, 145(1):166–192, 1998.
- [84] S. J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *J. Comput. Phys.*, 144(2):603–625, 1998.
- [85] S. J. Ruuth and B. Merriman. Convolution generated motion and generalized Huygens’ principles for interface motion. *SIAM J. Appl. Math.*, 60(33):868–890, 2000.
- [86] S. J. Ruuth and B. Merriman. Convolution-thresholding methods for interface motion. *J. Comput. Phys.*, 169(2):678–707, 2001.
- [87] S. J. Ruuth, B. Merriman, and S. Osher. Convolution generated motion as a link between cellular automata and continuum pattern dynamics. *J. Comput. Phys.*, 151:836–861, 1999.
- [88] S. J. Ruuth, B. Merriman, J. Xin, and S. Osher. Diffusion-generated motion by mean curvature for filaments. *J. Nonlinear Science*, 11(6):473–493, 2001.
- [89] S. J. Ruuth and B. T. R. Wetton. A simple scheme for volume-preserving motion by mean curvature. *Journal of Scientific Computing*, 19(1-3):373–384, 2003.
- [90] J. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [91] P. Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. *Journal of Scientific Computing*, 19(1-3):439–456, 2003.
- [92] A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston and Sons, Washington, D.C., 1977.
- [93] Y.-H. Tsai, L. Cheng, S. Osher, and H.-K. Zhao. Fast sweeping methods for a class of Hamilton-Jacobi equations. *SIAM Journal on Numerical Analysis*, 41(2):673–694, 2003.
- [94] Y. R. Tsai and S. Osher. Total variation and level set based methods in image science. *Acta Numerica*, pages 1–61, 2005.
- [95] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40:1528–1538, 1995.
- [96] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.
- [97] N. Walkington. Algorithms for computing motion by mean curvature. *SIAM J. on Num. Anal.*, 33(6):2215–2238, 1996.
- [98] T. Witelski, D.G. Schaeffer, and M. Shearer. A discrete model for an ill-posed nonlinear parabolic PDE. *Phys. D.*, 160(3-4):189–221, 2001.
- [99] A. Witkin. Scale-space filtering. *Int. Joint. Conf. Artificial Intelligence*, Karlsruhe, West Germany:1019–1021, 1983.

- [100] Y.-L. You and M. Kaveh. Fourth-order partial differential equations for noise removal. *IEEE Transactions on Image Processing*, 9(10):1723–1730, 2000.
- [101] H. Zhao. A fast sweeping method for Eikonal equations. *Mathematical of Computation*, 74(250):603–627, 2005.
- [102] H.-K. Zhao, B. Merrimann, S. Osher, and L. Wang. Capturing the behavior of bubbles and drops using the variational level set approach. *J. Comput. Phys.*, 143:495–518, 1998.