ENSTA PARISTECH
UNIVERSITÉ PARIS-SACLAY

RAPPORT DE STAGE

Numerical simulation using the minimizing movement with the 1-Wasserstein distance

---

Auteur:
Théo GOLVET

Enseignant référent ENSTA:
Pierre CARPENTIER

Superviseurs:
Jean-David BENAMOU
Guillaume CARLIER

Mention de confidentialité
Rapport non confidentiel et publiable sur internet.

Compte-rendu du stage effectué dans l'équipe

MOKAPLAN, Inria Paris
du 03/04/18 au 07/09/18

September 10, 2018

Abstract

We study the numerical implementation of a minimizing movement scheme using the 1-Wasserstein distance introduced in [1]. We implement this scheme using two different primal dual algorithm : a Augmented Lagrangian technique with the ADMM algorithm and the Chambolle-Pock algorithm. This scheme is used to solve a class of nonlinear evolution equations which can be seen as a model for growing sandpiles or for compression molding of plastic.

Remerciements

Je remercie tout d'abord Jean-David Benamou et Guillaume Carlier de m'avoir confié ce stage. Outre le sujet stimulant et leur aide précieuse (bien que j'ai pu être trop peu loquace), il s'agit pour moi d'une excellente introduction dans le monde de la recherche en France et plus particulièrement dans le petit monde du transport optimal dont je devrais encore faire partie pendant au moins quelques temps.
Je souhaiterais aussi remercier tous les autres membres de Mokaplan et de l'Inria pour leur accueil chaleureux, leur bonne humeur et pour les nombreuses discussions plus ou moins sérieuses. Merci à Lucas, Andrea, César, Simon, Irène, Giorgi, Thomas, Vincent, Géraldine et les autres.
Je remercie tout particulièrement François-Xavier Vialard et Guillaume Legendre pour avoir accepté de m'encadrer en thèse sur un sujet qui s'annonce passionnant et j'espère que notre collaboration sera fructueuse.
Merci à Pierre Carpentier d'avoir accepté d'être mon enseignant référent pour ce stage et merci à Filippo Santambrogio pour son dévouement à l'égard de ses élèves.
Enfin j'exprime mes plus chaleureux remerciements à tous mes proches qui, même en dehors du travail, ont contribué à rendre ce stage, et plus généralement ces dernières années d'enseignement supérieur, plus agréables. Il serait trop long de remercier tout le monde en détail mais merci à ma famille chez qui chaque séjour (bien que rare) est précieux. Merci à Servane de me supporter et de me soutenir à tout moment, il m'est impossible d'exprimer toute la gratitude que j'ai à son égard. Merci à Laurent d'être un colocataire si surprenant. Merci à tous les gens avec qui j'ai pu passer des week-ends ou des vacances loin du travail et de l'air parisien : Léopold, Julien, Léa, Pierre, Manu, Rodrigue, Marie et Guillaume. Enfin, merci à tous mes amis de l'ENSTA que je n'ai pas encore mentionnés pour être constamment si géniaux et que j'espère cotoyer encore bien longtemps : Aurore, Elise, Benoît, Rémi, Quentin, Guillaume, Florian, Elias, Juliette, Adrien, Pierre, Antoine et tous les autres...

Contents

List of Figures

Introduction

In the recent years, several links have been found between the optimal transport theory of Monge and Kantorovich, initially concerned with optimally rearranging one measure into another, and the theory of partial differential equations. The properties of the Wasserstein distance in optimal tranport and the structure of geodesics in the Wasserstein space appeared useful to design gradient flows techniques (or "minimizing movement schemes") to solve numerous PDEs. The first example of such techniques is the variational scheme introduced by Jordan, Kinderlehrer and Otto to solve the Fokker-Planck equation [15]. A detailed account on this field can be found in [2].
In this thesis we study a non linear evolution equation. This equation can be interpreted as an "infinitely fast/infinitely slow" diffusion equation and can be used to model the growth of sandpiles or the compression molding of plastic.
It appears that this equation is very much tied to the optimal transport problem associated to the 1-Wasserstein distance. These profound links are exhibited in [7]. Consequently, a minimizing movement scheme using the 1-Wasserstein distance has been introduced in [1] and it was proven that the solution of this scheme converges to a weak solution of the equation.
This work with Guillaume Carlier and Jean-David Benamou focuses on the numerical implementation of this scheme. The first chapter introduces the equation and the minimizing movement scheme. The second chapter presents the numerical implementation using two primal-dual algorithm : an Augmented Lagrangian technique with the ADMM algorithm and the Chambolle-Pock algorithm. Chapter 3 presents some numerical simulations obtained for the sandpile growth model and Chapter 4 studies the compression molding model.

Chapter 1

A nonlinear evolution problem

1.1 The PDE

Consider a bounded domain Ω of $\mathbb{R}^d$ and a convex function F . In the following we are concerned with solving the following PDE :

$$\partial_t u - \text{div}\,(a\nabla F'(u)) = f, (t,x) \in (0,T)\times\Omega, u_{|t=0} = u_0 \tag{1.1}$$

with the constraints

$$a \geq 0 \tag{1.2a}$$
$$|\nabla F'(u)| \leq 1 \tag{1.2b}$$
$$a(1 - |\nabla F'(u)|) = 0 \tag{1.2c}$$
$$a\nabla F'(u)\cdot\vec{n} = 0, texton\partial\Omega \tag{1.2d}$$

It is useful to mention that a is actually an unknown function and can actually be viewed as a Lagrange multiplier of constraint (1.2b). For this reason, a actually depends on u and the equation (1.1) is nonlinear.
As shown in [4], this equation can be seen as the limit, for $p \to \infty$ of the diffusion equation

$$\partial_t u - \Delta_p u = f$$

where Δ_p denotes the p-Laplacian. This allows to interpret equation (1.1) as an "infinitely fast/infinitely slow" diffusion process. In the zones where constraint (1.2b) is satisfied, no diffusion occurs but wherever (1.2b) is violated, diffusion occurs at an infinite speed to rearrange the system.
This equation can also be interpreted as the differential inclusion

$$f - \partial_t u \subset \partial I_\infty(F'(u)), \quad u_{|t=0} = u_0 \tag{1.3}$$

where ∂I_∞ is the subdifferential of the indicator function of the set $\{v \in L^2, |Dv| \leq 1\ a.e.\}$. In the case $F'(u) = u$ this equation was studied by Aronsson, Evans and Wu ([4]) and Prigozhin ([18],[17],[16]) who interpreted it as a (simple) model for the height of growing sandpiles. Equation (1.1) also arises as a model for compression molding (see [3]) or river networks (see [17], [11]).
Previous attempts to solve numerically this equation has been successful. These are based on a time discretization of the formulation (1.3) solved by an Augmented Lagrangian method (see [8]). It was understood by that equation (1.1) has very strong connections with the theory of optimal transport. In particular, the constraints (1.2) are linked to the 1-Wasserstein distance and the notion of Kantorovich potential. In [1] a numerical scheme was introduced to solve this equation. This scheme is akin

to a Jordan-Kinderlehrer-Otto scheme (see [15] but uses the 1-Wasserstein distance instead of the usual quadratic one. The following section based on [1] introduces this numerical scheme.

1.2 1-Wasserstein minimizing movement scheme

In this section we denote by Lip the set of Lipschitz function of Ω and by Lip_1 the set of 1-Lipschitz function on Ω.
Given $g \in \mathrm{Lip}$ we say that g is balanced if $\langle g, 1\rangle = 0$. In this case we define W_1 the dual semi norm of g by

$$W_1(g) = \sup_{\theta \in \mathrm{Lip}_1} \langle g, \theta \rangle.$$

When $g = g^+ - g^-$ where g^+ and g^- are probability measures, W_1 is known as the 1-Wasserstein distance between g^+ and g^-. For a balanced g we define the set of Kantorovich potentials of g as

$$K(g) = \{\theta \in \mathrm{Lip}_1, W_1(g) = \langle g, \theta \rangle\}$$

Fenchel-Rockafellar duality yields the following dual formula for W_1.

$$W_1(g) = \inf_{\sigma \in (L^\infty(\Omega, \mathbb{R}^d))'} \left\{ \|\sigma\|_{(L^\infty)'}, -\mathrm{div}(\sigma) = g \right\}. \tag{1.4}$$

A solution to (1.4) is called an optimal flow. An optimal flow σ and a Kantorovich potential θ are linked by the following extremality relation.

$$\langle \sigma, \nabla\theta \rangle = \|\sigma\|_{(L^\infty)'} \text{ with } \|\nabla\theta\|_{L^\infty} \leq 1.$$

This expresses that σ is supported on the set where $|\nabla\theta|$ is equal to 1. If σ is L^1 this implies that $\sigma = a\nabla\theta$ with $a \geq 0$ satisfying the constraint (1.2c).
Using the previous definitions it is possible to rewrite the PDE (1.1) with the constraints (1.2) as the inclusion problem

$$-F'(u) \in K(\partial_t u - f). \tag{1.5}$$

Indeed this implies that there exists an optimal flow σ and a function a such that $\sigma = a\nabla F'(u)$ which satisfy

$$-\mathrm{div}(a\nabla F'(u)) = \partial_t u - f$$

and the constraint (1.2).
It is interesting to note that in the sandpile model studied in [4], the PDE (1.1) was also interpreted as an inclusion problem but in a "reverse" order compared to (1.5). Indeed they proved that the solution of the PDE satisfied the inclusion

$$f - \partial u_t \in I_\infty(u)$$

where I_∞ is defined by

$$I_\infty(u) \begin{cases} 0 & \text{if } u \in L^2(\mathbb{R}^d), |\nabla u| \leq 1 \text{ a.e.}, \\ +\infty & \text{otherwise.} \end{cases}$$

It is natural to define the following time discretization for a time step $\tau > 0$.

$$-F'(u_{k+1}^\tau) \in K\left(u_{k+1}^\tau - u_k^\tau - \int_{k\tau}^{(k+1)\tau} f\right). \tag{1.6}$$

Inclusion (1.6) can actually be seen as the optimality conditions for the variational Euler implicite scheme defined by

$$u_{k+1}^\tau \in \arg\min_u \left\{ W_1(u - u_k^\tau - f_k^\tau) + \int_\Omega F(u)dx \right\} \tag{1.7}$$

where

$$f_k^\tau = \int_{k\tau}^{(k+1)\tau} f.$$

The scheme (1.7) is an exemple of minimizing movement scheme in the space of probabilities, in the spirit of the famous Jordan-Kinderlehrer-Otto scheme ([15]) but using the 1-Wasserstein distance instead of the 2-Wasserstein distance.
Using the sequence (u_k^τ) we can define two interpolating curves by

$$\begin{aligned} &u^\tau(t) = u_k^\tau + \frac{t - k\tau}{\tau}(u_{k+1}^\tau - u_k^\tau), \\ &\tilde{u}^\tau(t) = u_{k+1}^\tau, \\ &\text{for } t \in (k\tau, (k+1)\tau]. \end{aligned}$$

We state the convergence result obtained in [1] concerning this scheme

Theorem 1. *Assume $f \in L^1((0,T), C^{0,\alpha_0}(\bar{\Omega}))'$) for some $\alpha_0 \in [0,1)$. There exists a sequence of timesteps $\tau_n \to 0$ and $u \in L^\infty((0,T), C^0(\bar{\Omega}))$ such that*

$$u^{\tau_n} \to u \text{ in } L^p((0,T), C^0(\bar{\Omega})), \forall p \in [1, +\infty), \tag{1.8}$$

$$\tilde{u}^{\tau_n} \to u \text{ in } L^1((0,T), C^0(\bar{\Omega})), \tag{1.9}$$

and u is a weak solution of (1.1).

The following work is concerned with the numerical implementation of the discrete scheme (1.7).

Chapter 2

Numerical Resolution

2.1 Deriving a simpler optimization problem

Following the previous chapter we wish to compute the sequence (u_k^τ) using the following scheme :

$$u_{k+1}^\tau \in \arg\min_u W_1\left(u - \left(u_k^\tau + \int_{k\tau}^{(k+1)\tau} f\right)\right) + \int_\Omega F(u). \tag{2.1}$$

Denoting $u_k^s = u_k^\tau + \int_{k\tau}^{(k+1)\tau} f$ and using the definition of W_1 we are interested in the problem

$$\min_u \sup_{\phi \in Lip_1} \int_\Omega \phi(u - u_k^s) + \int_\Omega F(u).$$

We start by swapping the infimum and the supremum in the previous expression.

Proposition 1. *Assume F is lower semi continuous and convex. We have*

$$\min_u \sup_{\phi \in Lip_1} \int_\Omega \phi(u - u_k^s) + \int_\Omega F(u) = \sup_{\phi \in Lip_1} \min_u \int_\Omega \phi(u - u_k^s) + \int_\Omega F(u). \tag{2.2}$$

Proof. The set Lip_1 is convex and compact. The set $\{u, \langle u - u_k^s \rangle = 0\}$ is obviously convex. Moreover for all ϕ the function $u \mapsto F(u) + \phi u$ is convex and l.s.c. so the function $u \mapsto \int_\Omega \phi u + \int_\Omega F(u)$. is also convex and l.s.c. Finally, for all u the function $\phi \to \int_\Omega \phi(u - u_k^s)$ is continuous and linear so it is concave. Under these conditions, Sion's minimax theorem ensures that we can swap the inf and the sup. □

Optimality conditions in u in the right hand side of (2.2) lead to $\phi = -F'(u)$ and so, assuming that F' is invertible and denoting by G its inverse, we are left with the problem

$$\sup_{\phi \in Lip_1} \int_\Omega \phi G(-\phi) - \phi u_k^s + F(G(-\phi)$$

which we reformulate as

$$\inf \int_\Omega \phi u_k^s - \phi G(-\phi) - F(G(-\phi)) + \chi_B(\nabla\phi) \tag{2.3}$$

where $B = \bar{B}_\infty(0,1)$ is the closed unit ball in the L^∞ space.
This is a problem of the form

$$\min J(\phi) + H(L\phi) \tag{2.4}$$

where L is a linear operator and J and H are convex possibly non smooth functions. In the following we assume that J is differentiable.
To solve it numerically we will apply two different algorithms particularly suited for this kind of non-smooth optimization problems : the Alternative Direction Method of Multipliers (ADMM) and the Chambolle-Pock algorithm.

2.2 Alternative Direction Method of Multipliers (ADMM)

2.2.1 Algorithm

Inspired by [5] we use an augmented lagrangian method to solve problem (2.3). Indeed we can write (2.4) as

$$\inf_{\phi,q} \sup_{\sigma} J(\phi) + H(q) + \langle \mu, \nabla\phi - q\rangle + \frac{\lambda}{2}\|\nabla\phi - q\|^2 \tag{2.5}$$

Each step ADMM consists in

- Minimizing in ϕ with other variables fixed.
- Minimizing in q with other variable fixed.
- Updating μ

Algorithm 1 ADMM algorithm

Input: $\phi^0, q^0 = \nabla\phi^0, \mu^0, \lambda$
  while Optimality conditions are not verified **do**
    $\phi^{k+1} = \arg\min_\phi J(\phi) + \langle \mu^k, \nabla\phi - q^k\rangle + \frac{\lambda}{2}\|\nabla\phi - q^k\|^2$
    $q^{k+1} = \arg\min_q H(q) + \langle \mu^k, \nabla\phi^{k+1} - q\rangle + \frac{\lambda}{2}\|\nabla\phi^{k+1} - q\|^2$
    $\mu^{k+1} = \mu^k + \lambda(\nabla\phi^{k+1} - q^{k+1})$
  end while
Output: $u^{k+1} = G(-\phi^{k+1})$

Let us detail the first two steps of each iteration.

- **Step 1 : Minimizing in ϕ**
  Computing the first variation with respect to ϕ of the lagrangian (2.5) yields the following variational formulation defining ϕ^{k+1}.

$$\int_\Omega \nabla\phi^{k+1}\nabla\psi + J'(\phi^{k+1})\psi + \mu^k\nabla\psi - q^k\nabla\psi = 0 \tag{2.6}$$

  for all $\psi \in H^1(\Omega)$.

  This is equivalent to the Laplacian equation

$$-\Delta\phi^{k+1} + J'(\phi^{k+1}) = u_k^s + \mathrm{div}(\mu^k - q^k)$$

  with Neumann boundary condition.

- **Step 2 : Minimizing in q**
  Optimality conditions in q yield

$$\mu^k + \lambda \nabla \phi^{k+1} - q^{k+1} \in \partial H(q^{k+1}).$$

2.2.2 Optimality conditions

The first order optimality conditions associated to the problem (2.5) are

$$\begin{aligned} &\nabla \phi - q = 0, \\ &-\Delta \phi + J'(\phi) = \operatorname{div}(\mu - q), \\ &\mu + \lambda \nabla \phi - q \in \partial H(q). \end{aligned}$$

By virtue of the algorithm, the last step is satisfied at the end of each iteration.

2.2.3 Practical implementation

In this work, the ADMM algorithm was implemented in FreeFem++ [13]. Indeed it was well suited to solving the Laplacian equation of step 1 at each iteration.
In all of the examples presented, we restricted ourselves to the domain $\Omega = [0,1]^2$. It was discretized using a structured triangular mesh generated by FreeFem. Variables (ϕ^k) were chosen to belong to the family P2 of piecewise quadratic continuous finite elements and variables (q^k), (μ^k) were chosen to belong to the family P1 of piecewise linear continuous finite elements.

2.3 Chambolle-Pock algorithm

2.3.1 Algorithm

The Chambolle-Pock algorithm was introduced in [10] to solve problems of the form (2.4).
We start by rewriting (2.4) as a inf-sup problem using the definition of the conjugate function :

$$\inf_{\phi} \sup_{q} J(\phi) + \langle \nabla \phi, q \rangle - H^\star(q) \tag{2.7}$$

Algorithm 2 Chambolle-Pock algorithm

Input: $\phi^0, q^0 = \nabla \phi^0, \psi^0, \tau, \sigma > 0, \theta \in [0,1]$
  while Optimality conditions are not verified **do**
    $q^{k+1} = \operatorname{prox}_{\sigma H^\star}(q^k + \sigma \nabla \psi^k)$
    $\phi^{k+1} = \operatorname{prox}_{\tau J}(\phi^k + \tau \operatorname{div} q^{k+1})$
    $\psi k + 1 = \phi^k + \theta(\phi^{k+1} - \phi^k)$
  end while
Output: $u^{k+1} = G(-\phi^{k+1})$

Unlike the augmented lagrangian method which required to solve a PDE at each iteration, the Chambolle-Pock algorithm only requires to compute the proximal operator

of the two functionals J and $H^\star$. Note that thanks to the Moreau identity, it is equally easy to compute the proximal operator of $H^\star$ and H :

$$\text{prox}_{\lambda H^\star}(x) = x - \text{prox}_{\lambda^{-1}H}(\lambda^{-1}x)$$

Provided that these operators are able to be computed quickly, it is then expected that the computation time per iteration of the Chambolle-Pock algorithm will be much lower and that it will scale way better with the size of the grid than the other method which require to solve a PDE at each iteration.
This behaviour has already been shown in [14] where the Chambolle-Pock method was in an optimal transport setting to compute the W_2 distance between two densities using the Benamou-Brenier [5] formulation.

2.3.2 Optimality conditions

The first order optimality conditions associated to the problem (2.7) are

$$\begin{aligned} &J'(\phi) - \text{div} q = 0 \\ &\nabla\phi \in \partial H^\star(q) \end{aligned}$$

2.3.3 Practical implementation

The Chambolle-Pock algorithm was implemented in Matlab. Indeed as we will see in the following chapters, in the two cases we studied, the proximal algorithm of the functionals took the form of straightforward vectorial operations.
In all our tests we set $\theta = 1$, a case for which the convergence analysis is presented in [10]. In that case, for the algorithm to converge, the parameters τ and σ are required to satisfy the condition

$$\tau\sigma\|\nabla\|^2 < 1. \tag{2.8}$$

In all of the examples presented, we restricted ourselves to the domain $\Omega = [0,1]^2$. It was discretized using a uniform grid of $N \times N$ points and, following [9] we define a discrete gradient operator, for i, j such that $1 \leq i, j \leq N$, by

$$(\nabla f)_{i,j} = ((\nabla f)^1_{i,j}, (\nabla f)^2_{i,j})$$

with

$$(\nabla f)^1_{i,j} = \begin{cases} \frac{1}{h}(f_{i,j+1} - f_{i,j}) & \text{if } i < N, \\ 0 & \text{if } i = N, \end{cases}$$
$$(\nabla f)^2_{i,j} = \begin{cases} \frac{1}{h}(f_{i+1,j} - f_{i,j}) & \text{if } j < N, \\ 0 & \text{if } j = N. \end{cases}$$

Accordingly we set, for $g = (g^1, g^2)$,

$$(\mathrm{div} g)_{i,j} = \begin{cases} \frac{1}{h}(g^2_{i-1,j} - g^2_{i,j}) & \text{if } 1 < i < N, \\ -\frac{1}{h} g^2_{i,j} & \text{if } i = 1, \\ \frac{1}{h} g^2_{i-1,j} & \text{if } i = N, \end{cases} + \begin{cases} \frac{1}{h}(g^1_{i,j-1} - g^1_{i,j}) & \text{if } 1 < j < N, \\ -\frac{1}{h} g^1_{i,j} & \text{if } j = 1, \\ \frac{1}{h} g^1_{i,j-1} & \text{if } j = N, \end{cases}$$

so that we have, for all $f \in \mathbb{R}^{N\times N}$ and all $g \in \mathbb{R}^{N\times N} \times \mathbb{R}^{N\times N}$, $\langle \nabla f, g\rangle = \langle f, -\mathrm{div} g\rangle$, or, in other word, $\nabla^\star = -\mathrm{div}$.
From [9] we have $\|\nabla\|^2 \leq \frac{8}{h}$. Thus we set $\sigma = \tau = \frac{h}{3}$ so as to verify (2.8).

Chapter 3

Sandpile Growth

3.1 Sandpile Growth Model

We consider the previous PDE in the so called "linear case" when $F'(u) = u$.

$$\partial_t u - \text{div}\,(a\nabla u) = f, (t,x) \in (0,T) \times \Omega, u_{|t=0} = u_0 \tag{3.1}$$

with the constraints

$$a \geq 0 \tag{3.2a}$$
$$|\nabla u| \leq 1 \tag{3.2b}$$
$$a(1 - |\nabla u|) = 0 \tag{3.2c}$$
$$a\nabla u \cdot \vec{n} = 0, \text{ on } \partial\Omega \tag{3.2d}$$

This equation was studied by Prigozhin ([18]) and by Aronsson, Evans and Wu ([4] who interpreted it as a (simple) model for the height of growing sandpiles.
The idea behind the model is the following. Sand drops onto the domain Ω at a rate represented by a source function f. As long as the slope of the sandpile is "small", meaning that it doesn't exceed a certain value (in this case 1), the sandpile remains motionless. As soon as the slope becomes too high, the excess sand is immediately transported to the bottom of the slope.
We start with a function u_0 describing the initial shape of the sandpile. We assume that u_0 is 1-Lipschitz so that the sandpile is initially stable. Following the previous chapters we wish to compute the sequence (u_k^τ) using the following scheme :

$$u_{k+1}^\tau \in \arg\min_u W_1\left(u - \left(u_k^\tau + \int_{k\tau}^{(k+1)\tau} f\right)\right) + \frac{1}{2}\int_\Omega u^2. \tag{3.3}$$

Denote $u_k^s = u_k^\tau + \int_{k\tau}^{(k+1)\tau} f$.
At each timestep we study a problem of the form (2.4) where the functionals J and H are defined by

$$J(\phi) = \frac{1}{2}|\phi + u_k^s|^2, \tag{3.4}$$
$$H(q) = \chi_B(q), \tag{3.5}$$

where χ_B is the characteristic function of the set $B = \bar{B}_{\|\cdot\|_\infty}(0,1)$, the closed unit ball for the L^∞ distance. Bear in mind that, numerically, the variables are discretized on a structured mesh of $N \times N$ points and thus $q \in \mathbb{R}^{N\times N\times d}$. Thus B is actually the closed unit ball of the space $\mathbb{R}^{N\times N\times d}$ endowed with the norm

$$\|q\|_{\infty,2} = \max_{1\leq i,j\leq N} \|q_{i,j}\|_2.$$

The specific form of the algorithms are presented in the next sections.

3.2 Alternative Direction Method of Multipliers (ADMM)

In the sandpile cases, the first two steps of the ADMM algorithm take the following form

- **Step 1 : Minimizing in ϕ**
  We have $J'(\phi) = \phi + u_k^s$. Thus according to (2.6) the first step consists in solving the variational formulation

  $$\int_\Omega \nabla\phi^{k+1}\nabla\psi + \phi^{k+1}\psi + \sigma^k\nabla\psi - q^k\nabla\psi + u_k^s\psi = 0$$

  for all $\psi \in H^1(\Omega)$.

  This is equivalent to the laplacian equation

  $$-\Delta\phi^{k+1} + \phi^{k+1} = u_s^k + div(\sigma^k - q^k)$$

  with Neumann boundary condition.

- **Step 2 : Minimizing in q**
  The second step consists in finding q^{k+1} satisfying

  $$\sigma^k + \lambda\nabla\phi^{k+1} - q^{k+1} \in \partial\chi_B(q^{k+1}).$$

  This simply amounts to computing the pointwise projection of the vector $\nabla\phi^{k+1} + \frac{1}{\lambda}\sigma^k$ on the unit ball in $\mathbb{R}^d$ with respect to the Euclidean norm

  $$q_i^{k+1} = \text{Proj}_{B_{\|\cdot\|_2}(0,1)}((\nabla\phi^{k+1} + \frac{1}{\lambda}\sigma^k)_i)$$

3.3 Chambolle-Pock algorithm

The Chambolle-Pock algorithm requires to compute the proximal operator of the functionals J and H defined in (3.4).
First, we have

$$\text{prox}_{\tau J}(\psi) = \arg\min_\phi \frac{\tau}{2}\|\psi - \phi\|^2 + \|\phi + u_k^s\|^2 = \frac{1}{1-\tau}(\psi + \tau u_k^s).$$

Then, using the fact that the conjugate function of χ_B is $\|\cdot\|_{1,2}$ defined by

$$\|q\|_{1,2} = \sum_{1\leq i,j\leq N} \|q_{i,j}\|_2.$$

It is well known that we have

$$(\text{prox}_{\sigma H^\star}(p))_i = (\text{prox}_{\sigma G}(p))_i = \frac{p_i}{|p_i|}\max(0, |p_i| - \sigma).$$

Thus the first two steps of the algorithm are

- **Step 1 : Minimizing in q**
  Define q^{k+1} as

$$\begin{aligned} p &= q^k + \sigma\nabla\psi^k, \\ q_i^{k+1} &= (\text{prox}_{\sigma G}(p))_i = \frac{p_i}{|p_i|}\max(0, |p_i| - \sigma). \end{aligned}$$

- **Step 2 : Minimizing in ϕ**
  Define ϕ^{k+1} as

$$\phi^{k+1} = \text{prox}_{\tau F}(\phi^k + \tau\text{div } q^{k+1}) = \frac{1}{1-\tau}(\phi^k + \tau\text{div } q^{k+1} + \tau u_s).$$

3.4 Extensions

3.4.1 Working with different slopes

The constraint (3.2b) can be modified to take the more general form

$$|\nabla u(x)| \leq \gamma(x)$$

for $\gamma \geq 0$.
This allows to model more general granular than sand which have different angle of repose. For example the limit $\gamma \to 0$ can be used as a rough model for the behaviour of lake and rivers over a non-flat landscape (see [17], [11]). Again denote by $u(x,t)$ the height of the sandpile at position x and at time t. The function u must satisfy the equation (3.1) with the constraints

$$a \geq 0 \tag{3.6a}$$
$$|\nabla u(x)| \leq \gamma(x), \ a.e. \tag{3.6b}$$
$$a(x)(\gamma(x) - |\nabla u(x)|) = 0, \ a.e. \tag{3.6c}$$
$$a\nabla u \cdot \vec{n} = 0 \tag{3.6d}$$

It suffices to use the change of variables $\gamma v = u$ and to notice that v solves (3.1) associated to the source term γf with the constraints (3.2).

3.4.2 Advection

The equation (1.1) can be modified by adding an advection term that can be used to model wind blowing over the sandpile.
Given an advection field $\mathbf{ff}$, the equation becomes

$$\partial_t u + \mathbf{ff} \cdot \nabla u - \text{div}\,(a\nabla F'(u)) = f, (t,x) \in (0,T) \times \Omega, u_{|t=0} = u_0.$$

To solve this problem we used a simple explicit method. Namely we changed the previous term in the scheme (1.7) from $u_k^\tau + \int_{k\tau}^{(k+1)\tau} f$ to $u_k^\tau + \int_{k\tau}^{(k+1)\tau} f + \mathbf{ff} \cdot \nabla u_k^\tau$.

Another method would have been to use the operator splitting algorithm described in [8] which consists in solving successfully the equations

$$\partial_t u + \mathbf{ff} \cdot \nabla u = 0$$

and equation (1.1) with constraints (1.2).

3.4.3 An optimal pit problem

We consider a problem of optimal open pit mining inspired by [12].
Assume that the position of underground ore is known. One wants to dig an open pit to reach the ore by excavating the nonprofitable soil above it. Of course, the pit cannot be too steep otherwise it will collapse on itself. In that sense, the pit must satisfy a condition similar to (3.2b). To see the connection more clearly one can see the problem in a "reversed" way. Imagine that one wants to reach floating ores in the sky by building sandpiles.
This problem can be cast as an optimal control problem. The state variable denoted by u models the height of the sandpile. The control variable denoted by f denotes the rate at which one adds sand to the system. The problem is then

$$\inf_{u,f} J(u,f) \tag{3.7}$$

$$\partial_t u + \operatorname{div}(a\nabla u) = f \tag{3.8}$$

$$a \geq 0, |\nabla F'(u)| \leq 1, a(1 - |\nabla F'(u)|) = 0. \tag{3.9}$$

The cost function expresses the fact that dropping sand (or digging soil) is costly but reaching ore is profitable.
Using the minimizing movement scheme one cas consider a discretized version of the problem (3.7).

$$\inf_{(u^n),(f^n)} \tilde{J}(u^1 ... u^n, f^1, ..., f^n)$$

$$u^{n+1} = -\arg\min_{\phi 1-Lip} \int_\Omega |\phi + u_s^k|^2$$

3.5 Numerical results

In the following we present some numerical results. The Augmented Lagrangian algorithm was implemented in FreeFem++ [13], making use of its efficiency to solve linear PDEs. The domain $\Omega = [0,1]^2$ was discretized using a structured triangular mesh. The Chambolle-Pock algorithm, only necessiting simple vectorial operations, was implemented in Matlab. The domain $\Omega = [0,1]^2$ was discretized using a uniform grid.

Flat terrain We start by considering the case when the source term is a sum of pointwise sources $\sum_i f_i \delta_{x_i}(x)$. In this case, a simple theoretical solution exists and has been described in [4]. It takes the form of multiple interacting cones of slope 1.
For numerical purpose, a pointwise source is approximated by a Gaussian function

of the form

$$\mathcal{G}(x, y; x_0, y_0, I, \sigma) = I \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma}\right).$$

The first example shows the solution obtained using three pointwise sources using the following parameters.

x_0	y_0	I	σ
0.2	0.1	500	0.0001
0.8	0.6	100	0.0001
0.3	0.7	50	0.0001

The number of points on the mesh and the grid has been set to 2^6, the timestep is $\tau = 0.05$ and the tolerance for both algorithm is set to 10^{-6}. Since they are the most costly part of an iteration, the optimality conditions for Chambolle-Pock are only computed once every 20 iterations. For the augmented lagrangian, the parameter λ is set to 0.05. As discussed previously, for the Chambolle-Pock method, the parameters τ and σ are set to $2^{-6}/3$ and $\theta = 1$. Unless stated otherwise, the same set of parameters are used in all the other examples.
Results are visible on Figure 3.2 for Augmented Lagrangian and on Figure 3.3 for Chambolle-Pock. In both cases the solution takes the expected form of interacting cones. Because of the difference in the respective intensity of the sources, one cone almost swallows another at the last timestep. The comparison with the theoretical solutions and a quantitative study of the error are given in the next section. We give the number of iterations required by both methods to reach convergence at some timesteps in Table 3.1. On total the Augmented Lagrangian method took 6633 seconds to complete while Chambolle-Pock took 33 seconds.

Timestep	0.05	0.25	0.5	0.75	1
A.L.	2069	1172	1060	763	768
C-P	2240	2060	3100	3600	3640

TABLE 3.1: Number of iterations for example 1

The second example showcases the results using a cross-shaped source. Results at different timesteps are visible on Figure 3.4 for Augmented Lagrangian and Figure 3.5 for Chambolle-Pock. Notice how the sandpile keeps the shape of a cross at the beginning before progressively taking the shape of a cone to satisfy the slope condition. We give the number of iterations required by both methods to reach convergence at some timesteps in Table 3.2. On total the Augmented Lagrangian method took 14712 seconds to complete while Chambolle-Pock took 30 seconds.

Timestep	0.05	0.25	0.5	0.75	1
A.L.	120	1813	2803	2884	3574
C-P	2100	1880	1800	1780	1740

TABLE 3.2: Number of iterations for example 2

Sandpiles on non-flat terrain The initial surface u_0 can also be chosen to be non-constant. On Figure 3.7 it was set to $u_0(x, y) = 0.2(x + y)$ (Note that is satisfies the

slope constraint). The source term is composed of three pointwise sources approximated by gaussian sources using the following parameters

x_0	y_0	I	σ
0.2	0.1	40	0.0001
0.8	0.6	30	0.0001
0.3	0.7	20	0.0001

Results are visible on Figure 3.6 for Augmented Lagrangian and Figure 3.7 for Chambolle-Pock. The influence of gravity is particularly visible as "skewed" cones are formed with a wider base in the direction of descent of the slope. We give the number of iterations required by both methods to reach convergence at some timesteps in Table 3.3. On total the Augmented Lagrangian method took 6280 seconds to complete while Chambolle-Pock took 30 seconds.

Timestep	0.05	0.25	0.5	0.75	1
A.L.	2154	1196	964	854	923
C-P	2480	1860	1800	1760	1720

TABLE 3.3: Number of iterations for example 3

Moving sources It is also possible to study moving sources. We study the case of a pointwise source moving in a circle around the center of the domain and with varying rate.

$$f(x,y,t) = 20(\cos(5t)+1)\exp\left(-\frac{(x-0.5-0.2\cos(3t))^2+(y-0.5-0.2\sin(3t))^2}{2\times 0.0001}\right)$$

Results are visible on Figure 3.8 for Chambolle-Pock. We give the number of iterations required the algorithm to reach convergence at some timesteps in Table 3.4. On total the Chambolle-Pock algorithm took 147 seconds to complete

Timestep	0.05	1	2	3	4	5
C-P	2020	1940	1840	1720	1860	1880

TABLE 3.4: Number of iterations for example 4

Negative sources and Hoodoos Negative sources can be used to model erosion the formation of hoodoos (also called fairy chimneys). Hoodoos are tall towers of sedimentary rock topped by a harder, less easily eroded stone. This stone protects the lower rock from the rain falling and thus, from erosion.

FIGURE 3.1: Hoodoos in Cappadocia, Turkey (from Wikipedia)

Let us consider the domain $\Omega \in \mathbb{R}^2$ filled with granular material of angle of repose $\arctan(\gamma)$. Suppose that a more resistant stone occupies the space upon the domain $\Gamma \in \Omega$. Rain falls upon the domain and erosion occurs everywhere in Ω except in Γ. We model this by a negative source function f meaning that

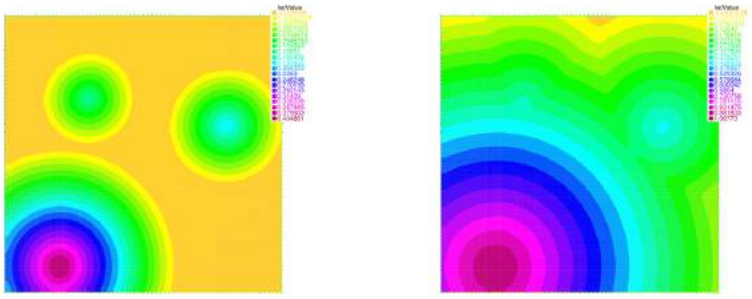
$$\begin{cases} f \leq 0 & \text{in } \Omega \backslash \Gamma, \\ f = 0 & \text{in } \Gamma. \end{cases}$$

Due to this the level of the soil will progressively decrease and in order to satisfy the slope constraint, the stone under the rock will start to erode on the top, creating the characteristic pillar shape of a hoodoo. The numerical results we obtained with Chambolle-pock using a uniform f may be seen on Figure 3.9 where the black cuboid is meant to represent the solid rock.

Advection The results of adding an advection term to the model can be seen on Figure 3.10 where a single pointwise source is perturbed by the uniform advection field $\mathbf{ff} = (0.3, 0)^\top$. We give the number of iterations required by Chambolle-Pock to reach convergence at some timesteps in Table 3.5. The total simulation time was 151 seconds.

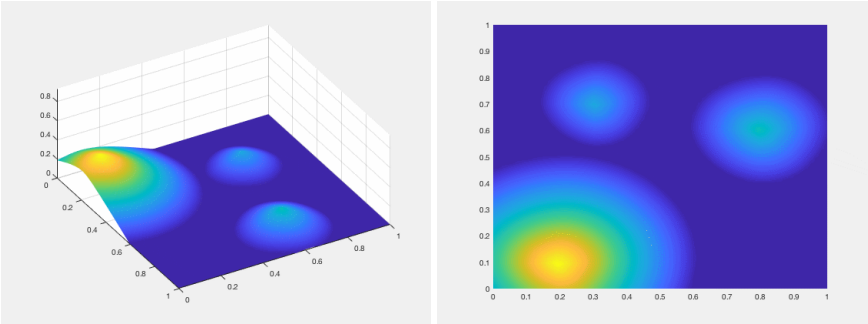
Timestep	0.05	1	2	3	4	5
C-P	2040	1820	1860	1860	1860	1860

TABLE 3.5: Number of iterations for example 4

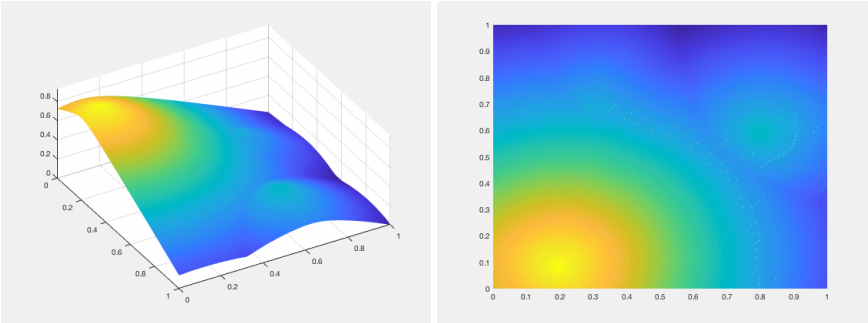


(A) t = 0.15 (B) t = 1

FIGURE 3.2: Evolution of a sandpile on a flat terrain with pointwise sources using Augmented Lagrangian.

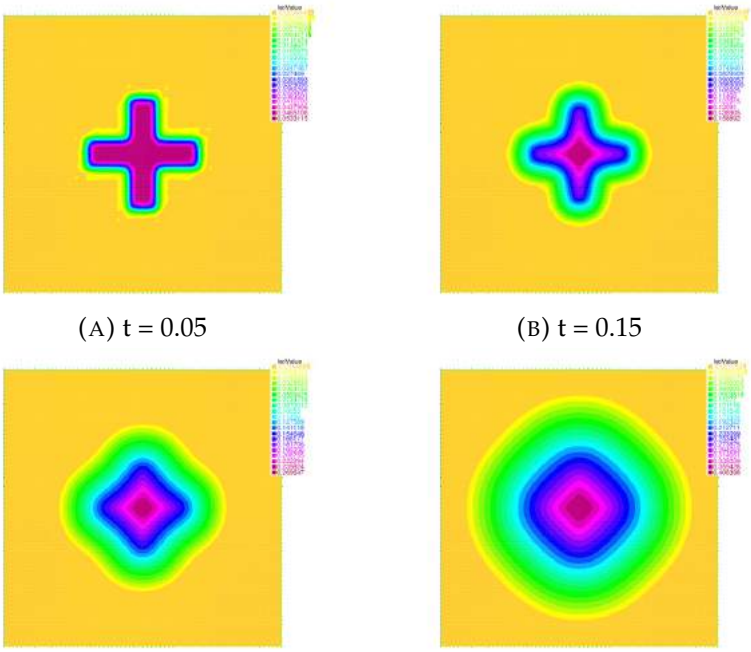


(A) t = 0.15 (B)



(C) t = 1 (D)

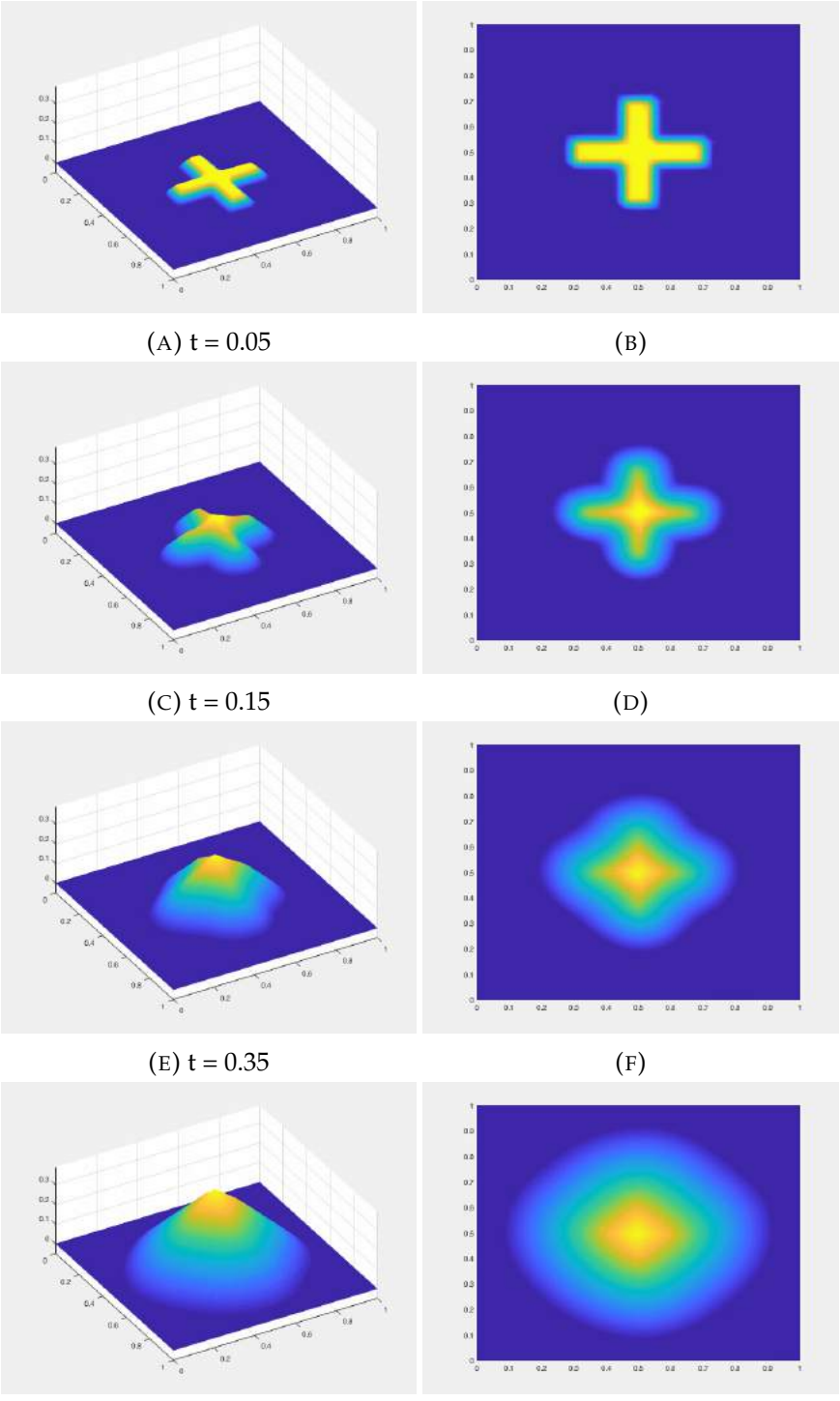
FIGURE 3.3: Evolution of a sandpile on a flat terrain with pointwise sources using Chambolle-Pock.



(A) t = 0.05 (B) t = 0.15

(C) t = 0.35 (D) t = 1

FIGURE 3.4: Evolution of a sandpile on a flat terrain with pointwise sources using Augmented Lagrangian.



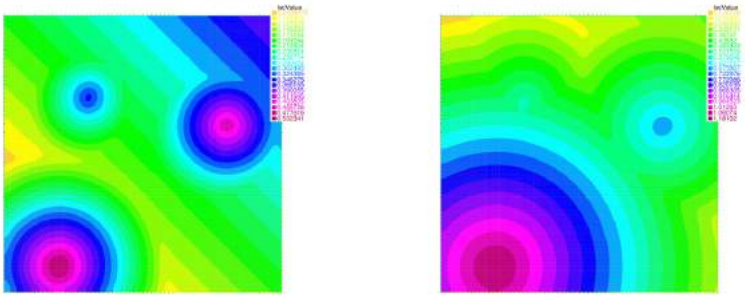
(A) t = 0.05 (B)

(C) t = 0.15 (D)

(E) t = 0.35 (F)

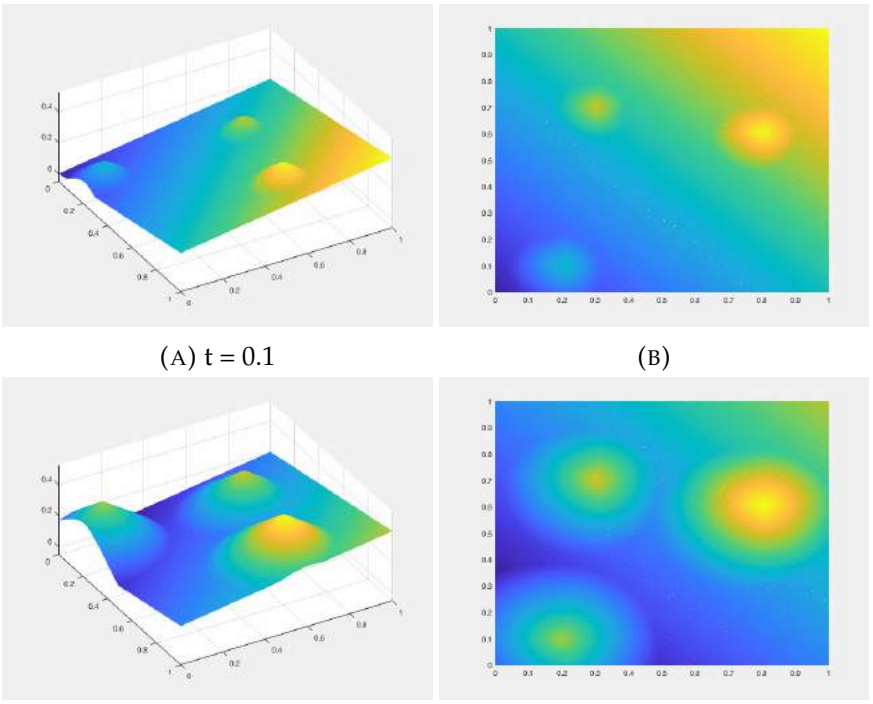
(G) t = 1 (H)

FIGURE 3.5: Evolution of a sandpile with a cross-shaped source using Chambolle-Pock



(A) t = 0.15 (B) t = 1

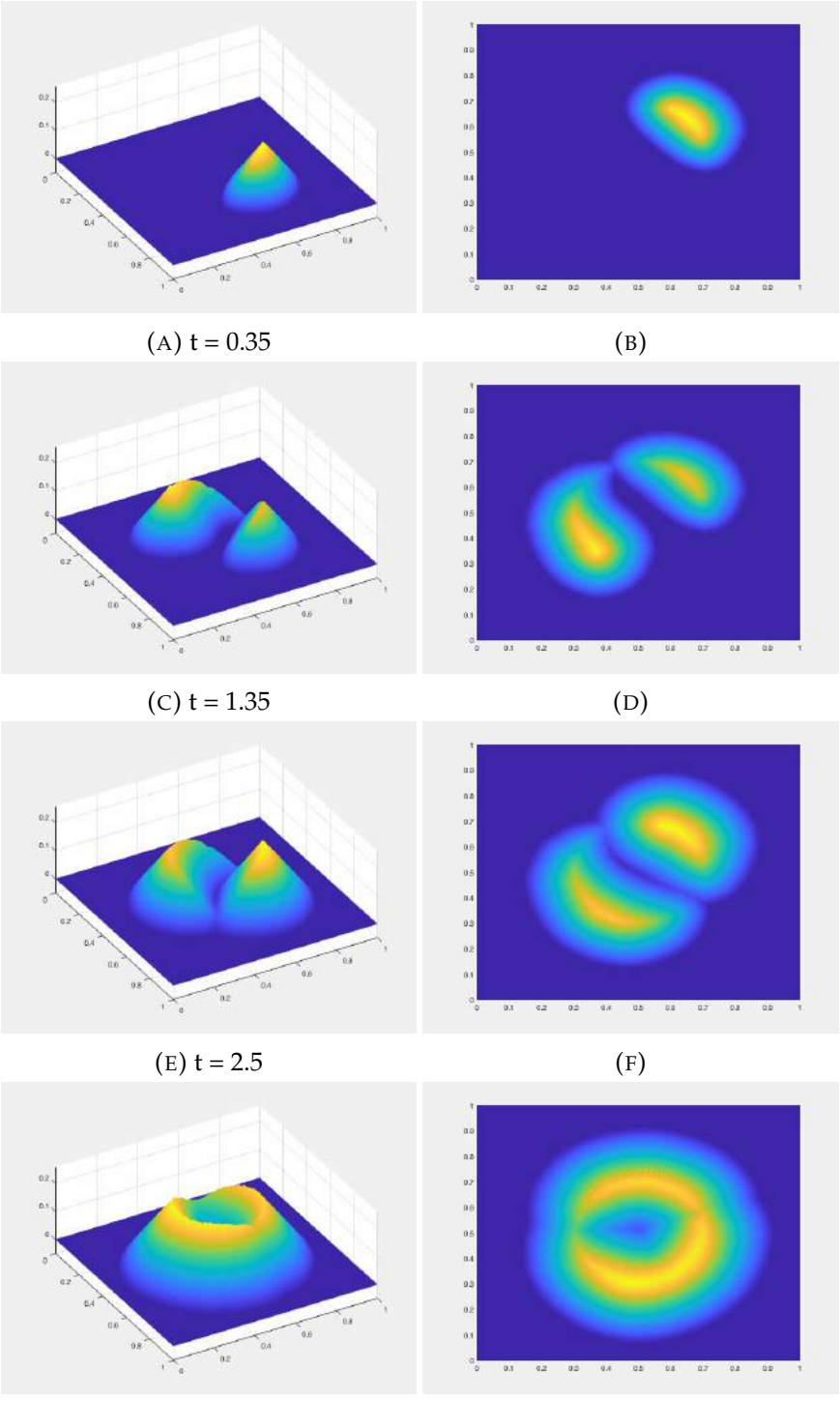
FIGURE 3.6: Evolution of a sandpile on a non-flat terrain with pointwise sources using Augmented Lagrangian.



(A) t = 0.1 (B)

(C) t = 1 (D)

FIGURE 3.7: Evolution of a sandpile on a non-flat terrain with pointwise sources using Chambolle-Pock



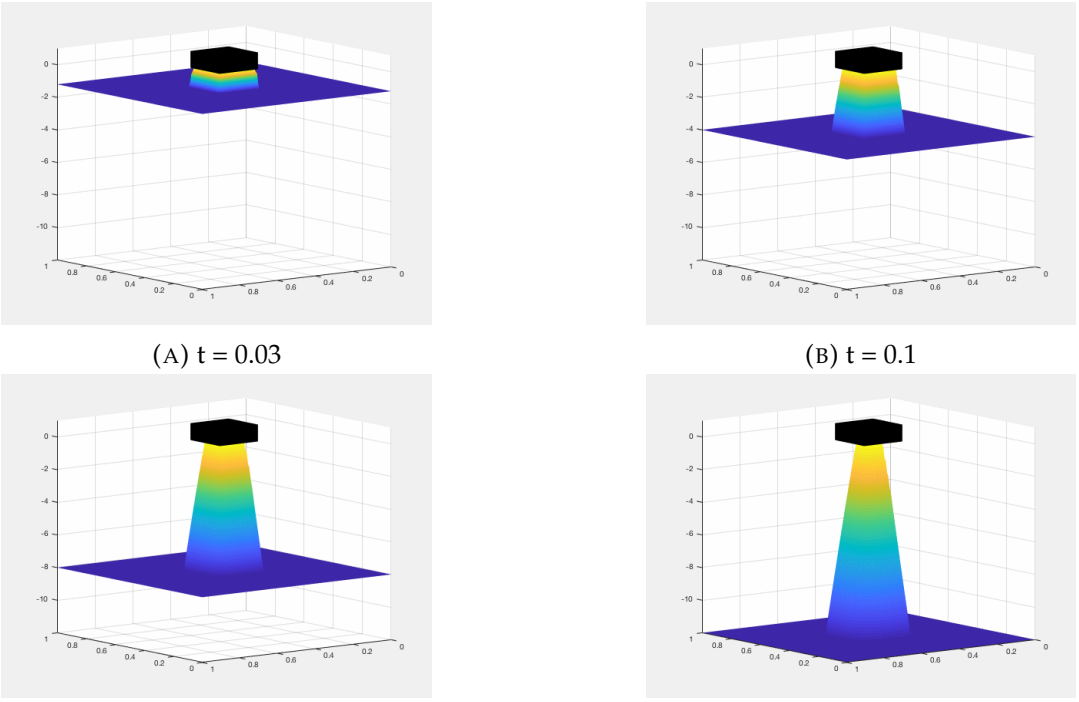
(A) t = 0.35 (B)

(C) t = 1.35 (D)

(E) t = 2.5 (F)

(G) t = 4.25 (H)

FIGURE 3.8: Evolution of a sandpile with a moving pointwise source of varying intensity. Results obtained with Chambolle-Pock



(A) t = 0.03 (B) t = 0.1

(C) t = 0.2 (D) t = 0.3

FIGURE 3.9: A hoodoo. Results obtained for $\gamma = 100$.

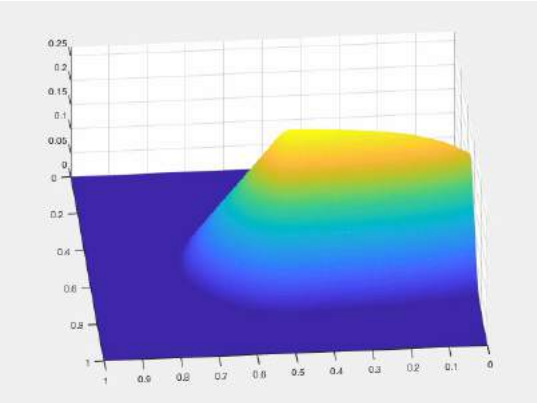


FIGURE 3.10: A single pointwise source under an advection field.

3.6 Comparison with the true solution

In the case where f is a pointwise source of the form

$$f(x,t) = \sum_{i=1}^{p} f_i(t)\delta_{x_i}(x),$$

it is known that the problem (3.1) admits a unique explicit solution in the form of p cones located at points x_p. More precisely the solution in this case is

$$u(x,t) = \max(u_0(x), z_1(t) - |x - x_1|, ..., z_p(t) - |x - x_p|) \tag{3.10}$$

where z_i is the height of the i-th cone, located at point x_i.
We use this explicit solution to evaluate the quality of our numerical results.
For this we choose to study the case $p = 1$ and a source term with constant intensity (i.e. $f_1(t) = I$) and we approximate the source term $f(x,t) = I\delta_{x_1}(x)$ by a gaussian function with small enough standard deviation. More explicitly we choose

$$f(x,t) = I \exp\left(-\frac{\|x - x_1\|^2}{2\sigma}\right).$$

Now denote by z^k the height of the sandpile at timestep $k\tau$. The total mass added to the system between instant 0 and $k\tau$ is equal to

$$m_k = \int_0^{k\tau} \int_{\mathbb{R}^d} I \exp\left(-\frac{\|x - x_1\|^2}{2\sigma}\right) dxdt = k\tau I(2\sigma\pi)^d.$$

We assume that even for our gaussian approximation the solution takes the form of a cone of slope 1. Denoting by $V^d(R)$ the volume of a d-sphere of radius R, the volume of the cone at time step $k\tau$ is defined by

$$V_k = \int_0^{z^k} V^d(z - z^k)dz = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} \int_0^{z^k} (z - z^k)^d dz = \frac{\pi^{\frac{d}{2}}}{(d+1)\Gamma(\frac{d}{2}+1)} (z^k)^{d+1}.$$

So in particular if $d = 1$ we get $V_k = (z^k)^2$ and if $d = 2$ we get $V_k = \frac{1}{3}\pi(z^k)^3$.
Now obviously we have the relation $m_k = V_k$ which leads to the height of the sandcone satisfying the relation

$$z^k = \left((d+1)\pi^{\frac{d}{2}}\Gamma(\frac{d}{2}+1)k\tau I(2\sigma)^d\right)^{\frac{1}{d+1}}.$$

We ran our algorithms with a single gaussian source located at the center of the domain (discretized using a grid with 2^{12} points) with parameters $I = 40, \sigma = 0.0001$. We set $\tau = 0.05$ and the tolerance parameter on optimality to 10^{-6}.
On Figure 3.11 we plot, for each timestep, the relative error in L^∞ norm $\frac{\|u^k - u^\star(\cdot, k\tau)\|_\infty}{z^k}$ where $u^\star$ is the explicit solution defined by (3.10) for the two algorithms after convergence was reached. We also plot the number of iterations needed to reach optimality at each timestep.
The ADMM algorithm exhibits no error propagation for the first timesteps contrary to the Chambolle-Pock algorithm. At a certain timestep, error propagation starts to appear for ADMM and from this point the error of both algorithm follows a similar pattern.

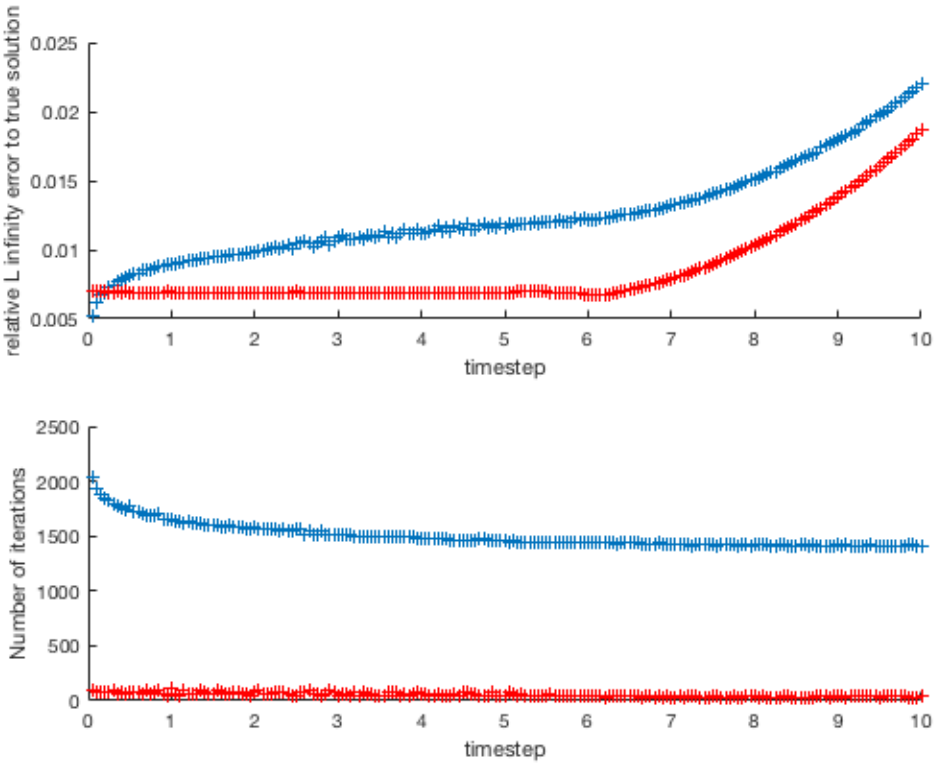


FIGURE 3.11: Top : relative L^∞ error between u_k and the true solution at each time step for ADMM (red) and Chambolle-Pock(blue). Bottom : number of iterations to reach convergence at each time step for ADMM (red) and Chambolle-Pock (blue).

Although it appears that ADMM converges in much less iterations than Chambolle-Pock, each iteration of ADMM is much longer than an iteration of CP.
It is important to bear in mind that the error comes from both the discretization in time and the discretization in space.
To evaluate this error caused by time discretization, on Figure 3.12 we plot the relative error in L^∞ norm $\frac{\|u^k - u^\star(\cdot, k\tau)\|_\infty}{z^k}$ for one run of the algorithm (one timestep) at the end of each iteration for both algorithms and for different number of discretization points. Results were obtained for $d = 2$ for a single gaussian source with parameters $I = 40, \sigma = 0.0001, \tau = 0.05$ and tol $= 10^{-6}$ and for the first timestep.

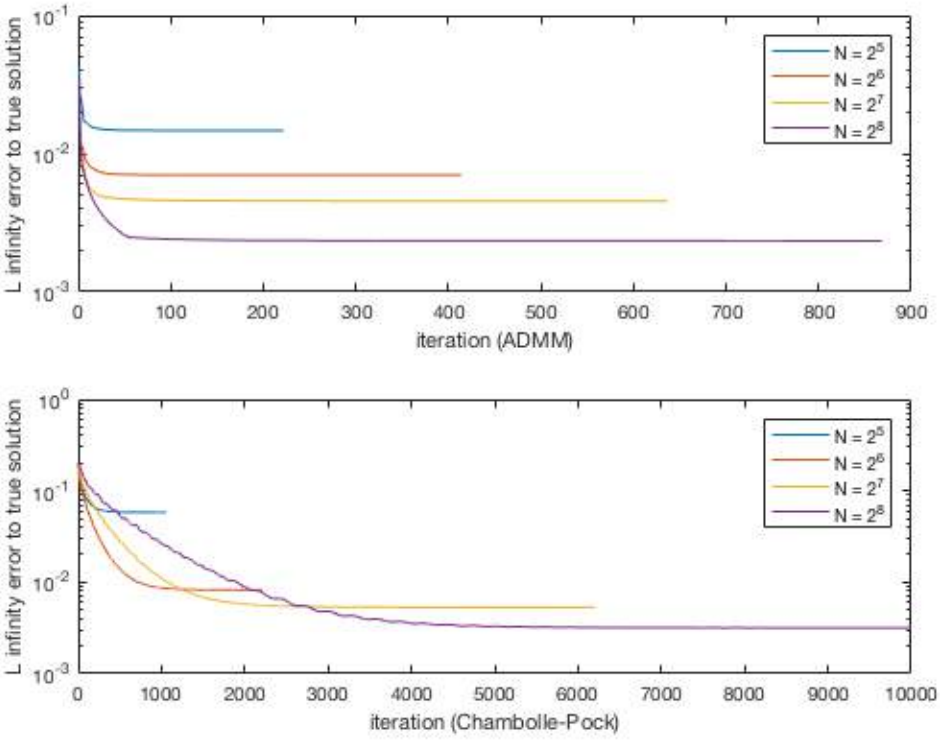


FIGURE 3.12: Relative L^∞ error between u_k and the true solution at each iteration. Top : ADMM, Bottom : Chambolle-Pock
Results were obtained for $d = 2$ for a single gaussian source with parameters $I = 40, \sigma = 0.0001$, $\tau = 0.05$ and tol $= 10^{-6}$.

Chapter 4

Compression Molding

4.1 The model

We know study the application of (1.1) to the problem of compression molding of plastic.
Consider a volume of plastic material lying between two horizontal plates separated by a distance h. As h goes to 0, we want to track the shape of the material.
Denote by U_t the projection of the material onto $\mathbb{R}^2$. Thus at time t, the plastic approximately fills the region $U_t \times (0, h)$. We denote by $w = \chi_{U_t}$ the indicator function of U_t and by $\Gamma_t = \partial U_t$ the boundary of U_t.
In [3] the authors proposed the following model for the evolution of w :

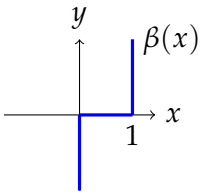
$$\partial_t w - \operatorname{div}(aDu) = w, \tag{4.1}$$
$$u \in \beta(w), \tag{4.2}$$

with constraints

$$a \geq 0 \tag{4.3a}$$
$$|\nabla u| \leq 1 \tag{4.3b}$$
$$a(1 - |\nabla u|) = 0 \tag{4.3c}$$
$$a \nabla u \cdot \vec{n} = 0 \tag{4.3d}$$

where β is a multivalued function defined on $[0, 1]$ by

$$\beta(r) = \begin{cases} (-\infty, 0] & \text{if } r = 0. \\ 0 & \text{if } 0 < r < 1. \\ [0, \infty) & \text{if } r = 1. \end{cases}$$



In this model, u is interpreted as the pressure inside the material and takes the very simple form $u(x, t) = \operatorname{dist}(x, \Gamma_t)$ inside U_t. Its behaviour is analogous to the one of the height of the sandpile in the previous chapter. The region U_t is defined as the region where the pressure is positive.
Remark that equation (4.1) is a particular case of equation (1.1) when taking $F' = \beta$. In fact, it is the limit case as $p \to \infty$ of equation (1.1) when taking a power linearity $F(u) = \frac{1}{p}|u|^p$.

Notice that we have

$$\partial \chi_{[0,1]} = \beta$$

where $\chi_{[0,1]}$ is the characteristic function of the segment $[0,1]$.
Using this fact, we can, at least formally, use the minimizing movement scheme (1.7) with $F = \chi_{[0,1]}$, meaning that we define a sequence (w_k^τ) by $w_0^\tau = w_0$ and

$$w_{k+1}^\tau \in \arg\min_{0\le w\le 1} W_1\left(w - \left(w_k^\tau + \int_{k\tau}^{(k+1)\tau} w_k^\tau\right)\right) \tag{4.4}$$

Note that the proof of convergence (1.7) in [1] is only valid for power nonlinearities of the form $F(u) = \frac{1}{p}|u|^p$. Hence, the convergence of (4.4) to a weak solution of (4.1) is not guaranteed but is strongly expected as it is a limitcase of (1.7).
Denoting $w_s = (1+\tau)w_k^\tau$, we are then interested in the problem

$$\sup_{\phi\ 1-Lip} \min_{0\le w\le 1} \int_\Omega \phi(w - w_s)dx. \tag{4.5}$$

Optimality condition in w tells us that

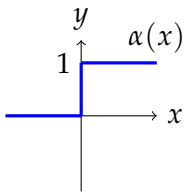
$$\phi + \beta(w) = 0.$$

Using $\alpha = \beta^{-1}$ we obtain

$$w = \alpha(-\phi) \tag{4.6}$$

where α is defined by

$$\alpha(r) = \begin{cases} 0 & \text{if } r < 0. \\ [0,1] & \text{if } r = 0. \\ 1 & \text{if } r > 0. \end{cases}$$



Using (4.6) in (4.5) yields

$$\inf_{\phi 1-Lip} \int_\Omega -\phi(\alpha(-\phi) - w_s)$$

which, for convenience, we write as the following problem using $\psi = -\phi$

$$\inf_{\psi 1-Lip} \int_\Omega \psi(\alpha(\psi) - w_s) \tag{4.7}$$

4.2 ADMM algorithm

The only difference between this model and the sandpile growth model is the cost function. As it is not smooth and non linear, the augmented Lagrangian techniques we used previously will not be well-suited.
Indeed, in this case the Lagrangian is

$$\mathcal{L}(\psi, q, \sigma) = \int_\Omega \psi(\alpha(\psi) - w_s) + \chi_B(q) + \langle \sigma, \nabla\psi - q\rangle + \frac{\lambda}{2}\|\nabla\psi - q\|^2$$

Define the functional J by

$$J(x) = x(\alpha(x) - w_s).$$

Let us try to apply the ADMM algorithm to this problem. This requires to compute the first order optimality condition in ϕ associated to the above Lagrangian. We obtain

$$\lambda\Delta\psi + w_s + \text{ div } (\sigma - \lambda q) \in \partial J(\psi) = \begin{cases} 0 & \text{if } \psi < 0 \\ [0,1] & \text{if } \psi = 0 \\ 1 & \text{if } \psi > 0 \end{cases}$$

Computing the solution of this very non linear PDE would be very complicated. However, the proximal operator of this function is easily computable so we will again use the Chambolle-Pock algorithm.

4.3 Chambolle-Pock Algorithm

It is necessary to compute the proximal operator of the functional J defined by

$$\text{prox}_{\tau J}(y) = \arg\min_x \frac{1}{2}(x-y)^2 + \tau x(\alpha(x) - w_s) \tag{4.8}$$

for some parameter $\tau > 0$. We define $G : x \to \tau x\alpha(x)$. The function G is lower semi-continuous and its subdifferential is equal to

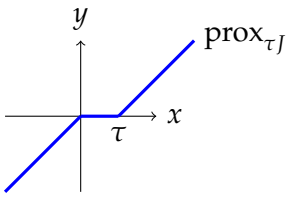
$$\partial G(x) = \begin{cases} 0 & \text{if } r < 0. \\ [0,\tau] & \text{if } r = 0. \\ \tau & \text{if } r > 0. \end{cases}$$

Optimality condition for problem (4.8) are :

$$z = \text{prox}_{\tau J}(y) \Rightarrow y - z + w_s \in \partial G(x)$$

which we can rewrite as

$$z = \begin{cases} y + w_s & \text{if } y + w_s < 0. \\ 0 & \text{if } 0 \le y + w_s \le \tau. \\ y + w_s - \tau & \text{if } y + w_s > \tau. \end{cases}$$



Notice that this is exactly (shifted by $\frac{\tau}{2}$) the expression of the proximal operator of the L^1 norm (also known as the "soft-thresholding operator").

4.4 Expected results

A geometric law for the evolution of the interface was obtained in [3]. It is given by

$$V = \gamma\left(1 - \frac{\kappa\gamma}{2}\right) \cdot \vec{\mathbf{n}}$$

where V is the outward normal velocity of Γ_t, κ is the signed curvature, γ is the distance to the "ridge" defined as the set of points for which there are at least two closest point to the boundary and $\overrightarrow{\mathbf{n}}$ is the outward normal to U_t.
Let us apply this formula to predict the behaviour of different simple shapes that will be useful to validate our numerical results.

- **Disc** :
  The curvature of the boundary is constant the points on the boundary are at a constant distance to the ridge (which in this case is the center of the disc). Thus the velocity is the same at all points and is equal to $\frac{1}{2}R_t$ where R_t is the radius of U_t. Thus the material will keep the form of a disc of radius $R_t = R_0 \exp(t/2)$.

- **Ring** :
  The outside circle will expand while the inner circle will shrink progressively until it disappears. At this point, the ridge distance will instantly moves from $\frac{R-r}{2}$ to R (denoting by R and r respectively the bigger and smaller radius), thus causing a sudden increase of velocity for the expansion of the outside circle.

- **Outward angle (for example the corners of a square)** :
  This corresponds to the case of $\kappa = +\infty$. Thus we get $V = 0$ so the corner have no velocity. In the case of a square the edges will progressively expand and round up. Indeed, the closer you are to the center of an edge, the further you are from the ridge, bigger is your velocity. A one point the corner will disappear and the whole shape will begin to spread out.

- **Inward angle** :
  On the contrary this leads to $V = +\infty$. Thus it has infinite speed and will instantly be removed.

Figure 4.1 taken from [6] shows the result of real life experiments using margarine and supports the above descriptions.

4.5 Numerical results

The algorithm was implemented in Matlab. Because computing them is the most costly part of the algorithm, we only check optimality conditions every 100 iterations. Figure 4.2 shows the compression of a disc. Results were obtained on a grid of 2^{18} points with a tolerance parameter of 10^{-6} on the optimality conditions and a time step $\tau = 0.1$. As expected the circle expands uniformly. To validate numerical results we tracked down the radius of the disc at each timestep. As shown on Figure 4.3, the radius follows the expected evolution $R_t = R_0 \exp(t/2)$.

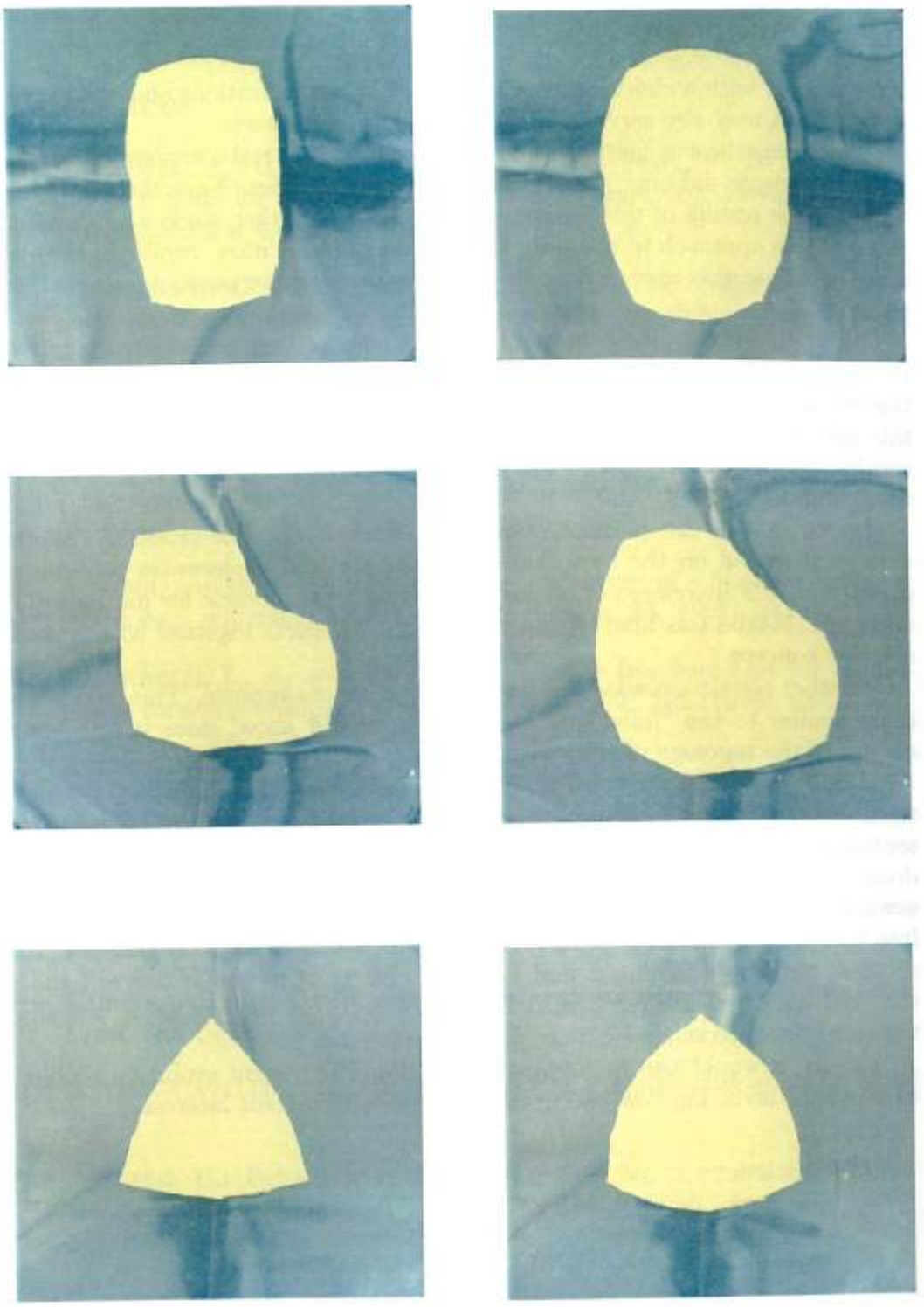


FIGURE 4.1: Compression of margarine. Taken from [6]

Time step	0.2	0.4	0.6	0.8	1
Nb of iterations	4800	5300	6200	13600	18800
CPU time (min)	2.7609	3.2399	3.3326	7.2886	9.4100
Time step	1.2	1.4	1.6	1.8	2
Nb of iterations	16200	21500	18900	31400	50400
CPU time (min)	7.8099	11.0967	10.0365	14.2926	23.7560

TABLE 4.2: Number of iterations and CPU time for the compression of a disc.

Figure 4.4 shows the compression of a ring. Results were obtained on a grid of 2^{16} points with a tolerance parameter of 10^{-6} on the optimality conditions and a time step $\tau = 0.2$. The resulting evolution is coherent with the theoretical analysis as the ring first closes to become a disc. As predicted we can notice an increase of expansion velocity after the hole closes.

Time step	0.2	0.4	0.6	0.8	1
Nb of iterations	2500	3200	6500	8200	13700
CPU time (min)	0.2301	0.2600	0.5384	0.6827	1.1704
Time step	1.2	1.4	1.6	1.8	2
Nb of iterations	42500	108300	127300	146900	191900
CPU time (min)	4.0265	9.5221	12.9530	13.8651	19.4344

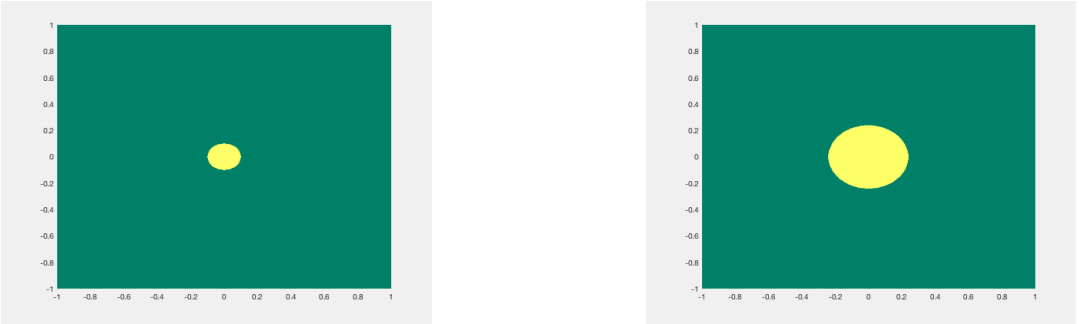
TABLE 4.4: Number of iterations and CPU time for the compression of a ring.

Figure 4.5 shows the compression of a square. Results were obtained on a grid of 2^{16} points with a tolerance parameter of 10^{-5} on the optimality conditions and a time step $\tau = 0.2$. We observe the expected behavior. The corners of the square stay motionless until being "swallowed" and the shape expands like a disc.

Time step	0.2	0.4	0.6	0.8	1
Nb of iterations	7200	10300	19600	25100	29200
CPU time (min)	0.7360	1.3461	1.6931	2.0640	2.3208
Time step	1.2	1.4	1.6	1.8	2
Nb of iterations	39100	53200	40700	43800	59900
CPU time (min)	4.0093	4.8629	4.0315	3.7581	5.7278

TABLE 4.6: Number of iterations and CPU time for the compression of a square.

Figure 4.6 shows the compression of a "Pac-Man", an interesting shape because it contains an inward angle. Results were obtained on a grid of 2^{16} points with a tolerance parameter of 10^{-5} on the optimality conditions and a time step $\tau = 0.2$. Again, the simulation match the theoretical predictions. The inward angle closes very fast.



(A) t = 0 (B) t = 1.9

FIGURE 4.2: Compression of a disc.

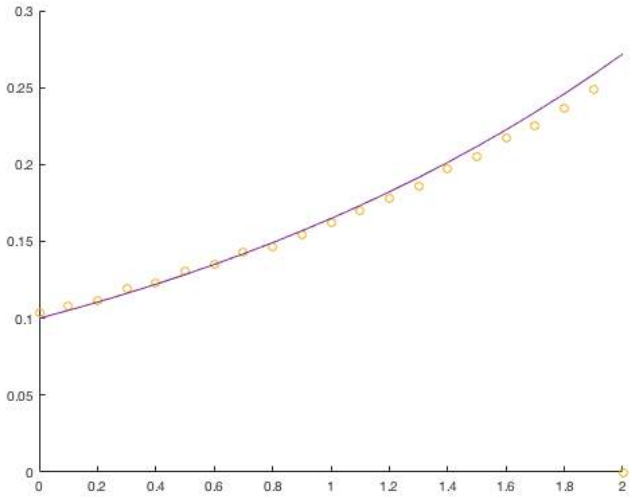
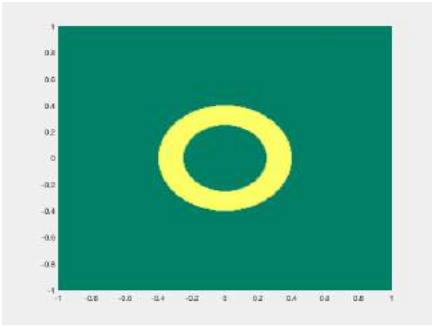


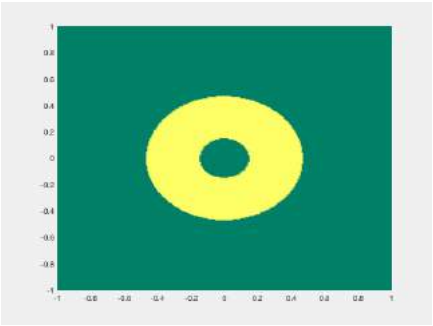
FIGURE 4.3: Scattered : Radius of the disc of Figure 4.2 at each timestep.
Filled : $R_0 \exp(t/2)$.

Time step	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6
Nb of iterations	17200	46400	37600	56600	63700	67300	72000	70100
CPU time (min)	1.2859	3.6627	2.8814	5.6868	4.8623	5.7639	5.5271	6.1300

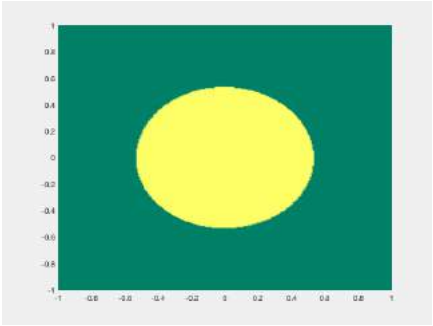
TABLE 4.8: Number of iterations and CPU time for the compression of a Pacman.



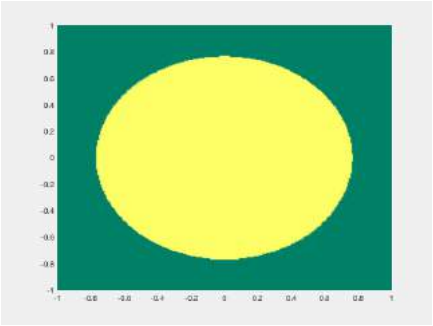
(A) t = 0



(B) t = 0.8

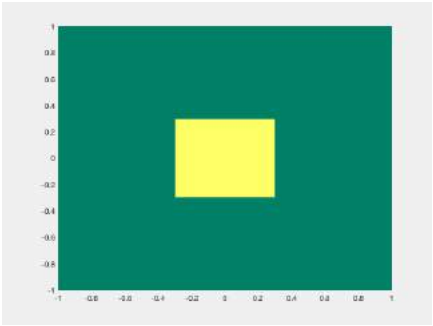


(C) t = 1.2

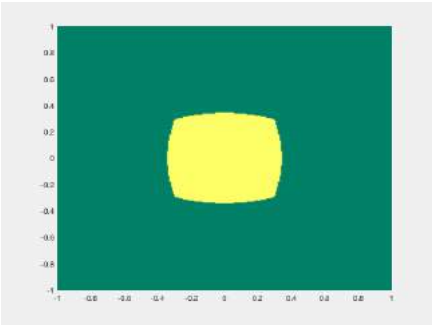


(D) t = 2

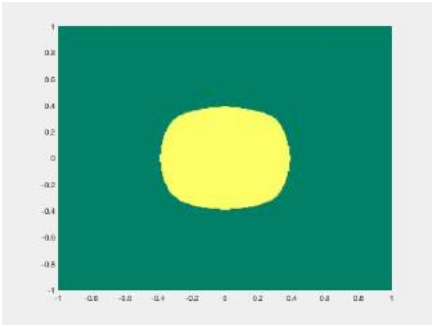
FIGURE 4.4: Compression of a ring.



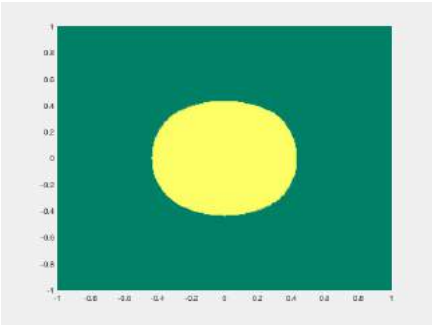
(A) t = 0



(B) t = 0.2

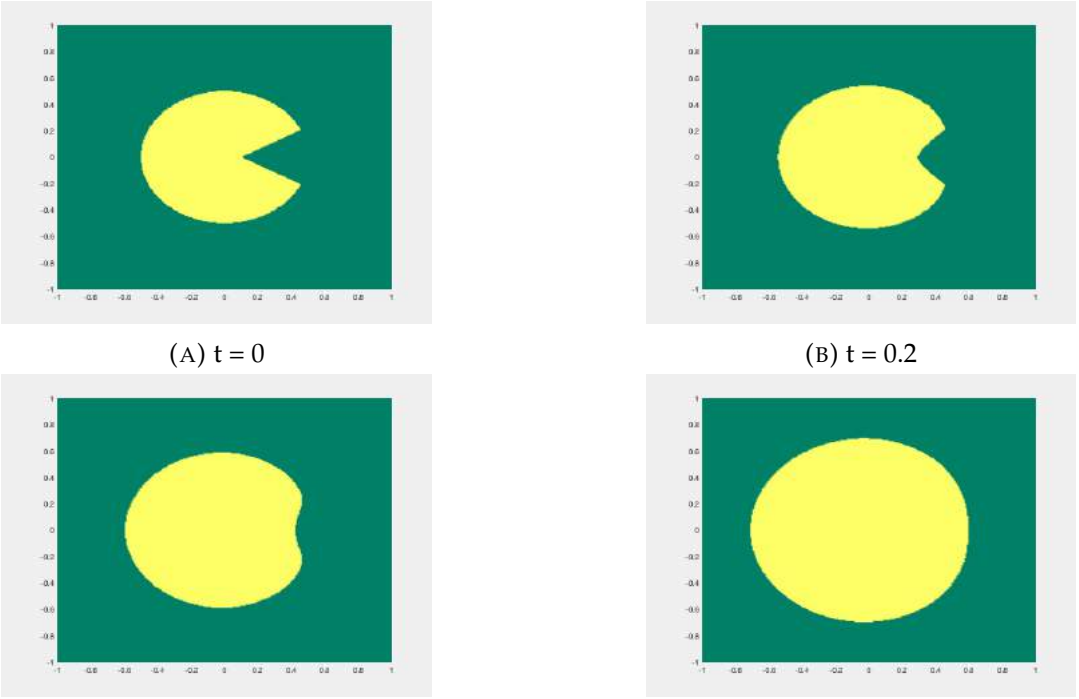


(C) t = 0.4



(D) t = 0.6

FIGURE 4.5: Compression of a square.



(A) t = 0 (B) t = 0.2

(C) t = 0.4 (D) t = 0.6

FIGURE 4.6: Compression of a Pacman.

Conclusion

The minimizing movement scheme with the 1-Wasserstein distance proved to be an efficient technique to solve the equation (1.1). Its numerical implementation using both the ADMM algorithm and the Chambolle-Pock algorithm are straightforward and efficient and our experiments yielded results that are coherent with both the expected theoretical results and previous attempts to solve numerically these equations. We were also able to present cases for which, to our knowledge, no numerical simulations have been presented before. For instance, moving sources for the sandpile problem or compression molding. We noticed the efficiency of the Chambolle-Pock algorithm compared to the Augmented Lagrangian method and thus we expect that it will used more frequently in numerical Optimal Transport.
Some perspectives of this work include

- Proving theoretically that the minimizing movement scheme we used for the compression molding problem converges to a weak solution of the equation. Indeed, the proof in [1] only deals with nonlinearities of the form $F(x) = |x|^p$ (the equation (4.1) being the limit case as $p \to \infty$).
- A further study of the optimal pit problem using this framework presented in section 3.4.3. using tools of optimal control.

Bibliography

[1] M. Agueh, G. Carlier, and N. Igbida. On the minimizing movement with the 1-Wasserstein distance. working paper or preprint, February 2017.

[2] L. Ambrosio, N. Gigli, and G. Savare. *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics. ETH Zürich. Birkhäuser Basel, 2005.

[3] G. Aronsson and L.C. Evans. An asymptotic model for compression molding. *Indiana University Mathematics Journal*, 51(1):1–36, 2002.

[4] G. Aronsson, L.C. Evans, and Y. Wu. Fast/slow diffusion and growing sandpiles. *Journal of Differential Equations*, 131(2):304 – 335, 1996.

[5] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, Jan 2000.

[6] Andreas Bergwall. *A geometric evolution problem arising in an asymptotic approach to compression moulding*. PhD thesis, Linköping University, 1998.

[7] Lawrence C. Evans and Wilfrid Gangbo. Differential equations methods for the monge-kantorovich mass transfer problem / l. c. evans, w. gangbo. 07 2018.

[8] Alexandre Caboussat and Roland Glowinski. A numerical method for a non-smooth advection-diffusion problem arising in sand mechanics. 8, 10 2008.

[9] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, Jan 2004.

[10] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, May 2011.

[11] Jonathan Dorfman and Lawrence C. Evans. A " lakes and rivers " interpretation for the singular limit of a nonlinear diffusion pde. 2009.

[12] Ekeland, Ivar and Queyranne, Maurice. Optimal pits and optimal transportation. *ESAIM: M2AN*, 49(6):1659–1670, 2015.

[13] F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.

[14] Morgane Henry, Emmanuel Maitre, and Valérie Perrier. Optimal Transport using Helmholtz-Hodge Decomposition and First-Order Primal-Dual Algorithms. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4748–4752, Quebec City, QC, Canada, September 2015. IEEE.

[15] R. Jordan, D. Kinderlehrer, and F. Otto. The variational formulation of the fokker-planck equation. *SIAM J. Math. Anal*, 29:1–17, 1999.

[16] L. Prigozhin. Quasivariational inequality describing the shape of a poured pile. *Zhurnal Vichislitel'noy Matematiki i Matematicheskoy Fiziki*, pages 1072–1080, 1986.

[17] L. Prigozhin. Sandpiles and river networks: Extended systems with nonlocal interactions. *Phys. Rev. E*, 49:1161–1167, Feb 1994.

[18] Leonid Prigozhin. Variational model of sandpile growth. *European Journal of Applied Mathematics*, 7(3):225–235, 1996.