Models for Particle Image Velocimetry:
Optimal Transportation and Navier-Stokes Equations

by

Louis-Philippe Saumier Demers
B.Sc., University of Sherbrooke, 2008
M.Sc., University of Victoria, 2011

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mathematics and Statistics

Models for Particle Image Velocimetry:
Optimal Transportation and Navier-Stokes Equations

by

Louis-Philippe Saumier Demers
B.Sc., University of Sherbrooke, 2008
M.Sc., University of Victoria, 2011

Supervisory Committee

Dr. Martial Agueh, Supervisor
(Department of Mathematics and Statistics)

Dr. Boualem Khouider, Supervisor
(Department of Mathematics and Statistics)

Dr. Slim Ibrahim, Departmental Member
(Department of Mathematics and Statistics)

Dr. Peter Oshkai, Outside Member
(Department of Mechanical Engineering)

Supervisory Committee

Dr. Martial Agueh, Supervisor
(Department of Mathematics and Statistics)

Dr. Boualem Khouider, Supervisor
(Department of Mathematics and Statistics)

Dr. Slim Ibrahim, Departmental Member
(Department of Mathematics and Statistics)

Dr. Peter Oshkai, Outside Member
(Department of Mechanical Engineering)

ABSTRACT

We introduce new methods based on the L^2 Optimal Transport (OT) problem and the Navier-Stokes equations to approximate a fluid velocity field from images obtained with Particle Image Velocimetry (PIV) measurements. The main idea is to consider two successive images as the initial and final densities in the OT problem, and to use the associated OT flow as an estimate of the underlying physical flow. We build a simple but realistic model for PIV data, and use it to analyze the behavior of the transport map in this situation. We then design and implement a series of post-processing filters created to improve the quality of the numerical results, and we establish comparisons with traditional cross-correlation algorithms. These results indicate that the OT-PIV procedure performs well on low to medium seeding densities, and that it gives better results than typical cross-correlation algorithms in some cases. Finally, we use a variational method to project the OT velocity field on the space of solutions of the Navier-Stokes equations, and extend it to the rest of the fluid domain, outside the particle locations. This extension provides an effective filtering of the OT solution beyond the local post-processing filters, as demonstrated by several numerical experiments.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would first like to acknowledge that a large part of this work has already been published. The corresponding references are as follows.

1. L.P.Saumier, B.Khouider and M.Agueh, **Optimal Transport for Particle Image Velocimetry: Real Data and Post-Processing Algorithms**, SIAM Journal on Applied Mathematics 75, no. 6 (2015): 2495-2514.

2. L.P.Saumier, B.Khouider and M.Agueh, **Optimal Transport for Particle Image Velocimetry**, Communications in Mathematical Sciences 13, no. 1 (2015): 269-296.

3. L.P.Saumier, M.Agueh and B.Khouider **An Efficient Numerical Algorithm for the L^2 Optimal Transport Problem with Periodic Densities**, IMA Journal of Applied Mathematics 80, no. 1 (2015): 135-157.

I would also like to express my gratitude to the National Sciences and Engineering Research Council of Canada (NSERC) as well as to the University of Victoria for funding me with scholarships and fellowships for many years during my studies. I would finally like to thank my advisors Dr. Martial Agueh and Dr. Boualem Khouider for the many years of valuable advice and fruitful discussions, which will have a profound impact on my future endeavors.

DEDICATION

To my wife Ellie and my daughter Telesa.

Chapter 1

Introduction

Particle Image Velocimetry, or PIV, is a widely used experimental fluid mechanics technique to measure the velocity of a fluid flow [1, 36, 57]. Small particle-tracers with desirable light-reflection properties are seeded into the moving fluid. A pulsing laser then illuminates the fluid at regular time intervals, and an image of the light reflected back by the particles is recorded at every pulse. The experimenter therefore collects a sequence of images and uses them to approximate the velocity field of the fluid. The resulting field is deemed a good representation of the actual flow velocity, under the assumption that the particles are small enough so that their effect on the fluid is negligible.

The velocity field is typically approximated from the discrete sequence of images using the cross-correlation algorithm [23, 58]. In a nutshell, one splits the images into smaller sub-images called interrogation windows. Then, the average displacement of particles inside every window is found by calculating the maximum of the cross-correlation function. A vector field is created on the grid given by the windows and is used to approximate the underlying physical field. While algorithms based on this procedure have been successful at obtaining good approximations of the velocity, accurate results can only be obtained with cross-correlation when at least several particles are present in each window. This limits the maximum resolution achieved by these methods. While so-called super-resolution methods have been developed to alleviate this issue [1, 20], improvements are still needed [9, 33, 45, 46, 47]. Also, issues arising from particles crossing the boundary of interrogation windows between pulses lead to errors in the recovered field [19]. The effect of these errors is usually dampened by the use of post-processing filters, but they are not always easy to detect and correct [36].

In this work, in order to address these issues, we first propose a new way to approximate the velocity field from PIV images. Our idea is to use the framework of the L^2 Optimal Transport (OT) problem, also know as the L^2 Monge-Kantorovich problem [31, 54, 55]. In this problem, the goal is to find the most efficient way of redistributing a material density f to a target density g, in the sense that the total L^2 transportation distance is minimized. Motivated by recent improvements in the numerical resolution of the L^2 OT problem [2, 3, 10, 17, 42, 39], we propose to use every two successive PIV images as a pair of initial and final mass distributions, and to find the flow related to the OT map to approximate the fluid velocity. While the optimal transportation theory has already been used in image processing [6, 18, 32, 35, 37, 51], it is the first time that it is applied to estimate velocity fields from PIV data.

In order to understand the behavior of the OT map in the context of PIV, we introduce in Chapter 3 a basic model of PIV-like images. More precisely, we approximate the images of particles by weighted sums of Gaussian distributions with a fixed standard deviation σ and an additive background noise constant $1 - r$ to mimic the randomly scattered light by the medium. The Gaussian assumption with uniform σ is motivated by the facts that particles are usually indistinguishable from one another, and that the light scattered back by a point source and captured by a lens is distributed according to the Airy function, which itself is very close to a Gaussian [36]. Also, the weight parameters m_i of these distributions capture the non-uniform brightness displayed in images. This non-uniformity is itself associated with the fact that particles in 2D-PIV may be located at different distances from the thin illuminated plane (with respect to the direction perpendicular to the plane). Equipped with this model, we derive rigorous error bounds for a particle's predicted final position as a function of the parameters σ, r and m_i, in the case where particles are far from one another (which we call the tracking range). We start with the model in dimension one to prove that the error converges linearly in σ and $(1-r)/r$, and then generalize the analysis to dimension 2. Then, we show that the ratio of the OT-PIV errors associated with two particles is inversely proportional to their weights. This result indicates that, in terms of the approximated velocity field, the vectors associated with the brightest particles are more faithful. We also demonstrate that the tracking of brighter particles may be feasible over longer distances, when they escape the tracking range.

After conducting this theoretical analysis, we proceed in Chapter 4 to numerical experiments. One possible way of recovering the field would be to first identify the meaningful particles and then solve the corresponding assignment problem. There are several algorithms available to solve such an assignment problem, some based on the simplex algorithm [16, 62], the Hungarian algorithm [25, 62], the auction algorithm [4], but in general, for densities consisting of thousands of Dirac delta distributions, this remains a challenging problem [15]. In the specific case of PIV, one can of course limit the valid assignments to close neighbors, but even then the number of possible matches can still be high. In addition, individual particles are not always easy to identify from the light density distribution; bright clusters of light can represent the light scattered by several particles. Motivated partly by these reasons, we use here a continuous approach for the numerical resolution of the problem. We therefore present in Chapter 4 the results of numerical experiments demonstrating some strengths and weaknesses of the OT method for PIV. For all the numerical tests, we use the procedure given in [39] to solve the OT problem. We begin by analyzing situations where the data are given analytically by our Gaussian model and use the results to gain intuition. Then we introduce a series of post-processing strategies in order to improve the accuracy of the retrieved velocity field. These filters are designed to either correct the errors in the OT target positions predicted by the theory, or to discard erroneous vectors due to the splitting of individual particle masses or the tracking of particles which do not carry enough weight in the light distribution captured by the images. Then, we present experiments with realistic PIV images, and compare our method with a typical cross-correlation algorithm [51, 52]. The various experiments show that the OT-PIV procedure can successfully follow individual particles within the tracking range, that tracking can be extended outside this range for brighter particles, and that the OT-PIV method performs better than the cross-correlation method for lower seeding densities. We also show that the OT procedure can be enhanced with multiple passes similar to the ones usually employed with cross-correlation.

The optimal transport theory for PIV has many major advantages. Indeed, it is a global approach, i.e. it is not necessary to split the region into interrogation windows to compute the velocity field; the solution is computed everywhere at once. Several algorithms are now available to compute the solution of the optimal transport problem, and some, like the one we employ here, are quite efficient. Moreover, even though we use Gaussian distributions to model individual tracers (which makes it

easier to mathematically analyze the behavior of OT applied to PIV), the flexibility of OT would easily allow us to take virtually any distribution of tracers. In addition, new advances in the field of Tomographic PIV [13] allow the technique to be used to approximate three-dimensional flows, and the OT-PIV technique presented here could easily be extended to tackle this problem. In that case, the assumption that the flow is not too turbulent would not be necessary anymore to ensure that the densities have the same amount of mass in the Optimal Transport problem.

Despite these many advantages, there is one major difficulty inherent to Optimal Transport that needs to be addressed. It is well-known [54] that the velocity field associated to the time-dependent OT map and corresponding density solves the pressureless Euler equations at zero temperature. The solutions to this system are far from solving the more physically relevant Navier-Stokes equations, and thus the OT field is limited to the non-uniformly distributed locations of the particles. In addition, as mentioned previously, not all the vectors at particle locations are reliable and the grid usually has to be further reduced to improve the quality of the approximation. Finally, the flow given by Optimal Transport is stationary by nature, and also limits the applicability of the method.

We therefore propose in Chapter 5 a new way to extend the discrete OT-PIV field to the whole physical domain and time-interval between the capture of two images. This method consists in finding the closest (in a certain way) Navier-Stokes equations solution to the field given by OT. More specifically, we use the vorticity form of the Navier-Stokes equations combined with a variational formulation of the error and a penalty parameter k_e on the kinetic energy, to find a field as a minimizer of the related functional with respect to the initial vorticity used in the Navier-Stokes equations. We take an approach similar to the vortex blob method [26] which consists in expending the initial vorticity using radial basis functions. We use Gaussian blobs of fixed width given by another parameter ϵ, and solve the corresponding optimization problem with Fourier transforms and a quasi-Newton method. The interplay of the parameters k_e, ϵ and the number of blobs selected N_b, contributes to filtering the vector field from noisy measurements, thus making it a data assimilation method [22].

While many methods to interpolate and filter a non-uniformly sampled vector field have been proposed in the past [21, 22, 24, 34, 38, 43, 44, 49, 56], our method stands out for a few different reasons. First, it is specifically designed to only use the approximated vector field and not the information from the PIV images. This makes the method applicable to other situations unrelated to PIV. Second, our procedure

only aims at recovering the flow on the time interval between the capture of two images. It could therefore be used on its own for longer periods of time by simply repeating it for every pair of images, or it could be used to initialize more involved algorithms taking into account multiple time-intervals. Third, we do not assume that the recovered flow is stationary; we allow it to vary in time by not neglecting the time-derivatives. Finally, the method we present is designed to give good results even for fields defined on sparse and non-uniform grids. While it is motivated by the fact that the vectors obtained with OT-PIV are mostly reliable for brighter particles when outside the tracking range, the extension technique via the Navier-Stokes equations could be used for discrete fields obtained for example by cross-correlation algorithms.

In summary, this dissertation is organized as follows. In Chapter 2, we review some related background material on Particle Image Velocimetry, Cross-Correlation and Optimal Transport. In Chapter 3, we layout our model for PIV images and derive analytical bounds for the error in the particles' predicted target positions by Optimal Transport. Then, we present in Chapter 4 numerical experiments with the OT-PIV method on synthetic and real data. We also introduce different post-processing filters and study their effect on selected examples. In Chapter 5, we introduce and analyze our technique to extend and filter the OT field with the Navier-Stokes equations. Finally, we give some additional remarks and conclusions in Chapter 6.

Chapter 2

Theoretical Background

2.1 Particle Image Velocimetry

Particle Image Velocimetry, or PIV, is a technique used to measure the velocity field of fluid flows in laboratory experiments [1, 36, 57]. The main idea is to seed many small particles in the fluid and let them evolve with the flow as they are illuminated by a pulsing laser. The distribution of light scattered by these particles is recorded every time the laser pulses, thus creating a discrete sequence of images. The data generated is then analyzed and a velocity field is created. Provided the particles accurately follow the motion of the fluid, this field gives a good approximation of the fluid velocity.

PIV presents several advantages for recovering experimental velocity fields [36]. Indeed, since the particles are selected small enough to not alter the flow, PIV provides non-intrusive velocity measurements. Also, thanks to the illumination with the pulsing laser, these measurements are instantaneous and the approximated field obtained from the discrete sequence of images covers the whole region under study. This explains why PIV is widely used in industry. It is used for example in aerodynamics [61], in biomechanics [27], in geotechnical testing [59] and in animal locomotion research [51, 53].

Tradionnally, PIV is employed for laminar flows and is performed by illuminating only a very thin slice of the fluid, so the image recorded is two-dimensional (see Figure 2.1). One main drawback of this method is the possible introduction of noise coming from out-of-plane loss of particles when the flow is not perfectly two-dimensional; that is when particles escape the thin slice illuminated with the laser between consecutive

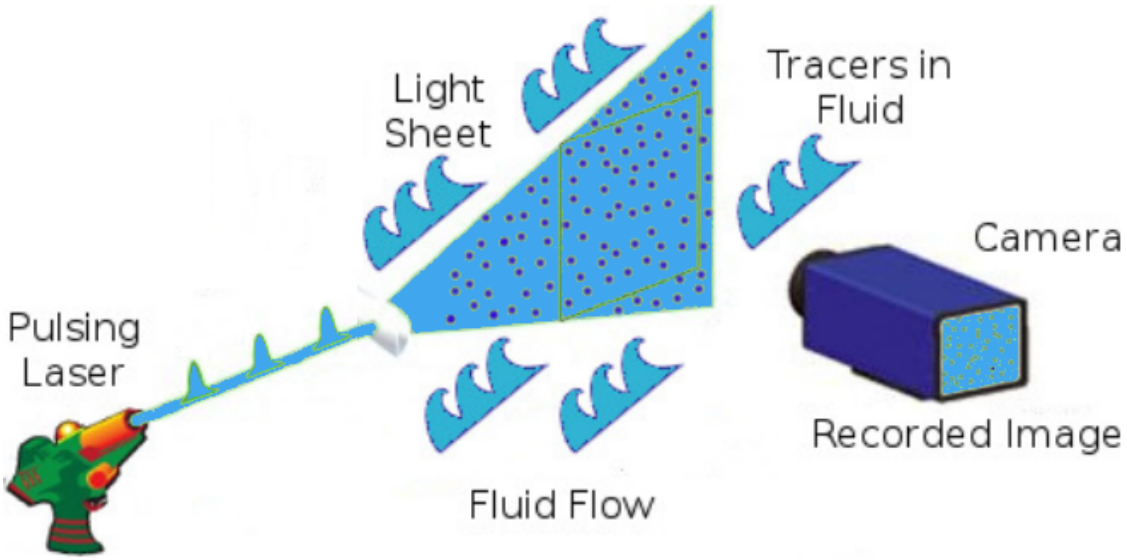
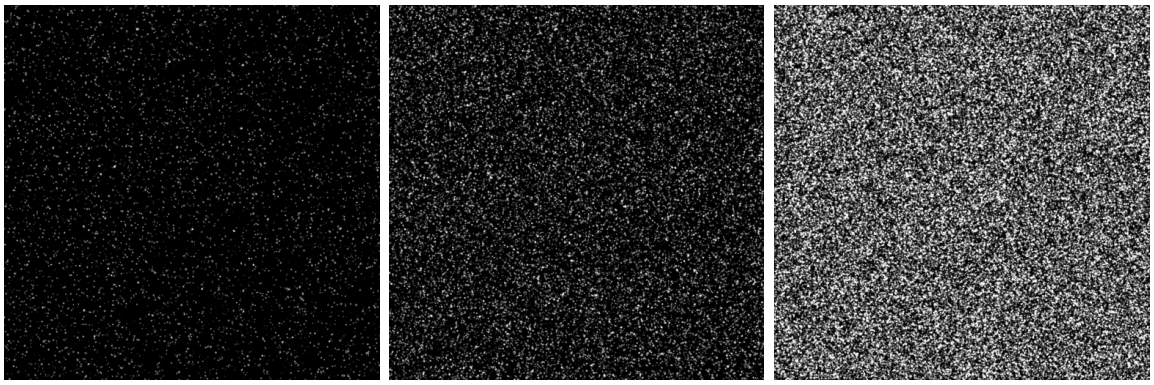


Figure 2.1: Illustration of a 2D-PIV setup.

pulses. However, new techniques such as Tomographic PIV make it possible to obtain three-dimensional data [13] and avoid this problem.

The particles are typically indistinguishable, and thus individual particle tracking is not always easy to accomplish, depending on the flow, the time-intervals considered and the particle concentration. In fact, when the average distance between distinct particles is much larger than the mean particle displacement, individual tracking of most particles becomes feasible and the technique of particle image velocimetry is referred to as particle tracking velocimetry (PTV). We also refer to this case as the tracking range. On the other hand, when the particle density is so high that it is difficult to identify individual particles from the images, the name Laser Speckle Velocimetry (LSV) is employed. The three modes of PIV are illustrated in Figure 2.2. In all three cases, the images displayed show particles with seemingly different brightness. This is mostly due to the following two reasons: several particles close to each other may appear as just one brighter particle on the image, and particles closer to the center of the thin illuminated plane will reflect more light than particles further away from the center (with respect to the direction perpendicular to the plane). The images in this figure are taken from the 2001 PIV challenge [9, 45]. The objective of this open challenge, which has been held four times up to now, is to assess the current state-of-the-art of PIV methods, and thus guide the future development of the field.



(a) PTV: Low Seeding Density (b) PIV: Medium Seeding Density (c) LSV: High Seeding Density

Figure 2.2: The different modes of Particle Image Velocimetry.

2.2 Cross-Correlation Algorithms

The standard way of extracting the velocity field from the sequence of recorded particle images is to split the images into smaller pieces called interrogation windows and then use the cross-correlation function on each window to create each vector [23, 36, 57]. More specifically, let us denote by f and g the density functions given by two consecutively recorded images, and by Ω the domain on which they are defined. Let us decompose Ω into (potentially overlapping) subregions B_i so that

$$\Omega = \bigcup_{i=1}^{b} B_i,$$

and take $f_i(x) = \mathcal{X}_{B_i}(x)f(x)$ and $g_i(x) = \mathcal{X}_{B_i}(x)g(x)$ to be the restriction of f and g to B_i. Here, $\mathcal{X}_{B_i}$ denotes the characteristic function of B_i. Consider the cross-correlation between f_i and g_i, which is defined as

$$(f_i \star g_i)(\Delta x) = \int_{\mathbb{R}^2} f_i(x) g_i(x + \Delta x)\, \mathrm{d}x, \tag{2.1}$$

where $\Delta x \in \mathbb{R}^2$ is a fixed displacement. Then, the average displacement of the tracers inside one interrogation window B_i can be found by computing the maximizer $\widetilde{\Delta x_i}$ of the cross-correlation function $f_i \star g_i$. The fluid's velocity field inside B_i is

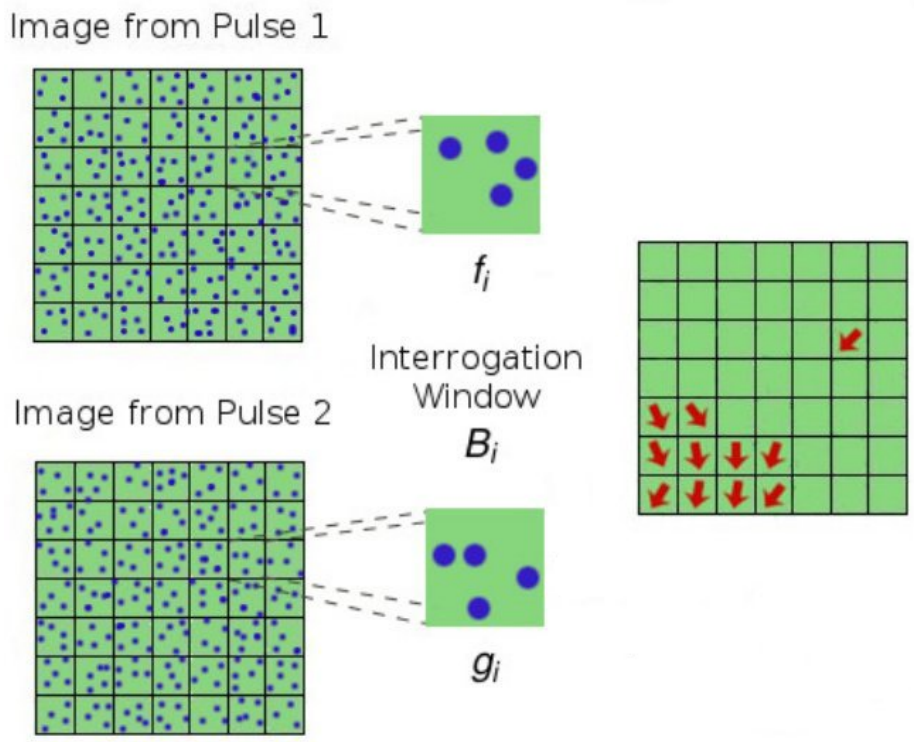


Figure 2.3: Illustration of interrogation windows.

approximated by the uniform field

$$\mathbf{v}_{\mathrm{CC}}\Big|_{B_i} = \frac{\widetilde{\Delta x}_i}{\Delta t}$$

where Δt is the time elapsed between the capture of the two consecutive images. By repeating this procedure for every interrogation window, one creates a grid of displacements from which the approximate velocity field is recovered. Figure 2.3 displays an illustration of this procedure. Note that in practice, because of the small time intervals Δt between pulses, the integral in (2.1) only needs to be computed for values of x inside or close to B_i.

The accuracy of the field obtained with this method relies on the assumption that the flow is more or less constant inside each interrogation window. However, it is not desirable to take the windows so small that only a few particles are present inside them. Indeed, in such a case, a particle crossing a window's boundary in between two pulses will have a greater effect on $\widetilde{\Delta x}_i$, and can lead to significant errors. This limits the resolution of the field obtained with this technique. In fact, this problem

of particles moving from one window to another may cause errors even when more particles are included, since particles may have different brightness on the images, or the seeding might not be uniform in Ω.

Some more advanced cross-correlation algorithms have been developed to dampen these issues. For example, instead of computing the maximum of the cross-correlation of f_i and g_i, we can consider the cross-correlation of f_i and a larger part of g than simply g_i, to account for particles moving out of B_i after Δt units of time. In addition, it is quite common to employ a multiple pass scheme where the aforementioned procedure is used as a first pass to obtain a first estimate for the velocity field v. After being processed to get rid of potential outliers, the approximate vector field is then used to offset the interrogation windows for the first density f, and the cross-correlation is recomputed with these new windows. The method is then repeated until the window offset converges to $\pm$ 1 pixel. Multi-resolution schemes in which the grid of interrogation windows is successively refined after each pass are often used in practice. Finally, typical resolution algorithms include subpixel approximations of the correlation peak, for example by interpolating the points close to the maximum with a 1D Gaussian distribution (in both directions), and then correcting the value of the maximum to match the maximum of the Gaussian. More details on all these procedures can be found in [36].

2.3 Optimal Transport

As explained in Chapter 1, one of the main ideas of this work is to use the solution of the Optimal Transport (OT) problem related to successive particle image distributions to approximate the velocity vector field for PIV. The OT problem [31, 54, 55] consists in finding the least expensive way of redistributing material from one configuration to another. By least expensive, we mean that the optimal map T redistributing the material minimizes a given transportation cost. More precisely, if x is the initial position of a point in a domain Ω and y is its final position in a domain Ω', then the cost of moving x to y is given by a function $c(x, y)$. Let Ω and $\Omega^{'}$ be closed and bounded subsets of $\mathbb{R}^d$. We model the material distributions by two probability densities denoted f and g. In the case where $c(x, y)$ is the square of the usual Euclidean

distance, this problem, called the L^2 optimal transport problem, reads

$$\text{Minimize } I[T] = \frac{1}{2}\int_\Omega |x - T(x)|^2 f(x)\,\mathrm{d}x, \tag{2.2}$$

over all maps T transporting f to g: $T\#f = g$. The expression $T\#f = g$, called the push-forward condition, is essentially a mass conservation constraint which can be expressed by

$$\int_{\Omega'} \mathcal{X}_E(y)g(y)\,\mathrm{d}y = \int_\Omega \mathcal{X}_{T^{-1}(E)}(x)f(x)\,\mathrm{d}x = \int_\Omega \mathcal{X}_E(T(x))f(x)\,\mathrm{d}x, \tag{2.3}$$

for any Borel set E in Ω'. Equivalently, we have

$$\int_{\Omega'} \zeta(y)g(y)\,\mathrm{d}y = \int_\Omega \zeta(T(x))f(x)\,\mathrm{d}x$$

for any $\zeta \in \mathcal{C}(\Omega')$. Then, by the change of variable $y = T(x)$, assuming T is at least $\mathcal{C}^1$, we obtain

$$\int_\Omega \zeta(T(x))g(T(x))|\det(\mathrm{D}T)|\,\mathrm{d}x = \int_\Omega \zeta(T(x))f(x)\,\mathrm{d}x.$$

Because ζ is arbitrary, we arrive at

$$g(T(x))|\det(\mathrm{D}T)| = f(x). \tag{2.4}$$

In [5], Brenier proved that (2.2) has a unique solution T which is the gradient of a convex potential, i.e. $T = \nabla\Psi$ where Ψ is convex. If we insert this formula in (2.4), we obtain the so-called Monge-Ampère equation

$$g(\nabla\Psi(x))\det(\mathrm{D}^2\Psi) = f(x). \tag{2.5}$$

This equation is a nonlinear second order elliptic partial differential equation. It is well-known (see [54] for example) that the gradient of a convex function $\Psi \in \mathcal{C}^2(\Omega)$ which solves (2.5) is the optimal solution of the L^2 OT problem (2.2). Caffarelli [7, 8] proved that if both f and g are of class $\mathcal{C}^{0,\alpha}(\Omega)$ for $0 < \alpha < 1$, then $\Psi \in \mathcal{C}^{2,\alpha}(\Omega)$, and thus in this case the existence of such a classical solution to the Monge-Ampère equation is guaranteed. The OT problem described so far is independent of time, but has a natural time-dependent extension. Indeed, consider the time-dependent L^2 OT

problem from time $t = 0$ to $t = \Delta t$, which reads

$$\text{Min} \int_\Omega \int_0^{\Delta t} \left| \frac{\mathrm{d}}{\mathrm{d}t} T_t(x) \right|^2 f(x)\, \mathrm{d}t\, \mathrm{d}x \tag{2.6}$$

over the set of all flow maps T_t such that $T_0 = \text{Id}$ and $T_{\Delta t} \# f = g$. Note that this new problem is linked to the previous one by the equality

$$\frac{1}{\Delta t}|y - x|^2 = \inf\left\{ \int_0^{\Delta t} \left| \frac{\mathrm{d}}{\mathrm{d}t} z_t \right|^2 \mathrm{d}t;\ z_t \in \mathcal{C}^1([0, \Delta t]; \Omega),\ z_0 = x,\ z_{\Delta t} = y \right\}.$$

The associated optimal solution $T_t(x)$ can be written as

$$T_t(x) = x + \frac{t}{\Delta t}\Big(T(x) - x\Big), \tag{2.7}$$

where $0 \leq t \leq \Delta t$ and $T(x)$ is the optimal solution of (2.2). The map T_t, which interpolates the identity map and the optimal map T, was first introduced by McCann in [30]. Observe that the trajectories are straight lines and that the associated velocity field v_T is constant:

$$v_{\mathrm{T}} = \frac{T(x) - x}{\Delta t}. \tag{2.8}$$

In addition, one can consider the density ρ_t at intermediate times $\rho_t = T_t \# f$. This time-dependent density satisfies $\rho_0 = f$ and $\rho_{\Delta t} = g$. The system of partial differential equations describing the evolution of v_{T} and ρ_t is given by

$$\begin{cases} \dfrac{\partial}{\partial t} v_{\mathrm{T}} + (v_{\mathrm{T}} \cdot \nabla) v_{\mathrm{T}} = 0 \\ \dfrac{\partial}{\partial t} \rho_t + \operatorname{div}(\rho_t v_{\mathrm{T}}) = 0, \end{cases} \tag{2.9}$$

and corresponds to the continuity equation coupled with the pressureless Euler equations, which together form the Eulerian system of optimal time-dependent mass transportation [54]. Finally, since $T = \nabla \Psi$, we can also see that the approximate flow given by OT corresponds to a pressureless potential flow

$$v_T = \nabla \left(\frac{\Psi(x) - |x|^2/2}{\Delta t} \right).$$

All this information on the OT flow will be useful to understand the behavior of the method presented in subsequent chapters.

Chapter 3

Optimal Transport Model for Particle Image Velocimetry

3.1 Particle Model and Approximate Field

In the context of PIV, it is natural to consider the densities f, g in the OT problem as the images captured with the light scattered back by particles at time $t = 0$ and $t = \Delta t$, respectively, where Δt is the time elapsed between the two successive images. The OT map T can potentially provide a good approximation of the particle trajectories on $[0, \Delta t]$, provided Δt is small enough. In this case, the velocity field given by the time-dependent OT problem (2.8) can be used to approximate the field associated with the PIV problem on $[0, \Delta t]$. Therefore, we wish to study in this chapter the behavior of the Optimal Transport map in such a situation.

In order to analyze this behavior, we build a simple but realistic model for f and g. Let $\Omega \subset \mathbb{R}^2$ be the domain corresponding to the region on which the images are given. For f, g on Ω, we select sums of Gaussian distributions that are centered respectively at the initial and final positions of the tracer particles with a common standard deviation σ:

$$\begin{aligned} f(x_1, x_2) =& \frac{r}{M} \sum_{i=1}^{N_p} \frac{m_i}{k_{\lambda_i}} \exp\Big(- \frac{1}{2\sigma^2} \big[(x_1 - \lambda_{i_1})^2 + (x_2 - \lambda_{i_2})^2 \big] \Big) + (1 - r), \\ g(x_1, x_2) =& \frac{r}{M} \sum_{i=1}^{N_p} \frac{m_i}{k_{\mu_i}} \exp\Big(- \frac{1}{2\sigma^2} \big[(x_1 - \mu_{i_1})^2 + (x_2 - \mu_{i_2})^2 \big] \Big) + (1 - r), \end{aligned} \tag{3.1}$$

where

$$M = \sum_{i=1}^{N_p} m_i, \quad k_{\lambda_i} = \int_\Omega \exp\Big(- \frac{1}{2\sigma^2}\big[(x_1 - \lambda_{i_1})^2 + (x_2 - \lambda_{i_2})^2\big]\Big)\mathrm{d}x,$$

and the other normalizations constants k_{μ_i} are defined similarly. In addition, $r \in [0, 1]$, N_p represents the number of particles in Ω, and $\lambda_i = (\lambda_{i_1}, \lambda_{i_2})$ and $\mu_i = (\mu_{i_1}, \mu_{i_2})$ are respectively the initial and final positions of particle i.

The choice of distributions is physically motivated, because the response of an imaging system to a point source, namely the point spread function, is well approximated by a Gaussian distribution [36]. Note that it may also mimic the measurement uncertainty for the actual locations of the particles. The uniform standard deviation σ is motivated by the fact that the particles used as tracers are typically indistinguishable from each other. However, we also allow the Gaussian distributions to have different weights m_i in the total density to account for the non-uniform brightness displayed in the images, which is itself mostly due to the variable perpendicular distance to the center of the thin illuminated plane (this is easily seen in the images displayed in Figure 2.2). Besides the standard deviation σ and the weights m_i, we introduce another parameter r to lift up f and g away from zero by adding the constant $1 - r$. This constant can be viewed as a parameter modeling a uniformly distributed background noise in the light measurements due to the medium in which the particles evolve. Finally, we rescale the densities' Gaussian parts so they remain well-defined probability densities on the bounded domain Ω.

There are a few other underlying assumptions with this model. First, we implicitly assume that no particle escapes or enters Ω within the time interval $[0, \Delta t]$, i.e. N_p remains constant on that interval. Second, we assume that the shape and weight of the Gaussian distributions is not altered by the flow. In other words, this corresponds to assuming that σ and the weights m_i are constant on $[0, \Delta t]$ so that g is obtained from f only by moving the center of the Gaussian distributions from λ_i to μ_i. Both assumptions are not always satisfied in practice. For example, a particle moving slightly out of the camera's focus zone will appear blurrier on the image, hence changing the Gaussian's standard deviation. In addition, when using particles with varying shapes such as aerosols, smaller particles may coalesce around a larger particle after a short time, changing the physical size of the tracer and thus changing σ. However, it is not unrealistic to assume that if the flow is not too turbulent, if

the particles do not change physical shape and if Δt is small enough, the number of particles N_p does not change and the weight and shape differences between the initial and final densities remain small. This explains the equality assumptions.

Let us now consider the map T solving the L^2 OT problem (2.2) with densities given by (3.1). As previously mentioned, we consider the corresponding velocity field (2.8) associated with the time-dependent L^2 OT problem:

$$v_{\mathrm{T}}(x) = \frac{T(x) - x}{\Delta t}$$

to approximate the target field in the PIV problem. Indeed, due to the optimal nature of the map T and the small particle displacements because Δt is small, we expect $T(\lambda_i)$ to give a good estimate of the final position μ_i of particle i. This means that $v_T(\lambda_i)$ should be close to $(\mu_i - \lambda_i)/\Delta t$, a first order in time approximation of the target vector at position λ_i. However, for values of x which do not coincide with particle locations, it is not guaranteed that $v_T(x)$ gives a good estimate of the target velocity vector at position x. In fact, the system of equations (2.9) describing the evolution of v_T is limited to very specific physical situations and thus will not accurately represent the target field in most cases. The approximate field obtained with Optimal Transport should therefore be restricted to the non-uniform distribution of discrete points given by the particle's initial locations. We will present in Chapter 5 a method to extend this field to the whole domain Ω in a physically meaningful way, but for now, let us focus on understanding the error created with this approximation for $x = \lambda_i$. More specifically, we have

$$|v_T(\lambda_i)| = \left| \frac{T(\lambda_i) - \lambda_i}{\Delta t} \right| \leq \frac{|\lambda_i - \mu_i|}{\Delta t} + \frac{|T(\lambda_i) - \mu_i|}{\Delta t}. \tag{3.2}$$

We will therefore devote the rest of this chapter to the analysis of the behavior of $|T(\lambda_i) - \mu_i|$ with respect to the different parameters involved in the model (3.1).

3.2 Behavior of the Transport Map

3.2.1 Case of One 1D Particle

We first consider the simpler case where there is only one particle evolving in the flow. This situation will give us insight on what to expect from the error $|T(\lambda) - \mu|$

for a particle far from other particles. This can happen for example when the seeding density is low in a region of Ω. We begin by analyzing this situation in dimension one because in this case, the Monge-Ampère equation (2.5) can be solved explicitly to obtain an analytical expression for the OT map T. We consider for simplicity $\Omega = [0, 1]$. The model (3.1) reduces to

$$f(x) = \frac{r}{k_\lambda} e^{-\frac{(x-\lambda)^2}{2\sigma^2}} + (1-r), \quad g(x) = \frac{r}{k_\mu} e^{-\frac{(x-\mu)^2}{2\sigma^2}} + (1-r), \tag{3.3}$$

where

$$k_\lambda = \int_0^1 e^{-\frac{(t-\lambda)^2}{2\sigma^2}} \mathrm{d}t, \quad k_\mu = \int_0^1 e^{-\frac{(t-\mu)^2}{2\sigma^2}} \mathrm{d}t, \tag{3.4}$$

λ, μ being two points in Ω and $r \in [0, 1]$. Our technique consists in approximating the velocity vector at λ by $(T(\lambda) - \lambda)/\Delta t$, and thus we would hope that $T(\lambda)$ would be equal to μ. However, the parameters σ and r induce an error that sends $T(\lambda)$ off the target μ. In fact, we have the following theorem [40].

Theorem 3.2.1. *Assume $2/3 < r \leq 1$, $0 < \epsilon \leq \lambda \leq \mu \leq 1 - \epsilon$ for $0 < \epsilon < 0.5$, and $0 < \sigma < 1$. If T is the optimal map solving the one-dimentional L^2 transport problem with densities f and g given by* (3.3)*, then*

$$0 \leq \mu - T(\lambda) \leq C_1 \sigma \frac{(1-r)}{r}(\mu - \lambda) + C_2 \frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\sigma}(\mu - \lambda), \tag{3.5}$$

where C_1 and C_2 are positive constants independent of σ, λ, μ, r, and ϵ. Moreover, if all other parameters are fixed, $\mu - T(\lambda)$ does not decrease faster than linearly when $\sigma \to 0$.

Before we present the proof of Theorem 3.2.1, we state and prove the following technical lemma [40].

Lemma 3.2.2. *Under the assumptions of Theorem 3.2.1, there exists a constant C_2 independent of σ, λ, μ, r, and ϵ, such that*

$$0 \leq \frac{k_\mu \left(\frac{1}{k_\mu} \int_0^\mu e^{-\frac{(t-\mu)^2}{2\sigma^2}} \, dt - \frac{1}{k_\lambda} \int_0^\lambda e^{-\frac{(t-\lambda)^2}{2\sigma^2}} \, dt \right)}{k_\mu \frac{(1-r)}{r} + \int_0^1 e^{-\frac{t^2 (T(\lambda)-\mu)^2}{2\sigma^2}} \, dt} \leq C_2 \frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\sigma}(\mu - \lambda), \tag{3.6}$$

where k_λ and k_μ are given in (3.4)*.*

Proof of Lemma 3.2.2. If $\lambda = \mu$, the result is trivial. Assume $\lambda < \mu$. We introduce the function

$$G(\gamma) = \frac{\int_0^\gamma e^{-\frac{(t-\gamma)^2}{2\sigma^2}}\,\mathrm{d}t}{\int_0^1 e^{-\frac{(t-\gamma)^2}{2\sigma^2}}\,\mathrm{d}t}.$$

We observe that its derivative,

$$G'(\gamma) = \frac{e^{-\frac{\gamma^2}{2\sigma^2}}\int_\gamma^1 e^{-\frac{(t-\gamma)^2}{2\sigma^2}}\,\mathrm{d}t + e^{-\frac{(\gamma-1)^2}{2\sigma^2}}\int_0^\gamma e^{-\frac{(t-\gamma)^2}{2\sigma^2}}\,\mathrm{d}t}{\big(\int_0^1 e^{-\frac{(t-\gamma)^2}{2\sigma^2}}\,\mathrm{d}t\big)^2},$$

is always positive, which means that G is an increasing function. By the mean-value theorem,

$$\exists\,\xi:\ \lambda \le \xi \le \mu \quad \text{such that} \quad G(\mu) - G(\lambda) = G'(\xi)\,(\mu - \lambda).$$

Moreover, because $\lambda, \mu \in [\epsilon, 1-\epsilon]$, we can bound G' in the following way:

$$G'(\xi) \le \frac{e^{-\frac{\epsilon^2}{2\sigma^2}}\int_\xi^1 e^{-\frac{(t-\xi)^2}{2\sigma^2}}\,\mathrm{d}t + e^{-\frac{\epsilon^2}{2\sigma^2}}\int_0^\xi e^{-\frac{(t-\xi)^2}{2\sigma^2}}\,\mathrm{d}t}{\big(\int_0^1 e^{-\frac{(t-\xi)^2}{2\sigma^2}}\,\mathrm{d}t\big)^2} = \frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\int_0^1 e^{-\frac{(t-\xi)^2}{2\sigma^2}}\,\mathrm{d}t}.$$

Combining the above estimate and the fact that the optimal map T sends $[0,1]$ to $[0,1]$, we have

$$\begin{aligned}
\frac{k_\mu(G(\mu) - G(\lambda))}{k_\mu\frac{(1-r)}{r} + \int_0^1 e^{-\frac{t^2(T(\lambda)-\mu)^2}{2\sigma^2}}\,\mathrm{d}t} &\le \frac{\int_0^1 e^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t}{\int_0^1 e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t}\frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\int_0^1 e^{-\frac{(t-\xi)^2}{2\sigma^2}}\,\mathrm{d}t}(\mu-\lambda)\\
&\le \frac{\int_0^1 e^{-\frac{(t-1/2)^2}{2\sigma^2}}\,\mathrm{d}t}{\int_0^1 e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t}\frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\int_0^1 e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t}(\mu-\lambda)\\
&= \frac{\int_{-1/2}^{1/2} e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t}{\int_0^1 e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t}\frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\sigma\int_0^{1/\sigma} e^{-\frac{t^2}{2}}\,\mathrm{d}t}\,(\mu-\lambda),
\end{aligned}$$

where in the last line we performed two changes of variables. Finally, from the symmetry of the Gaussian distribution and the assumption $0 < \sigma < 1$, we get

$$\frac{\int_{-1/2}^{1/2} e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t}{\int_0^1 e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t}\frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\sigma\int_0^{1/\sigma} e^{-\frac{t^2}{2}}\,\mathrm{d}t}\,(\mu-\lambda) \le 2\frac{1}{\int_0^1 e^{-\frac{t^2}{2}}\,\mathrm{d}t}\frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\sigma}\,(\mu-\lambda).$$

The inequality (3.6) follows with $C_2 = 2/\int_0^1 e^{-\frac{t^2}{2}}\,\mathrm{d}t$. □

Proof of Theorem 3.2.1. We divide the proof into three main parts. First, we will show that

$$0 \le \mu - T(\lambda) \le K(\sigma)\frac{(1-r)}{r}(\mu-\lambda) + C_2\frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\sigma}(\mu-\lambda), \tag{3.7}$$

where $K(\sigma)$ is a function of σ which is bounded by 2, and C_2 is a constant. Second, we will prove that there exists a constant C_1 such that $K(\sigma) \le C_1\sigma$, and finally, we will show that $\mu - T(\lambda)$ cannot decay faster than linearly in σ.

To prove the first claim, we introduce the cumulative distribution functions for $f(x)$ and $g(x)$,

$$\begin{aligned} F(x) &= \frac{r}{k_\lambda}\int_0^x e^{-\frac{(t-\lambda)^2}{2\sigma^2}}\,\mathrm{d}t + (1-r)x, \\ G(x) &= \frac{r}{k_\mu}\int_0^x e^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t + (1-r)x. \end{aligned} \tag{3.8}$$

Because the solution of the L^2 transport problem is the gradient of a convex function (which in dimension one corresponds to a function with nonnegative second derivative), in this simple 1D case the Monge-Ampère equation (2.5) becomes

$$g(\Psi'(x))\Psi''(x) = f(x),$$

which integrates to

$$T(x) \equiv \Psi'(x) = G^{-1}(G(T(0)) + F(x)).$$

Now, because the optimal map T is increasing and invertible, we have $T(0) = 0$ and $G(0) = 0$, so $T(x) = G^{-1}(F(x))$. We want to estimate $\mu - T(\lambda) = \mu - G^{-1}(F(\lambda))$. Let $y = T(\lambda)$. According to (3.8), we have $G(y) = F(\lambda)$, i.e.,

$$\frac{1}{k_\mu}\int_0^y e^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t + \frac{(1-r)}{r}y = \frac{1}{k_\lambda}\int_0^\lambda e^{-\frac{(t-\lambda)^2}{2\sigma^2}}\,\mathrm{d}t + \frac{(1-r)}{r}\lambda. \tag{3.9}$$

Next, we consider

$$N(y) = \frac{1}{k_\mu}\int_0^y e^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t,$$

and compute its Taylor expansion about μ with integral remainder:

$$
\begin{aligned}
N(y) &= N(\mu) + N'(\mu)(y-\mu) + (y-\mu)^2 \int_0^1 (1-t) N''(\mu + t(y-\mu))\,\mathrm{d}t \\
&= \frac{1}{k_\mu}\int_0^\mu e^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t + \frac{(y-\mu)}{k_\mu} - \frac{(y-\mu)^3}{k_\mu \sigma^2}\int_0^1 t(1-t) e^{-\frac{t^2(y-\mu)^2}{2\sigma^2}}\,\mathrm{d}t \\
&= \frac{1}{k_\mu}\int_0^\mu e^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t + \frac{(y-\mu)}{k_\mu}\int_0^1 e^{-\frac{t^2(y-\mu)^2}{2\sigma^2}}\,\mathrm{d}t.
\end{aligned}
$$

Integration by parts is used to obtain the last equality. Factoring out $\mu - y$ in (3.9) and multiplying by k_μ yields

$$
\begin{aligned}
&\left(k_\mu \frac{(1-r)}{r} + \int_0^1 e^{-\frac{t^2(y-\mu)^2}{2\sigma^2}}\,\mathrm{d}t\right)(\mu - y) \\
&\quad = k_\mu \frac{(1-r)}{r}(\mu - \lambda) + k_\mu\left(\frac{1}{k_\mu}\int_0^\mu e^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t - \frac{1}{k_\lambda}\int_0^\lambda e^{-\frac{(t-\lambda)^2}{2\sigma^2}}\,\mathrm{d}t\right).
\end{aligned} \tag{3.10}
$$

From Lemma 3.2.2, we know that the right-hand side is nonnegative, which implies that the first inequality in (3.5) is satisfied. Moreover, because T maps $[0,1]$ to itself, we have $\mu - y \le 1$, and thus $\exp(-\frac{t^2}{2\sigma^2}) \le \exp(-\frac{t^2(y-\mu)^2}{2\sigma^2})$. We get

$$
\frac{k_\mu}{k_\mu \frac{(1-r)}{r} + \int_0^1 e^{-\frac{t^2(y-\mu)^2}{2\sigma^2}}\,\mathrm{d}t} \le \frac{\int_0^1 e^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t}{\int_0^1 e^{-\frac{t^2(y-\mu)^2}{2\sigma^2}}\,\mathrm{d}t} \le \frac{\int_0^1 e^{-\frac{(t-1/2)^2}{2\sigma^2}}\,\mathrm{d}t}{\int_0^1 e^{-\frac{t^2}{2\sigma^2}}\,\mathrm{d}t} \le 2, \tag{3.11}
$$

from which we conclude the existence of $K(\sigma)$ in (3.7). In addition, the second term on the right of (3.10) leads to the expression in Lemma 3.2.2, which in turn yields (3.7).

Second, we prove the linearity in σ. Let $z(\sigma) = \mu - T(\lambda)$. Notice that (3.9) can be rewritten as

$$
\frac{\int_{-\frac{\mu}{\sigma}}^{-\frac{z(\sigma)}{\sigma}} e^{-\frac{t^2}{2}}\,\mathrm{d}t}{\int_{-\frac{\mu}{\sigma}}^{\frac{(1-\mu)}{\sigma}} e^{-\frac{t^2}{2}}\,\mathrm{d}t} + \frac{(1-r)}{r}(\mu - \lambda) = \frac{\int_{-\frac{\lambda}{\sigma}}^{0} e^{-\frac{t^2}{2}}\,\mathrm{d}t}{\int_{-\frac{\lambda}{\sigma}}^{\frac{(1-\lambda)}{\sigma}} e^{-\frac{t^2}{2}}\,\mathrm{d}t} + \frac{(1-r)}{r} z(\sigma). \tag{3.12}
$$

We proceed by contradiction. Assume that for every $n \in \mathbb{N}$, there exists a σ_n such that $z(\sigma_n) > n\sigma_n$. Because $z(\sigma) < 1$ for every σ, the sequence $1/\sigma_n$ is bounded below by n and therefore $1/\sigma_n \to \infty$ as $n \to \infty$. Inserting σ_n in (3.12), taking the limit with respect to n, and using the fact that both λ and μ are bounded away from 0

and 1, respectively, yields

$$\frac{(1-r)}{r}(\mu-\lambda)=\frac{1}{2}+\frac{(1-r)}{r}\lim_{n\to\infty}z(\sigma_n)\geq\frac{1}{2}.$$

The last inequality results from $z(\sigma)$ being nonnegative for every σ, as shown in (3.7). Because $\mu-\lambda$ is smaller than 1, we get

$$\frac{(1-r)}{r}\geq\frac{1}{2},$$

which is a contradiction for $r>2/3$. We conclude the existence of a positive constant C_1 such that $K(\sigma)\leq C_1\sigma$.

Finally to prove that $z(\sigma)=\mu-T(\lambda)$ cannot decay faster than linearly in σ, we proceed again by contradiction. Indeed, if we assume $\lim_{\sigma\to 0}z(\sigma)/\sigma=0$, and take the limit with respect to σ on both sides of (3.12), we get

$$\frac{1}{2}+\frac{(1-r)}{r}(\mu-\lambda)=\frac{1}{2},$$

which yields another contradiction for fixed $r\neq 1$ and $\lambda\neq\mu$. □

Observe that for any fixed $\epsilon>0$, the decay of the second term in (3.5) is always going to be exponential when σ goes to 0. However, if one wants to vary ϵ as a function of σ, then we need to take $\epsilon\geq\sigma^{1-\alpha}$ for $\alpha>0$ to preserve this exponential decay. The result given by Theorem 3.2.1 provides information on the quality of the approximation of μ by $T(\lambda)$. Indeed, we see that if $1-r\neq 0$, $\sigma\neq 0$ and of course if $\lambda\neq\mu$, then the center of the initial Gaussian is mapped to a point located before the center of the target Gaussian and the resulting vector $v_T(\lambda)$ is smaller in magnitude than the target vector at that point. This is also the case when $1-r=0$, $\sigma>0$ and $\mu\neq\lambda$, but the error is negligible for small σ.

3.2.2 Case of One 2D Particle

In higher dimensions, it is not possible to obtain the transport map explicitly, which makes it harder to rigorously derive error bounds. Nonetheless, we will be able to present in this section convincing arguments on why we expect the same error behavior as in the one-dimensional case. For the case of one particle in 2D evolving

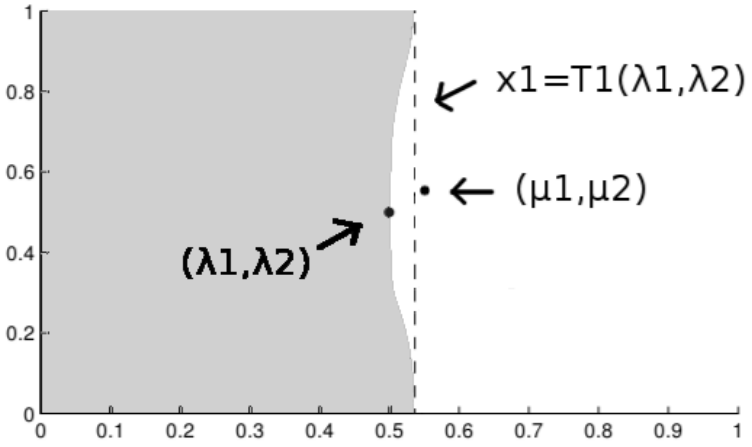


Figure 3.1: Illustration of the sets E_1 and S_1. The set E_1 corresponds to the part of $[0,1]^2$ which is on the left of the dashed line $x_1 = T_1(\lambda_1, \lambda_2)$ whereas the set S_1 corresponds to the shaded region. The initial position (λ_1, λ_2) and the final position (μ_1, μ_2) of the particle are also indicated.

in $\Omega = [0,1]^2$, the densities f, g are given by

$$\begin{aligned} f(x_1,x_2) =& \frac{r}{k_\lambda}\exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\lambda_1)^2+(x_2-\lambda_2)^2\big]\Big) + (1-r), \\ g(x_1,x_2) =& \frac{r}{k_\mu}\exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\mu_1)^2+(x_2-\mu_2)^2\big]\Big) + (1-r), \end{aligned} \tag{3.13}$$

where

$$k_\lambda = \int_{[0,1]^2} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\lambda_1)^2+(x_2-\lambda_2)^2\big]\Big)\mathrm{d}x$$

and k_μ is defined similarly. Here, (λ_1, λ_2) and (μ_1, μ_2) denote the initial and final positions of the particle. Let $T(x_1, x_2) = (T_1(x_1, x_2), T_2(x_1, x_2))$ be the Optimal Transport map associated with f and g. We introduce the sets

$$\begin{aligned} E_1 =& \{(x_1,x_2) \in [0,1]^2 : x_1 \le T_1(\lambda_1,\lambda_2)\} \\ \text{and} \qquad S_1 = T^{-1}(E_1) =& \{(x_1,x_2) \in [0,1]^2 : T_1(x_1,x_2) \le T_1(\lambda_1,\lambda_2)\}. \end{aligned}$$

These sets will be useful to recover an analogue of our one-dimensional arguments for the first coordinate of the OT map (the same arguments will apply as well for the second coordinate). Before we see how, let us look at the illustration in Figure

3.1. Similar to the one-dimensional case, we expect the first coordinate of the image of (λ_1, λ_2) to be on the left of μ_1 somewhere along the dashed line $x_1 = T_1(\lambda_1, \lambda_2)$, before the line $x_1 = \mu_1$. The set E_1 corresponds in this case to the part of the domain to the left of this dashed line. Now, if we consider the set of all points for which the first coordinate of the transport map is smaller than this dashed line, which we called S_1, then we expect to get a region similar to the shaded region in Figure 3.1. Indeed, close to the center of the particle, the optimal map should move the mass towards the final position (μ_1, μ_2) of the particle. However, for points further than several standard deviations to the center, the mass should stay roughly at the same position. Now, going back to our analysis, by the conservation of mass constraint (2.3), taking $E = E_1$, we have

$$\int_0^1 \int_0^{T_1(\lambda_1,\lambda_2)} g(x_1, x_2)\mathrm{d}x_1\mathrm{d}x_2 = \int\int_{S_1} f(x_1, x_2)\mathrm{d}x.$$

This in turn yields

$$\begin{aligned} &\frac{1}{k_\mu}\int_0^1 \int_0^{T_1(\lambda_1,\lambda_2)} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\mu_1)^2 + (x_2-\mu_2)^2\big]\Big)\mathrm{d}x_1\mathrm{d}x_2 \\ &\quad + \frac{(1-r)}{r}T_1(\lambda_1,\lambda_2) \\ =&\frac{1}{k_\lambda}\int\int_{S_1} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\lambda_1)^2 + (x_2-\lambda_2)^2\big]\Big)\mathrm{d}x + \frac{(1-r)}{r}\mathrm{m}(S_1). \end{aligned} \tag{3.14}$$

Here, $\mathrm{m}(S_1)$ denotes the Lebesgue measure of the set S_1. Observe that (3.14) is very close to (3.9), the equality we used in the one-dimensional proof. Then, we can rewrite (3.14) in a similar way as in (3.10) by using a Taylor expansion. We get

$$\begin{aligned} &(\mu_1 - T_1(\lambda_1,\lambda_2))\Bigg[\frac{(1-r)}{r} \\ &\qquad + \frac{1}{k_\mu}\int_0^1\int_0^1 \exp\Big(-\frac{1}{2\sigma^2}\big[t^2(T_1(\lambda_1,\lambda_2)-\mu_1)^2 + (x_2-\mu_2)^2\big]\Big)\mathrm{d}t\mathrm{d}x_2\Bigg] \\ &= \frac{1}{k_\mu}\int_0^1\int_0^{\mu_1} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\mu_1)^2 + (x_2-\mu_2)^2\big]\Big)\mathrm{d}x_1\mathrm{d}x_2 \\ &\quad - \frac{1}{k_\lambda}\int\int_{S_1} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\lambda_1)^2 + (x_2-\lambda_2)^2\big]\Big)\mathrm{d}x \\ &\quad + \frac{(1-r)}{r}\big[\mu_1 - \mathrm{m}(S_1)\big]. \end{aligned} \tag{3.15}$$

Before we pursue the analysis further, we shall investigate the properties of the set S_1. We have the following lemma [40].

Lemma 3.2.3. *There exists a continuous function* $h_1 : [0,1] \to [0,1]$ *such that* $h_1(\lambda_2) = \lambda_1$ *and*

a) $T\big\{(x_1,x_2) \in \Omega : x_1 = h_1(x_2)\big\} = \big\{(x_1,x_2) \in \Omega : x_1 = T_1(\lambda_1,\lambda_2)\big\}$,

b) $S_1 = \displaystyle\bigcup_{x_2 \in [0,1]} \Big([0, h_1(x_2)] \times \{x_2\} \Big)$.

Prior to proving this lemma, we state and prove the following intermediate result [40].

Lemma 3.2.4. *Let* $T = (T_1, T_2)$ *be the unique solution to the optimal transport problem* (2.2) *on* $\Omega = [0,1]^2$ *with densities* f *and* g *bounded away from* 0 *and at least* $C^{0,\alpha}(\Omega)$ *for* $0 < \alpha < 1$. *Then* T *maps every side of the boundary to itself. More precisely,* $T_2(x_1, 0) = 0$, $T_2(x_1, 1) = 1$ *for every* $x_1 \in [0,1]$, *and* $T_1(0, x_2) = 0$, $T_1(1, x_2) = 1$ *for every* $x_2 \in [0,1]$. *As a consequence,* T *maps every corner of the square* $[0,1]^2$ *to itself.*

Proof of Lemma 3.2.4. Recall that $T = \nabla\Psi$ where Ψ is a convex function. Because f and g are positive densities bounded away from 0, by the Monge-Ampère equation (2.5), $\det(D^2\Psi) > 0$ and thus Ψ is strictly convex. This yields

$$(T(x) - T(y)) \cdot (x - y) = (\nabla\Psi(x) - \nabla\Psi(y)) \cdot (x - y) > 0, \tag{3.16}$$

for $x \neq y$, $x, y \in [0,1]^2$. We first proceed by contradiction to show that $T_1(0, x_2) = 0$. Assume there is a point $(0, x_2) \in [0,1]^2$ for which $T(0, x_2) = (y_1, y_2)$ and $y_1 > 0$. Then, because the optimal map T is invertible (hence bijective), there exists another point $(z_1, z_2) \in [0,1]^2$ which gets mapped to $(0, y_2)$. Using these two points in (3.16), we get

$$(T(0,x_2) - T(z_1,z_2)) \cdot ((0,x_2) - (z_1,z_2)) = (y_1, 0) \cdot (-z_1, x_2 - z_2) = -y_1 z_1 > 0,$$

which yields a contradiction because $y_1 z_1 \geq 0$ by assumption. We conclude that $T_1(0, x_2) = 0$ for every $x_2 \in [0,1]$. Using similar arguments, we can prove that the other sides of the square get mapped to themselves. □

Proof of Lemma 3.2.3. Consider the line $\mathcal{L} = \{(x_1, x_2) \in [0,1]^2 : x_1 = T_1(\lambda_1, \lambda_2)\}$. Because T is continuous, the inverse image of that line by T has to be a continuous

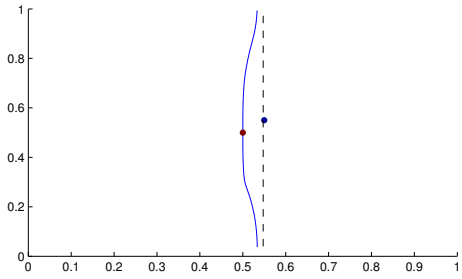
curve lying in $[0,1]^2$. Denote this curve by $\mathcal{C}$. From Lemma 3.2.4 and by the bijectivity of T, we have that the inverse image of $(T_1(\lambda_1,\lambda_2),1)$ is on the line $[0,1]\times\{1\}$ and the inverse image of $(T_1(\lambda_1,\lambda_2),0)$ is on the line $[0,1]\times\{0\}$. Therefore, by continuity, for every $x_2 \in [0,1]$ there exists a $x_1 \in [0,1]$ such that $T(x_1,x_2) \in \mathcal{L}$. Moreover, thanks to (3.16), for every fixed $x_2 \in [0,1]$, $T_1(x_1,x_2)$ is a strictly increasing function of x_1. This implies that for any $x_2 \in [0,1]$, there is a unique $x_1 \in [0,1]$ such that $(x_1,x_2) \in \mathcal{C}$, which in turn proves the existence of a continuous function h_1 of x_2 such that $C = \{(x_1,x_2) \in [0,1]^2 : x_1 = h_1(x_2)\}$, that is, $\mathcal{C}$ is the graph of h_1. In addition, when $x_2 = \lambda_2$, $T_1(h(\lambda_2),\lambda_2) = T_1(\lambda_1,\lambda_2)$. This yields $h_1(\lambda_2) = \lambda_1$ because $x_1 \mapsto T_1(x_1,\lambda_2)$ is a bijection on $[0,1]$, and statement a) is thus verified. For b), we have

$$\begin{aligned} S_1 &= \{(x_1,x_2) \in [0,1]^2 : T_1(0,x_2) \le T_1(x_1,x_2) \le T_1(\lambda_1,\lambda_2)\} \\ &= \{(x_1,x_2) \in [0,1]^2 : T_1(0,x_2) \le T_1(x_1,x_2) \le T_1(h_1(x_2),x_2)\} \\ &= \{(x_1,x_2) \in [0,1]^2 : 0 \le x_1 \le h_1(x_2)\} \\ &= \bigcup_{x_2\in[0,1]} \Big([0,h_1(x_2)] \times \{x_2\} \Big), \end{aligned}$$

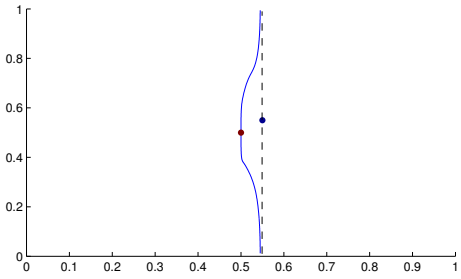
because $0 = T_1(0,x_2)$. This concludes the proof. □

Note that by symmetry, we can derive similar results for μ_2, $T_2(\lambda_1,\lambda_2)$, and the set $S_2 = \{(x_1,x_2) \in [0,1]^2 : T_2(x_1,x_2) \le T_2(\lambda_1,\lambda_2)\}$ with the function h_1 replaced by its analogue h_2.

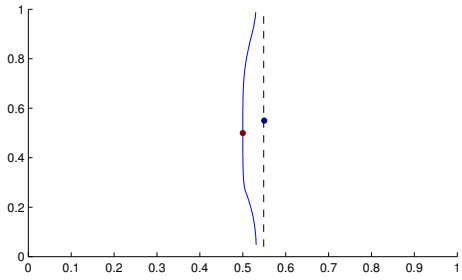
In order to better understand the behavior of the function h_1, and thus the set S_1, we present in Figure 3.2 some examples of h_1 for different values of the parameters. These results were obtained by using the algorithm described in Section 4.1 to find the optimal map T associated with the densities f and g given by (3.13). From Figure 3.2, we observe that for values of x_2 close to λ_2, h_1 is essentially a vertical line, which means that T acts as a translation in a neighborhood of λ_2. The length of this neighborhood depends on the standard deviation σ and seems to suggest that the map T is a translation up to the point where the part of the mass that is due to the particle becomes negligible. In addition, outside this neighborhood of λ_2, the function h_1 appears to converge to another vertical line as x_2 goes to 0 or 1, and this line is close to the line $x_1 = T(\lambda_1,\lambda_2)$. Note that this line lays very close to (and before) $x_1 = \mu_1$. This means that the map T is almost the identity map when taking



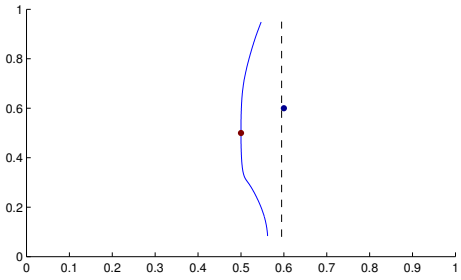
(a) $\lambda_1 = \lambda_2 = 0.5$, $\mu_1 = \mu_2 = 0.55$, $\sigma = 1/12$ and $r = 0.8$.



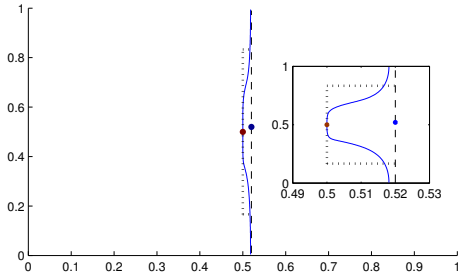
(b) Same parameters as a), but $\sigma = 1/24$.



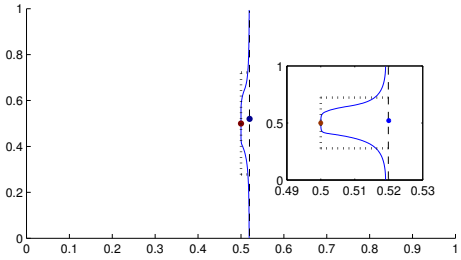
(c) Same parameters as a), but $r = 0.9$.



(d) Same parameters as a), but $\mu_1 = \mu_2 = 0.6$.



(e) $\lambda_1 = \lambda_2 = 0.5$, $\mu_1 = \mu_2 = 0.52$, $\sigma = 1/24$ and $r = 0.8$.



(f) Same parameters as e), but $\sigma = 1/36$.

Figure 3.2: Some examples of the function h_1, which is represented here by the solid line. The vertical axis is x_2 and the horizontal axis is x_1. The dots represent the initial and final position of the particle. The dashed lines represent $x_1 = T_1(\lambda_1, \lambda_2)$. For e) and f), the vertical dotted line is the line connecting $\lambda_2 - 8\sigma$ to $\lambda_2 + 8\sigma$ and the two horizontal dotted lines link the bottom and top of the previous line segment to the line $x_1 = T_1(\lambda_1, \lambda_2)$. In addition, for e) and f), a subplot is displayed to better visualize the figure.

points for which the Gaussian contribution to the densities f and g is negligible. This confirms the intuition that the optimal map translates the mass around the particle's location and leaves the rest of the domain (almost) unchanged. Accordingly, the plots

in Figure 3.2 suggest the following assertions [40]:

1) $\mu_1 \geq T_1(\lambda_1, \lambda_2)$,

2) $h_1(x_2) \geq \lambda_1$ for any $x_2 \in [0, 1]$,

3) There exist constants $C > 0$ and $\sigma_0 > 0$ such that $C\sigma_0 \leq \min\{\lambda_2, 1 - \lambda_2\}$, and for every $\sigma < \sigma_0$, we have

$$T_1(\lambda_1, \lambda_2) - 2C\sigma(T_1(\lambda_1, \lambda_2) - \lambda_1) \leq \mathrm{m}(S_1).$$

The first two assertions are clear in plots a) to f) of Figure 3.2, whereas the third one is a bit more technical and in order to better visualize it we included the dotted lines in plots e) and f). This third property actually states that the Lebesgue measure of the set

$$\tilde{S}_1 = \Big\{ (x_1, x_2) \in [0, 1]^2 : 0 \leq x_1 \leq \lambda_1 \text{ if } \lambda_2 - C\sigma \leq x_2 \leq \lambda_2 + C\sigma \\ \text{and } 0 \leq x_1 \leq T_1(\lambda_1, \lambda_2) \text{ elsewhere} \Big\}$$

is smaller than $\mathrm{m}(S_1)$. Therefore, the vertical dotted line in e) and f) corresponds to the interval $[\lambda_2 - C\sigma, \lambda_2 + C\sigma]$ in x_2, where we took $C = 8$ as an example. We observe that in both cases, $\mathrm{m}(\tilde{S}_1)$ is indeed smaller than $\mathrm{m}(S_1)$. In the limiting case where $\sigma \to 0$, that is when the particle becomes a point mass (or Dirac delta function), the optimal map only moves the mass at the particle's location and thus assertion 3 is trivially satisfied. Despite all these observations in favor of assertions 1,2 or 3, obtaining a rigorous proof showing that they actually hold true for σ and $1 - r$ small enough remains an open problem. However, intuitively we argue that all these assertions are reasonable, because of the optimality of the transport map T. Moreover, we proved in the previous section that assertion 1 holds in dimension one, whereas 2 and 3 are trivial in this case.

Let us now show that, assuming assertions 1 and 2 hold, we can recover the linear behavior in dimension two for the error with respect to $(1 - r)/r$, as in Theorem 3.2.1. Moreover, if in addition we assume that assertion 3 holds, we recover the linear behavior in σ. We have the following theorem [40].

Theorem 3.2.5. *Assume* $2/3 < r \leq 1$, $0 < \epsilon \leq \lambda_1 \leq \mu_1 \leq 1 - \epsilon$ *and* $0 < \epsilon \leq \lambda_2 \leq \mu_2 \leq 1 - \epsilon$ *for* $0 < \epsilon < 0.5$, *and* $0 < \sigma < 1$. *Let* $T = (T_1, T_2)$ *be the optimal map solving the two-dimensional* L^2 *transport problem with densities* f *and* g *given by* (3.13). *Suppose in addition that assertions 1 and 2 hold. Then we have the following inequality:*

$$\mu_1 - T_1(\lambda_1, \lambda_2) \leq C_1(\sigma)\frac{(1-r)}{r}(\mu_1 - \lambda_1) + C_2\frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\sigma}(\mu_1 - \lambda_1), \tag{3.17}$$

where C_1 *is a positive function of* σ *which is bounded by 2, and* C_2 *is a positive constant. Both* C_1 *and* C_2 *are independent of* $\lambda_1, \lambda_2, \mu_1, \mu_2, r$ *and* ϵ. *If in addition we assume that 3 holds, then*

$$C_1(\sigma) \leq 4C\sigma, \tag{3.18}$$

where C *is the same constant as in assumption 3. A similar result holds for* $\mu_2 - T_2(\lambda_1, \lambda_2)$.

Proof. We prove the result for $\mu_1 - T_1(\lambda_1, \lambda_2)$; by symmetry, similar arguments can be used for $\mu_2 - T_2(\lambda_1, \lambda_2)$. The starting point is equation (3.15). By assumption 2, we have $\lambda_1 \leq h(x_2)$ for any $x_2 \in [0, 1]$ so that

$$\begin{aligned}
\int_0^1 \int_0^{\lambda_1} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1 - \lambda_1)^2 + (x_2 - \lambda_2)^2\big]\Big)\mathrm{d}x_1\,\mathrm{d}x_2 \\
&\leq \int_0^1 \int_0^{h_1(x_2)} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1 - \lambda_1)^2 + (x_2 - \lambda_2)^2\big]\Big)\mathrm{d}x_1\,\mathrm{d}x_2 \\
&= \int\int_{S_1} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1 - \lambda_1)^2 + (x_2 - \lambda_2)^2\big]\Big)\mathrm{d}x_1\,\mathrm{d}x_2.
\end{aligned}$$

Moreover, using Lemma 3.2.3 and assumption 2, we have $[0, \lambda_1] \times [0, 1] \subset S_1$ and thus $\lambda_1 \leq \mathrm{m}(S_1)$. Combining these estimates with the fact that both k_μ and k_λ can be written as the product of two integrals, (3.15) implies

$$\begin{aligned}
\mu_1 - T_1(\lambda_1, \lambda_2) \leq \Bigg[\frac{(1-r)}{r}\big(\mu_1 - \lambda_1\big) + \frac{1}{k_{\mu_1}}\int_0^{\mu_1} \exp\Big(-\frac{1}{2\sigma^2}(x_1 - \mu_1)^2\Big)\mathrm{d}x_1 \\
- \frac{1}{k_{\lambda_1}}\int_0^{\lambda_1} \exp\Big(-\frac{1}{2\sigma^2}(x_1 - \lambda_1)^2\Big)\mathrm{d}x_1\Bigg]k_{\mu_1} \\
\Bigg(k_{\mu_1}\frac{(1-r)}{r} + \int_0^1 \exp\big(-\frac{1}{2\sigma^2}t^2(T_1(\lambda_1, \lambda_2) - \mu_1)^2\big)\mathrm{d}t\Bigg)^{-1}
\end{aligned} \tag{3.19}$$

where k_{λ_1} and k_{μ_1} are defined by respectively replacing λ and μ by λ_1 and μ_1 in (3.4). Notice now that we can split the right-hand side of (3.19) into two terms, one for which bound (3.6) in Lemma 3.2.2 applies, and one for which the arguments displayed in (3.11) apply. This yields the existence of C_1 and C_2, as claimed. Finally, if in addition we assume that assumption 3 holds, then

$$\begin{aligned}\mu_1 - m(S_1) &\leq \mu_1 - T_1(\lambda_1, \lambda_2) + (T_1(\lambda_1, \lambda_2) - \lambda_1)2C\sigma \\ &\leq \mu_1 - T_1(\lambda_1, \lambda_2) + (\mu_1 - \lambda_1)2C\sigma.\end{aligned}$$

Using this in (3.15), we cancel the terms $(\mu_1 - T_1(\lambda_1, \lambda_2))(1-r)/r$ on both sides to obtain

$$\begin{aligned}&(\mu_1 - T_1(\lambda_1, \lambda_2)) \\ &\qquad \times \frac{1}{k_\mu}\int_0^1\int_0^1 \exp\Big(-\frac{1}{2\sigma^2}\big[t^2(T_1(\lambda_1,\lambda_2)-\mu_1)^2 + (x_2-\mu_2)^2\big]\Big)\mathrm{d}t\,\mathrm{d}x_2 \\ &\leq \frac{1}{k_\mu}\int_0^1\int_0^{\mu_1} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\mu_1)^2 + (x_2-\mu_2)^2\big]\Big)\mathrm{d}x_1\mathrm{d}x_2 \\ &\quad - \frac{1}{k_\lambda}\int\int_{S_1} \exp\Big(-\frac{1}{2\sigma^2}\big[(x_1-\lambda_1)^2 + (x_2-\lambda_2)^2\big]\Big)\mathrm{d}\mathbf{x} \\ &\quad + \frac{(1-r)}{r}(\mu_1-\lambda_1)2C\sigma.\end{aligned}$$

We see that the previous inequality is very close to (3.19), provided we remove the term $k_{\mu_1}(1-r)/r$ in the denominator. Even if this term is not removed, we can still get the same upper bounds as in Lemma 3.2.2 and as in the arguments in (3.11), which yields the result. □

We will present in Section 4.2 numerical simulations validating the linear behavior with respect to $(1-r)/r$ and σ, for the case where we consider multiple particles in dimension two. The results will be similar to the results in the one-dimensional case presented in Section 3.2.1. Finally, it is worth mentioning that even though we restricted ourselves to dimension 2, all the arguments presented can be employed to get the same results in any dimension.

3.2.3 Case of Two Particles Within the Tracking Range

Thanks to the previous analysis, we now understand better the effect of r and σ on the error $|T(\lambda) - \mu|$. To analyze the effect of the weights m_i, we consider in this section the scenario in which two particles are evolving in the fluid. Just as in Section 3.2.1, we look at the one-dimensional case so that the transport map can be obtained explicitly. Therefore, let the domain be $\Omega = [0, 1]$ and the centers of the particles be respectively at λ_1 and λ_2. Let the particles evolve and arrive at positions μ_1 and μ_2 in $\Omega = [0, 1]$ after Δt units of time. We first consider the case for which the particles stay within the tracking range, that is λ_1, μ_1 are close to each other and far from λ_2, μ_2 so that the OT map may not mistakenly select μ_2 as the image of λ_1 and vice-versa. Let the initial mass density be given by

$$f(x) = \frac{r}{m_1 + m_2} \sum_{i=1}^{2} \frac{m_i}{k_{\lambda_i}} e^{\frac{-(x-\lambda_i)^2}{2\sigma^2}} + (1 - r), \; k_{\lambda_i} = \int_0^1 e^{\frac{-(x-\lambda_i)^2}{2\sigma^2}} \, \mathrm{d}x, \quad i = 1, 2.$$

Similarly, define the final density g by replacing λ_i by μ_i in the definition of f. We will establish an estimate for the ratio of the corresponding transport errors, $|T(\lambda_1) - \mu_1|/|T(\lambda_2) - \mu_2|$, in terms of the particle masses m_1 and m_2 [41]. Note that in practice, the weights m_1 and m_2 in the initial distribution f may be different from the ones in the final distribution g. However, it is not unrealistic to assume that the weight differences between the initial and final densities remain small if Δt is small enough, which explains the equality assumption. This in principle does not depend on whether or not the particles are inside or outside the tracking range as long as the flow considered is laminar enough (a necessary assumption for 2D-PIV).

Theorem 3.2.6. [41] Let T be the OT solution mapping the previously defined one-dimensional density f to its target g. Fix $0 < r < 1$, $0 < \epsilon < \lambda_1, \mu_1 < \lambda_2, \mu_2 < 1 - \epsilon < 1$ for $\epsilon > 0$, where $\lambda_1 \neq \mu_1$, $\lambda_2 \neq \mu_2$ and

$$\max_{i=1,2} |\lambda_i - \mu_i| < \min \left(\min_{1 \leq i \neq j \leq 2} |\lambda_i - \mu_j|, |\lambda_1 - \lambda_2|, |\mu_1 - \mu_2| \right) \tag{3.20}$$

so that the particles remain far from each other. Assume that the weights $m_1, m_2 > 0$. Then, for $\sigma > 0$ small enough, we have

$$\frac{|\mu_1 - T(\lambda_1)|}{|\mu_2 - T(\lambda_2)|} = \mathcal{O}\left(\frac{m_2}{m_1}\right).$$

Proof. The beginning of this proof is similar to the case of one particle in dimension one and thus we will simply summarize the initial arguments. Let $T(x)$ be the L^2 OT map sending f to g. Then $T : [0,1] \to [0,1]$ is increasing, bijective, and satisfies $g(T(x))T'(x) = f(x)$. Integrating this relation over $[0,x]$, we get $G(T(x)) - G(T(0)) = F(x)$, where F and G are the cumulative distributions functions of f and g:

$$F(x) = \frac{1}{m_1+m_2}\left(\frac{m_1\, r}{k_{\lambda_1}}\int_0^x e^{-\frac{(t-\lambda_1)^2}{2\sigma^2}}\,\mathrm{d}t + \frac{m_2\, r}{k_{\lambda_2}}\int_0^x e^{-\frac{(t-\lambda_2)^2}{2\sigma^2}}\,\mathrm{d}t\right) + (1-r)x,$$

and $G(x)$ is defined similarly. This implies that $G(T(0)) = G(0) = 0$ and $G(T(x)) = F(x)$. Using $x = \lambda_1$ in the previous relation and setting $y = T(\lambda_1)$ leads to $G(y) = F(\lambda_1)$. Consider now the following Taylor expansion with integral remainder:

$$\frac{1}{k_{\mu_1}}\int_0^y e^{-\frac{(t-\mu_1)^2}{2\sigma^2}}\,\mathrm{d}t = \frac{1}{k_{\mu_1}}\int_0^{\mu_1} e^{-\frac{(t-\mu_1)^2}{2\sigma^2}}\,\mathrm{d}t + \frac{(y-\mu_1)}{k_{\mu_1}}\int_0^1 e^{-\frac{t^2(y-\mu_1)^2}{2\sigma^2}}\,\mathrm{d}t.$$

Using this expansion in $G(y) = F(\lambda_1)$ and adding $\mu_1(m_1+m_2)(1-r)/r$ on both sides, we get,

$$\begin{aligned}
&\left[\mu_1 - T(\lambda_1)\right]\left(\frac{m_1}{k_{\mu_1}}\int_0^1 e^{-\frac{t^2(T(\lambda_1)-\mu_1)^2}{2\sigma^2}}\mathrm{d}t + \frac{(1-r)}{r}(m_1+m_2)\right) \qquad (3.21)\\
&= \; m_1\left(\frac{1}{k_{\mu_1}}\int_0^{\mu_1} e^{-\frac{(t-\mu_1)^2}{2\sigma^2}}\mathrm{d}t - \frac{1}{k_{\lambda_1}}\int_0^{\lambda_1} e^{-\frac{(t-\lambda_1)^2}{2\sigma^2}}\mathrm{d}t\right) + \frac{(1-r)}{r}(m_1+m_2)(\mu_1-\lambda_1)\\
&\quad + m_2\left(\frac{1}{k_{\mu_2}}\int_0^{T(\lambda_1)} e^{-\frac{(t-\mu_2)^2}{2\sigma^2}}\mathrm{d}t - \frac{1}{k_{\lambda_2}}\int_0^{\lambda_1} e^{-\frac{(t-\lambda_2)^2}{2\sigma^2}}\mathrm{d}t\right).
\end{aligned}$$

We observe as in Lemma 3.2.2, that the difference

$$\epsilon_{11} = \frac{1}{k_{\mu_1}}\int_0^{\mu_1} e^{-\frac{(t-\mu_1)^2}{2\sigma^2}}\mathrm{d}t - \frac{1}{k_{\lambda_1}}\int_0^{\lambda_1} e^{-\frac{(t-\lambda_1)^2}{2\sigma^2}}\mathrm{d}t =: I(\mu_1) - I(\lambda_1)$$

decays exponentially in σ, provided λ_1, μ_1 are far from the boundary (both integrals account for roughly half the mass under the exponentials). Indeed, by the mean value theorem, there exists ξ between λ_1 and μ_1 such that $I(\mu_1) - I(\lambda_1) = I'(\xi)\,(\mu_1 - \lambda_1)$. By explicitly computing I', centering the Gaussian at 0 instead of ξ in the resulting

denominator and then using a change of variable in the integral, we have

$$I'(\xi) \leq \frac{e^{-\frac{\epsilon^2}{2\sigma^2}} \int_\xi^1 e^{-\frac{(t-\xi)^2}{2\sigma^2}} \,\mathrm{d}t + e^{-\frac{\epsilon^2}{2\sigma^2}} \int_0^\xi e^{-\frac{(t-\xi)^2}{2\sigma^2}} \,\mathrm{d}t}{\left(\int_0^1 e^{-\frac{(t-\xi)^2}{2\sigma^2}} \,\mathrm{d}t\right)^2} \leq \frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\int_0^1 e^{-\frac{t^2}{2\sigma^2}} \,\mathrm{d}t} = \frac{e^{-\frac{\epsilon^2}{2\sigma^2}}}{\sigma \int_0^{\frac{1}{\sigma}} e^{-\frac{t^2}{2}} \,\mathrm{d}t}.$$

This proves the exponential decay in σ for σ small enough, as claimed. Next, the difference

$$\epsilon_{12} = \frac{1}{k_{\mu_2}} \int_0^{T(\lambda_1)} e^{-\frac{(t-\mu_2)^2}{2\sigma^2}} \mathrm{d}t - \frac{1}{k_{\lambda_2}} \int_0^{\lambda_1} e^{-\frac{(t-\lambda_2)^2}{2\sigma^2}} \mathrm{d}t$$

also decays exponentially in σ (the Gaussians are integrated over their tails). More specifically, because the integrands are increasing functions of t, we have

$$|\epsilon_{12}| \leq \left[T(\lambda_1) \frac{e^{-\frac{(T(\lambda_1)-\mu_2)^2}{2\sigma^2}}}{k_{\mu_2}} + \lambda_1 \frac{e^{-\frac{(\lambda_1-\lambda_2)^2}{2\sigma^2}}}{k_{\lambda_2}} \right].$$

Then, by assumption (3.20) and the optimality of the transport map T, ($T(\lambda_1)$ will be close to μ_1 which is itself far from μ_2), we get $|T(\lambda_1) - \mu_2| > |\lambda_2 - \mu_2|$ and $|\lambda_1 - \lambda_2| > |\lambda_2 - \mu_2|$ (for example), which in turn yields

$$|\epsilon_{12}| \leq e^{-\frac{(\lambda_2-\mu_2)^2}{2\sigma^2}} \left[\frac{1}{k_{\mu_2}} + \frac{1}{k_{\lambda_2}} \right] \leq e^{-\frac{(\lambda_2-\mu_2)^2}{2\sigma^2}} \frac{2}{\int_0^1 e^{-\frac{t^2}{2\sigma^2}} \,\mathrm{d}t} = \frac{e^{-\frac{(\lambda_2-\mu_2)^2}{2\sigma^2}}}{\sigma} \frac{2}{\int_0^{\frac{1}{\sigma}} e^{-\frac{t^2}{2}} \,\mathrm{d}t}.$$

This shows that ϵ_{12} decays exponentially in σ for σ small enough. By applying the same technique, we obtain an identity similar to (3.21) for $\mu_2 - T(\lambda_2)$ and the same observations apply to the dominating terms. Gathering these estimates, we have

$$\begin{aligned} \frac{\mu_1 - T(\lambda_1)}{\mu_2 - T(\lambda_2)} = & \left(\frac{m_1\epsilon_{11} + \frac{(1-r)}{r}(m_1+m_2)(\mu_1-\lambda_1) + m_2\epsilon_{12}}{m_1\epsilon_{21} + \frac{(1-r)}{r}(m_1+m_2)(\mu_2-\lambda_2) + m_2\epsilon_{22}} \right) \\ & \times \left(\frac{m_2}{m_1} \right) \left(\frac{k_{\mu_1}}{k_{\mu_2}} \right) \left(\frac{\int_0^1 e^{-\frac{t^2(T(\lambda_2)-\mu_2)^2}{2\sigma^2}} \mathrm{d}t + k_{\mu_2}(1-r)\frac{(m_1+m_2)}{m_2}}{\int_0^1 e^{-\frac{t^2(T(\lambda_1)-\mu_1)^2}{2\sigma^2}} \mathrm{d}t + k_{\mu_1}(1-r)\frac{(m_1+m_2)}{m_1}} \right) \end{aligned}$$

where ϵ_{21} and ϵ_{22} are defined similarly to ϵ_{11} and ϵ_{12}. Notice that the ratio $k_{\mu_1}/k_{\mu_2} \approx 1$ for small σ. Moreover, we know that $T(\lambda_i) - \mu_i$ decays linearly in σ in the case of one particle. Here, since the two particles are assumed to be far from each other, then the properties of the OT map will preserve this decay, and thus there exist constants

$C_1 > 0$ and $C_2 > 0$ such that

$$\int_0^1 e^{-\frac{t^2(T(\lambda_2)-\mu_2)^2}{2\sigma^2}} \mathrm{d}t \approx \int_0^1 e^{-C_2 t^2} \mathrm{d}t \quad \text{and} \quad \int_0^1 e^{-\frac{t^2(T(\lambda_1)-\mu_1)^2}{2\sigma^2}} \mathrm{d}t \approx \int_0^1 e^{-C_1 t^2} \mathrm{d}t. \quad (3.22)$$

From (3.22) and the decay in σ of the ϵ_{ij}, there exists $C_3 > 0$ such that

$$\frac{|\mu_1 - T(\lambda_1)|}{|\mu_2 - T(\lambda_2)|} \leq C_3 \frac{|\mu_1 - \lambda_1|}{|\mu_2 - \lambda_2|} \frac{m_2}{m_1} \left(\frac{\int_0^1 e^{-C_2 t^2} \mathrm{d}t + k_{\mu_2}(1-r)\frac{(m_1+m_2)}{m_2}}{\int_0^1 e^{-C_1 t^2} \mathrm{d}t + k_{\mu_1}(1-r)\frac{(m_1+m_2)}{m_1}} \right).$$

Finally, using that both k_{μ_1} and k_{μ_2} are at least $\mathcal{O}(\sigma)$, we conclude that

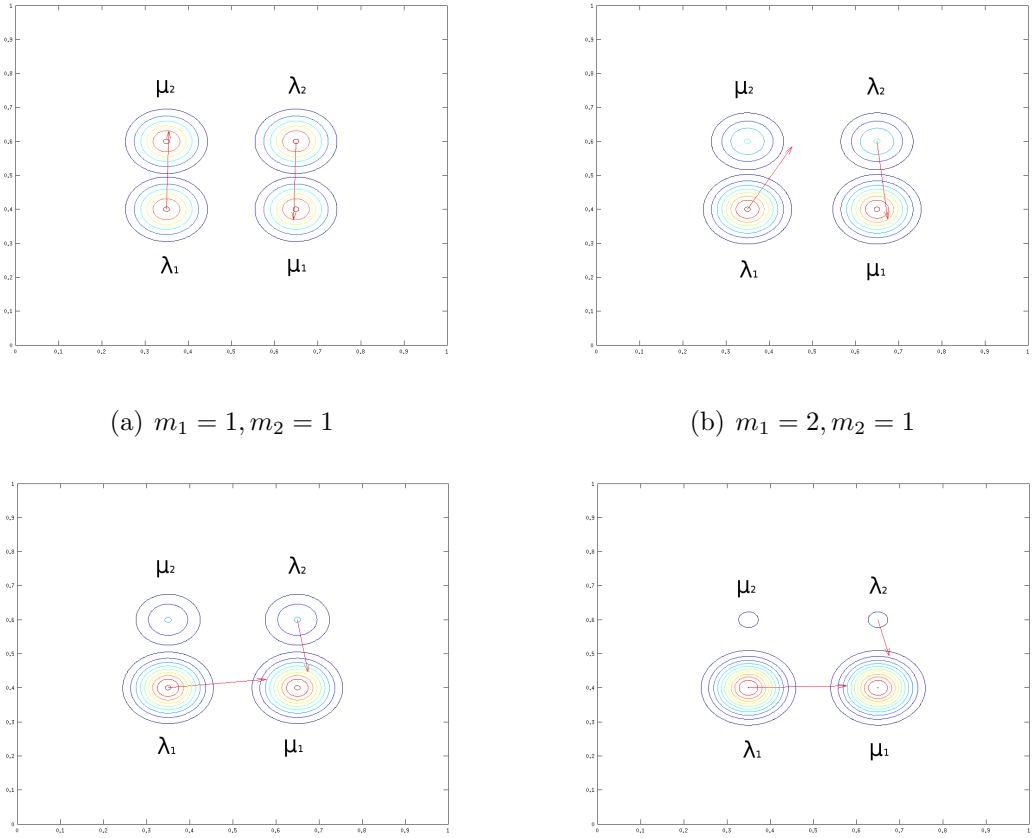
$$\frac{|\mu_1 - T(\lambda_1)|}{|\mu_2 - T(\lambda_2)|} \leq C_4 \frac{|\mu_1 - \lambda_1|}{|\mu_2 - \lambda_2|} \frac{m_2}{m_1} \left(\frac{\int_0^1 e^{-C_2 t^2} \mathrm{d}t}{\int_0^1 e^{-C_1 t^2} \mathrm{d}t} \right) \leq C_5 \frac{m_2}{m_1}$$

for appropriate constants C_4, C_5, and for σ small enough. □

Theorem 3.2.6 shows that the accuracy of the matching is proportional to the mass of a particle: the more mass there is, the more precise the matching is, and vice-versa. This information will be crucial when designing accurate numerical algorithms for the OT method for PIV, as we will see in the next chapter. This result, derived analytically in dimension one, will still be valid in higher dimensions due to the optimal mass redistribution properties of the transport map. We will demonstrate this through numerical experiments in Section 4.2.

3.2.4 Case of Two Particles Outside the Tracking Range

We finish this chapter with a quick discussion on the situation where two particles evolve in a flow so that the final position of each particle is closer to the other particle's initial position (we are thus outside the tracking range). If the masses of the two particles m_1 and m_2 are the same, then we expect the OT solution to mismatch the particles and give the wrong vectors. However, this is not necessarily the case if the weights are not the same. In order to see this, we compute the solution of the OT problem in the case of a crossover between two 2D particles with initial positions $\lambda_1 = (0.35, 0.4)$, $\lambda_2 = (0.65, 0.6)$ and final positions $\mu_1 = (0.65, 0.4)$, $\mu_2 = (0.35, 0.6)$, respectively. The results are presented in Figure 3.3. We see that the error in this regime is also smaller for brighter particles, as in the case analyzed in the previous section. It is therefore feasible to track the brighter particles over longer distances,



(a) $m_1 = 1, m_2 = 1$

(b) $m_1 = 2, m_2 = 1$

(c) $m_1 = 3, m_2 = 1$

(d) $m_1 = 10, m_2 = 1$

Figure 3.3: Results for the OT method applied to two particles for different values of their respective weights. The resulting field obtained via the OT algorithm described in Section 4.1 is displayed at the particles' initial locations.

and this observation will be useful in the following chapter when we develop numerical algorithms to recover the flow.

Chapter 4

Numerical Experiments and Applications

4.1 Numerical Algorithm for Optimal Transport

Before presenting the results of the various numerical experiments conducted for PIV, we introduce in this section our procedure to obtain the solution of the L^2 optimal transport problem in $\mathbb{R}^d$, $d \geq 1$. There are several methods available to solve numerically the transport problem [2, 10, 17, 39, 42]. In [39, 42], we derived an efficient numerical algorithm, based on the work of [28] and [48], for solving the L^2 OT problem (2.2) through the Monge-Ampère equation (2.5) with periodic boundary conditions. As we will use this procedure in all subsequent numerical experiments, we first summarize it in this section. For comparisons with some existing numerical algorithms solving (2.2), we refer to [42].

For simplicity, we consider the case where both the initial and target densities are defined on the torus $\Omega = [0,1]^d$, i.e. that we have periodic boundary conditions. In this setting, it has been proven in [11] that, provided $f, g \in \mathcal{C}^{0,\alpha}(\Omega)$ for $0 < \alpha < 1$ and f, g are bounded away from 0, the optimal map $\nabla\Psi$ is additive, unique up to a constant, and $\mathcal{C}^{2,\beta}(\Omega)$ for $0 < \alpha < \beta$. By additivity, we mean $\nabla\Psi(x+p) = \nabla\Psi(x)+p$ for every $x \in \mathbb{R}^d$ and for all $p \in \mathbb{Z}^d$. This property implies that $\Psi(x)$ can be written in the form $\Psi(x) = |x|^2/2 + u(x)$ where u is a periodic function on Ω. The optimal map can thus be expressed as $T(x) = \nabla\Psi(x) = x + \nabla u(x)$. Assuming Ψ is convex,

the Monge-Ampère equation becomes

$$g(x + \nabla u(x))\det(\mathcal{I} + \mathrm{D}^2 u(x)) = f(x), \tag{4.1}$$

and the velocity field associated with the transport map is given by

$$v_T(x) = \frac{T(x) - x}{\Delta t} = \frac{\nabla u}{\Delta t}.$$

To solve (4.1), we use a damped Newton method based on a generalization of the work of [28]. The main steps are as follows:

Damped Newton algorithm

$$\begin{cases} \text{With } u_0 \text{ given, loop over } n \in \mathbb{N} \\ \text{Compute } f_n = g(x + \nabla u_n)\, \det\left(\mathcal{I} + D^2 u_n\right) \\ \text{Solve the linearized Monge-Ampère equation} \\ \qquad L_n \theta_n := D_{u_n} M \cdot \theta_n = \dfrac{1}{\tau}(f - f_n) \\ \text{Update the solution: } u_{n+1} = u_n + \theta_n. \end{cases} \tag{4.2}$$

Here, $D_{u_n} M \cdot \theta_n$ is the linearization of the Monge-Ampère operator in direction θ_n:

$$D_u M \cdot \theta_n = g(x+\nabla u)\, \mathrm{Tr}\left(\mathrm{Adj}(\mathcal{I}+D^2 u) D^2 \theta_n\right) + \det\left(\mathcal{I} + D^2 u\right) \nabla g(x+\nabla u) \cdot \nabla \theta_n, \tag{4.3}$$

where $\mathrm{Adj}(A) = \det(A) \cdot A^{-1}$. We will denote $D_{u_n} M \cdot \theta_n$ by $L_n \theta_n$ for simplicity. Moreover, the damping factor τ is introduced as a parameter controlling the "step-size" of the Newton iterations, which is necessary to ensure that we remain within the class of convex potentials while iterating towards the solution. For the resolution of the linearized Monge-Ampère equation (4.2), which is an elliptic linear PDE with non-constant coefficients, we employ a technique introduced in [48] which consists in first preconditionning the system in the following way:

$$\begin{cases} L_n \overline{L}_n^{-1} \sigma(x) &= h(x), \\ \overline{L}_n \theta(x) &= \sigma(x), \end{cases} \tag{4.4}$$

where $\overline{L}_n$ is the preconditionner obtained from L_n by averaging the coefficients on $[0,1]^d$. Then, (4.4) is solved with the fast Fourier transform (FFT) and the General-

ized Minimal Residual (GMRES) algorithms, with a finite differences discretization [39, 42]. We use the following stopping criterion for the Newton algorithm iterations:

$$\|u_n - u_{n-1}\|_{l^2_{\text{norm}}} + \|f - f_n\|_{l^2_{\text{norm}}} < \text{TOL}_{\text{Newton}}, \tag{4.5}$$

where $\text{TOL}_{\text{Newton}}$ is a given tolerance, and $\|\cdot\|_{l^2_{\text{norm}}}$ is the normalized l^2 norm defined for $z = (z_1, z_2, \cdots, z_m) \in \mathbb{R}^{m\times k}$ by

$$\|z\|_{l^2_{\text{norm}}} := \sqrt{\frac{1}{m}\sum_{i=1}^{m}\|z_i\|^2_{l^2}}.$$

We use a tolerance $\text{TOL}_{\text{GMRES}} = 10^{-1}$ for the GMRES algorithm employed to solve the linear equation at every Newton step, and we take the tolerance $\text{TOL}_{\text{Newton}} = 10^{-6}$ for the main Newton algorithm. Note that taking $\text{TOL}_{\text{GMRES}} = 10^{-1}$ significantly decreases the amount of computations, and even in the cases presented in this chapter for which the algorithm did not converge, using a smaller tolerance (e.g. $\text{TOL}_{\text{GMRES}} = 10^{-2}$ or $\text{TOL}_{\text{GMRES}} = 10^{-4}$) did not change the outcome. To increase the speed of the algorithm, we use different values for the damping factor according to the precision reached in the stopping criterion. We select an initial value τ_i for τ sufficiently large to guarantee that the damped Newton algorithm proceeds in the right direction. We change this value to a smaller τ_f after the stopping criterion reaches a certain threshold, denoted τ_{t}. From the experiments conducted here, we found that $\tau_{\text{t}} = 10^{-2}$ was typically a good value to take and we employed it to obtain all the results presented in this work. We will also use different types of errors to analyze the results, namely the total relative error e_{total} between v and v_T defined by

$$e_{\text{total}} = \frac{\|v - v_T\|_{l^2_{\text{norm}}}}{\|v\|_{l^2_{\text{norm}}}}$$

and the relative error due to the direction e_{dir} between v and v_T defined by

$$e_{\text{dir}} = \frac{\|\tilde{v} - \tilde{v}_T\|_{l^2_{\text{norm}}}}{\|\tilde{v}\|_{l^2_{\text{norm}}}}$$

where $\tilde{v}$ is the locally normalized vector field:

$$\tilde{v}(x_1, x_2) = \frac{v(x_1, x_2)}{\|v(x_1, x_2)\|_{l^2}},$$

if $\tilde{v}(x_1, x_2) \neq 0$, for any $(x_1, x_2) \in [0,1]^2$. The error e_{dir} will be useful to isolate the error in direction from the error in magnitude.

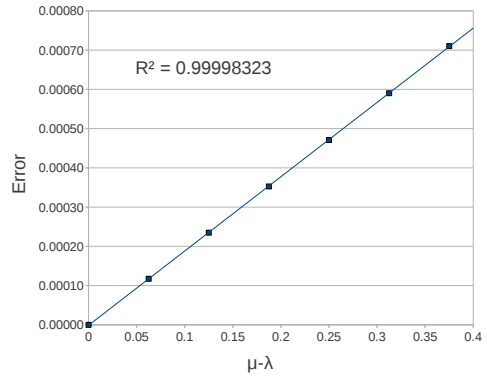
Finally, there are two other points related to this method that are worth mentioning. First, the method actually requires the densities f and g to be bounded away from zero in order to converge. While this may seem like a restriction on the PIV images selected, we found through the numerous experiments that this did not limit its usage, at least for the cases considered in this work. Second, with the current FORTRAN implementation, a typical resolution with 20 Newton iterations for a numerical grid size of $N = 512 \times 512$ points takes only a few seconds on a personal laptop of type Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz. This speed is comparable to the resolution speed of mainstream cross-correlation algorithms [52].

4.2 Validations of Theoretical Results

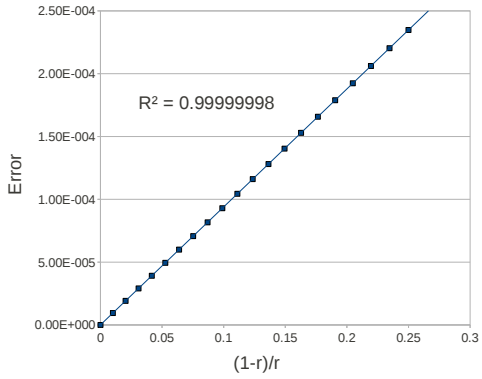
For our first numerical experiments, we validate the theoretical results obtained in Chapter 3. We begin by verifying the results of Theorem 3.2.1, i.e. the linear convergence of $T(\lambda)$ to μ when $\mu - \lambda$, $(1-r)/r$, and σ go to 0. We take the densities f and g as given by (3.3) and solve for $T(\lambda)$ the equation $G(T(\lambda)) = F(\lambda)$ using MATLAB's fzero function, when the three parameters are varied one at the time. Notice that in this case, varying $\mu - \lambda$ corresponds to varying the time-step Δt. The results are presented in figure 4.1 (a), (b), and (c). On these graphs, we also include the least square linear regression lines and the corresponding coefficients of determination R^2. Recall that the coefficient of determination is defined by

$$R^2 = 1 - \frac{\sum_{i=1}^{m} (z_i - l_i)^2}{\sum_{i=1}^{m} (z_i - \overline{z})^2},$$

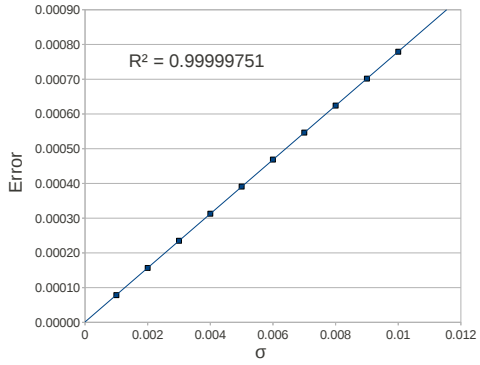
where m is the number of data points, $z = (z_1, \cdots, z_m)$ are the data points, $l = (l_1, \cdots, l_m)$ are the corresponding predicted points on the line, and $\overline{z}$ is the mean of the observed data. We see that in all three cases, R^2 is very close to 1, which confirms the linear behavior in dimension one. This validates the conclusion that the image of the initial position of the center of the particle is sent closer to the real position of the center of the particle after Δt units of time when either $r \to 1$, $\sigma \to 0$, or $|\mu - \lambda| \to 0$,



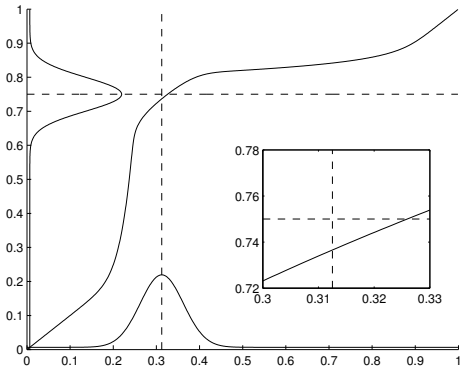
(a) $|T(\lambda)-\mu|$ as a function of $|\mu-\lambda|$, for $r=0.8$ (i.e. $(1-r)/r=0.25$), and $\sigma=0.003$.



(b) $|T(\lambda)-\mu|$ as a function of $(1-r)/r$, for $\sigma=0.003$, $\lambda=5/16$, and $\mu=7/16$.



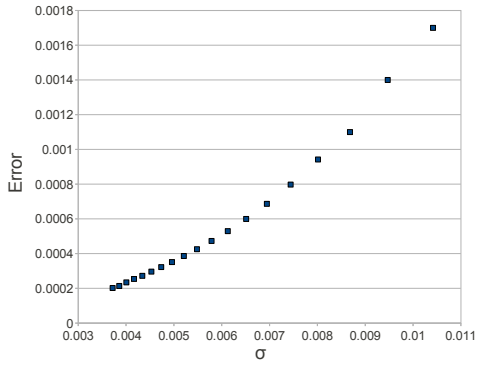
(c) $|T(\lambda)-\mu|$ as a function of σ, for $r=0.8$, $\lambda=5/16$, and $\mu=7/16$.



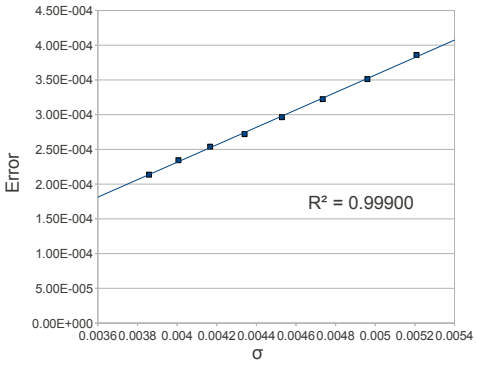
(d) Plot of T for $\lambda=5/16$, $\mu=12/16$, $r=0.8$, and $\sigma=0.05$.

Figure 4.1: Numerical validations of the results of Theorem 3.2.1. For the three error graphs, the linear regression line passing through the data is included, with the corresponding coefficient of determination. For the plot of T, we also added a vertical line at λ, a horizontal line at μ, the densities f and g (normalized to fit in the box) and a zoombox displaying the area of the graph around (λ,μ).

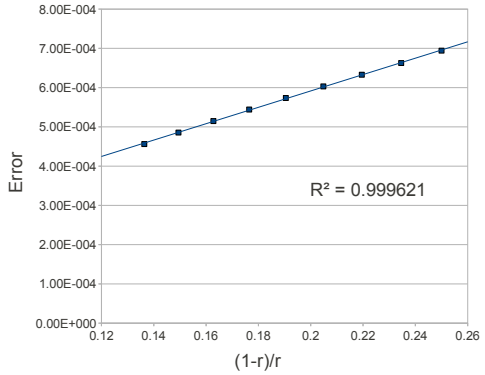
and that the only way to get an exact answer is to take $\Delta t=0$ (i.e. $|\mu-\lambda|=0$) or $\sigma=0$. Notice that taking $r=1$ yields an error that decays exponentially with σ and linearly in $|\mu-\lambda|$. However, this is not warranted in practice for the reasons mentioned earlier: (i) the background noise is dictated by the PIV images and (ii) the convergence of the numerical algorithm in Section 4.1 for the Monge-Ampère equation is guaranteed only when the densities are bounded away from zero. In figure 4.1 (d), we plot the map T for a typical set of parameters. This map was



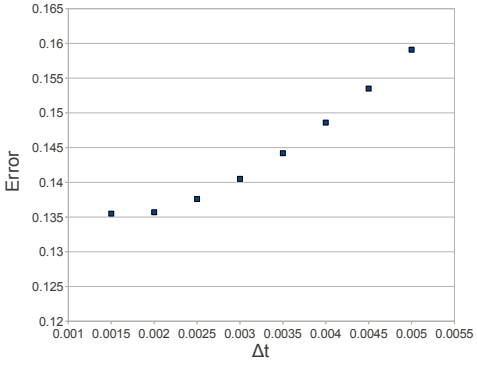
(a) $\|T(\lambda) - \mu\|_{l^2_{\text{norm}}}$ as a function of σ for $N = 1024^2$, $N_p = 16^2$, $r = 0.8$ and $\Delta t = 0.005$.



(b) Tail of the error graph 4.2 a).



(c) $\|T(\lambda) - \mu\|_{l^2_{\text{norm}}}$ as a function of $(1-r)/r$ for $N = 1024^2$, $N_p = 16^2$, $\sigma = 1/144$ and $\Delta t = 0.005$.



(d) $\|v - v_{\text{approx}}\|_{l^2_{\text{norm}}}$ as a function of Δt for $N = 1024^2$, $N_p = 16^2$, $\sigma = 1/144$ and $r = 0.8$.

Figure 4.2: Numerical validation of the results of Theorem 3.2.5 for multiple particles. In the cases where the data follows a linear trend, the linear regression line passing through the data is included, with the corresponding coefficient of determination.

obtained by laying out a uniform grid x_i, $i = 0, \cdots, N$ on $[0, 1]$ with $N = 1024$ and solving $G(y_i) - F(x_i) = 0$ for y_i, at every grid point x_i (once we computed y_i, we assigned $T(x_i) = y_i$). We observe that $T(\lambda)$ is indeed mapped before μ, as predicted by Theorem 3.2.1.

We continue the validations by analyzing the error behavior with respect to the parameters r, σ and Δt for the case of multiple particles in 2D. We take $N_p = 16^2$ particles with evenly distributed initial positions in $\Omega = [0, 1]^2$ and we use the MATLAB ode45 procedure to find their corresponding final positions due to their

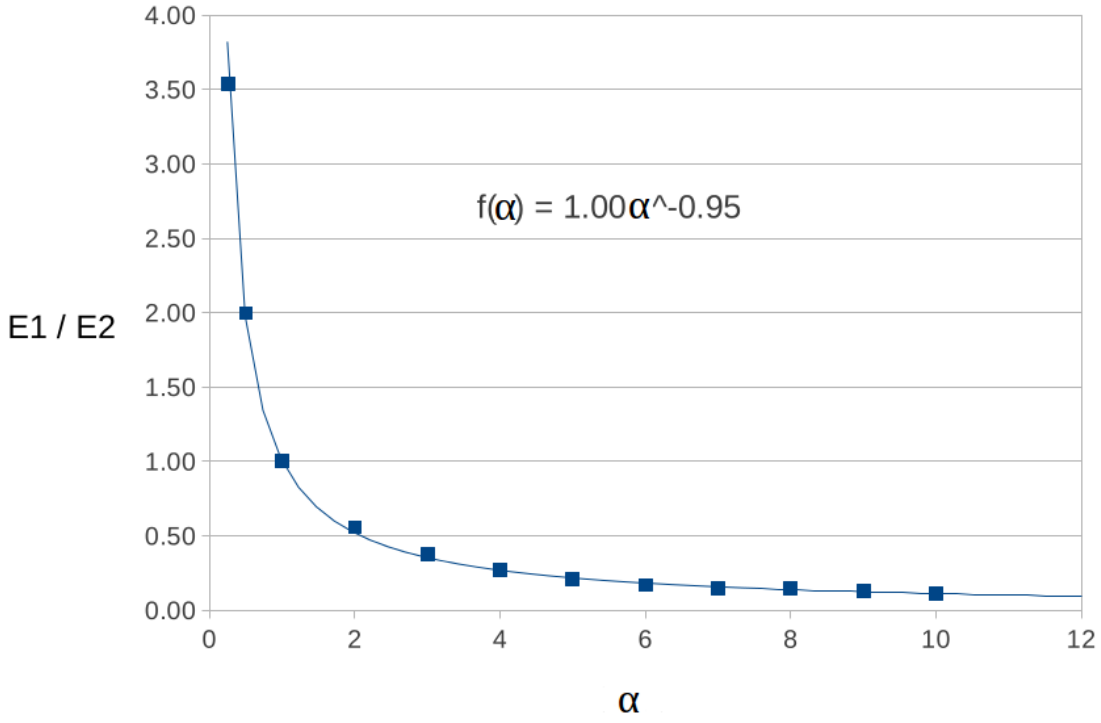


Figure 4.3: Numerical validation of the results of Theorem 3.2.6 for the case of two particles in dimension two. The weight of one particle was varied with respect to the weight of the other particle. The equation of the best power curve passing through all data points is also displayed.

displacement by the synthetic flow given by the array of vortices

$$v(x_1, x_2) = (\sin(2\pi\gamma_1 x_1), \cos(2\pi\gamma_2 x_2) + \pi/2), \tag{4.6}$$

where $\gamma_1 = 2$ and $\gamma_2 = 4$. We then take initial and final densities f and g defined by the model (3.1) and obtain the transport map with the algorithm detailed in the previous section. The results are shown in Figure 4.2. As we see in Figure 4.2 c), the error decreases linearly with respect to $(1 - r)/r$. For the standard deviation, Figure 4.2 a) depicts that for the larger values on the graph, we are in an exponential regime. Then, as σ decreases, we enter a linear regime, as seen in Figure 4.2 b), which is consistent with the results of Theorem 3.2.5. Finally, we display in Figure 4.2 d) the error as a function of Δt. We see that, since the grid size was fixed to $N = 1024$, as Δt decreases, the displacements of the tracers are not all detected by the grid. The

vector field becomes under-resolved and the error stagnates. Before this transition, e_{total} decreases with Δt, but not linearly. This is expected since the trajectories in the flow are not linear.

The last validation we present confirms the behavior of the error with respect to mass variations in the case of two particles within the tracking range, as detailed by Theorem 3.2.6. To demonstrate that this result holds in higher dimension, we run some simple simulations of two particles evolving in dimension two. The results of such an experiment are presented in Figure 4.3 where we took $\sigma = 0.05$, $r = 0.95$, $m_1 = \alpha$, $m_2 = 1$, $\lambda_1 = (0.2, 0.2)$, $\mu_1 = (0.3, 0.2)$, $\lambda_2 = (0.6, 0.6)$, $\mu_2 = (0.7, 0.6)$ and plotted

$$\frac{E_1}{E_2} = \frac{\|\mu_1 - T(\lambda_1)\|_{l^2}}{\|\mu_2 - T(\lambda_2)\|_{l^2}}$$

as a function of α. As expected, we observe from the plot that the ratio of errors is inversely proportional to α because m_2 was fixed and $m_1 = \alpha$ was varied.

4.3 Synthetic Data

Now that we validated the behavior of the error $|T(\lambda) - \mu|$ with respect to the different parameters of the model, we proceed with simple numerical tests using synthetic flows and densities to analyze the quality of the approximation of the target field v by the OT field v_T. We begin with f and g still given by our model (3.1) and with a Δt small enough so that we remain well within the realm of PTV (the tracking range), and then move on to more realistic examples.

Test 1

For this first experiment, we consider a plane shear flow with stationary vector field

$$v(x_1, x_2) = (-\cos(2\pi\gamma x_2), 0),$$

where $\gamma = 2$. Our goal is to investigate the behavior of the approximation with respect to the different parameters involved. We invite the reader to keep in mind that for real data, these parameters would be dictated by the PIV image (whether some pre-processing was applied to it or not) and thus it would not necessarily be possible to vary them at the user's leisure. We first fix $r = 0.8$ and we vary both σ and Δt. The results are presented in Table 4.1 for two different values of the numerical

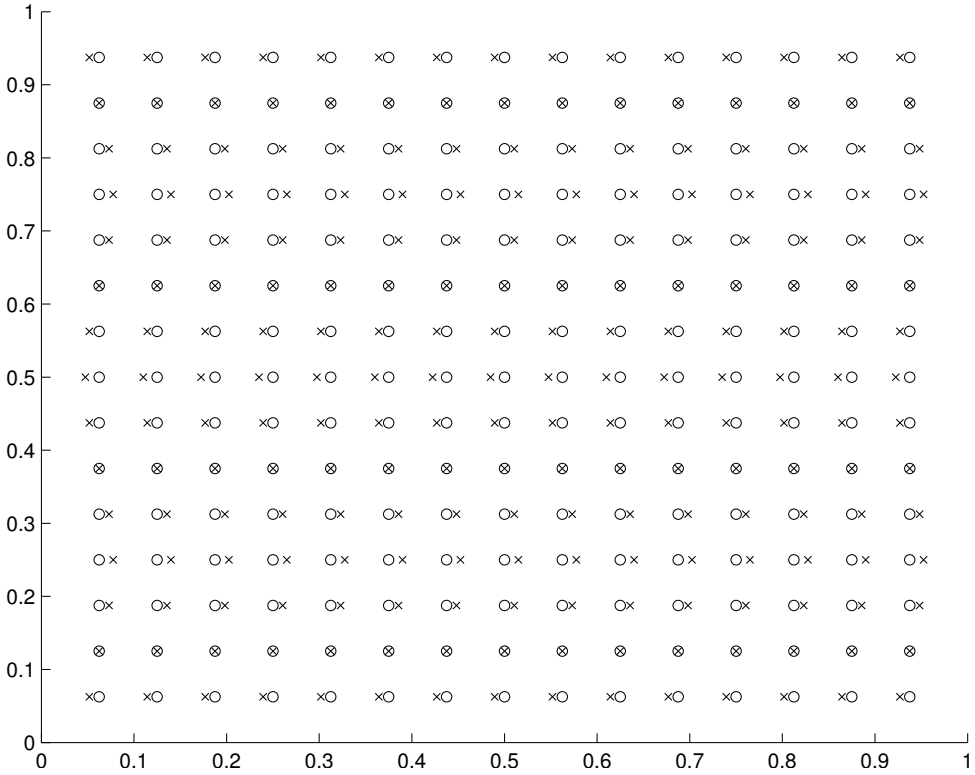
$N_p = 16^2$, $r = 0.8$			e_{total}			e_{dir}		
			Δt			Δt		
			0.01	0.005	0.0025	0.01	0.005	0.0025
$\sigma = 1/96$	N	512	0.2493	0.2267	0.2201	0.0239	0.0203	0.0192
		1024	0.2496	0.2268	0.2202	0.0239	0.0203	0.0192
$\sigma = 1/144$	N	512	0.0899	0.0819	0.0806	0.0046	0.0048	0.0076
		1024	0.0902	0.0820	0.0806	0.0045	0.0048	0.0076
$\sigma = 1/192$	N	512	0.0455	0.0441	0.0438	0.0036	0.0027	0.0046
		1024	0.0476	0.0444	0.0438	0.0018	0.0025	0.0046

Table 4.1: Results for synthetic test 1 when σ is varied. The errors e_{total} and e_{dir} presented are computed for v and v_T at the particle's initial locations.

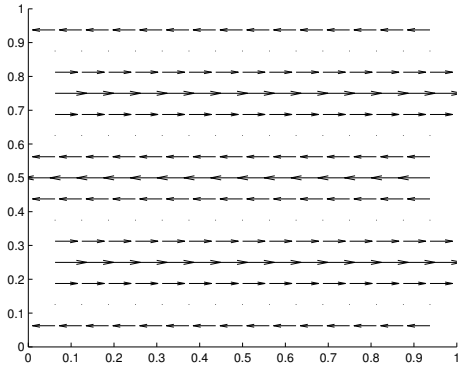
grid $N = N_1 \times N_2 = 512 \times 512$ and 1024×1024.

We observe that the relative error e_{tol} between v_T and v decreases when we decrease σ, as predicted by the theory. The fairly high grid size employed is required for the algorithm to converge for small values of σ. Indeed, if we consider that a particle's physical width spans over approximately 6 standard deviations (6σ), then for $\sigma = 1/96$, taking $N = 128 \times 128$ gives about 8 grid points per particle and this was sufficient to reduce the error to the desired tolerance. However, for $\sigma = 1/192$, the algorithm did not converge for $N = 128 \times 128$ (4 points per particles) nor $N = 256 \times 256$ (8 points per particle) when $\Delta t = 0.01$. We therefore selected a higher ratio of 16 ($N = 512 \times 512$) or 32 ($N = 1024 \times 1024$) grid points per particle to ensure a good enough resolution so that the variations of the density are properly captured by the underlying mesh. For the locally normalized vector fields, the relative error e_{dir} is much smaller than e_{total}, which means that most of the error is located in the magnitude of the individual vectors. This observation can be explained by the fact that particles remain well within the tracking range for this first test. We can visualize some results for this experiment by looking at a specific case in Figure 4.4, where one example of the field v_T is presented beside the target field v.

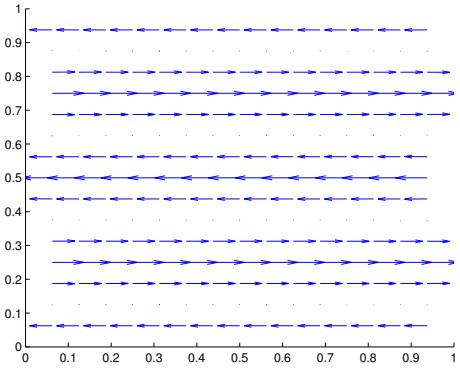
Observe next that the relative error for the locally normalized vector fields e_{dir} also decreases with σ, but ceases to decrease beyond a certain level with respect to Δt for a fixed σ. The reason behind this is that for the specific values of Δt taken, the relative error due to the finite differences approximation is negligible, but increasing. More specifically, for $\Delta t = 0.01$, $\Delta t = 0.005$ and $\Delta t = 0.0025$, this error is respectively 2.4×10^{-14}, 5.8×10^{-14}, and 1.6×10^{-13} (for comparison, the machine epsilon used



(a) Initial and final distributions of particles. The circles represent the initial positions and the crosses the final positions. Note that the standard deviations are not represented on this plot.



(b) Target vector field v.



(c) Approximate vector field v_T.

Figure 4.4: Results for synthetic test 1, when $N_p = 16^2$, $\sigma = 1/192$, $r = 0.8$, $N = 1024^2$ and $t = 0.01$. The vector fields were scaled in order to better visualize these results. The maximum magnitude is 1 for the target vector field and 0.9569 for the approximate vector field.

$N_p = 16^2$, $\sigma = 1/144$			e_{total}			$\|T(\lambda) - \mu\|_{l^2_{\text{norm}}}$		
			Δt			Δt		
			0.01	0.005	0.0025	0.01	0.005	0.0025
$r = 0.825$	N	512	0.0816	0.0731	0.0713	5.58e-04	2.50e-04	1.22e-04
		1024	0.0818	0.0732	0.0714	5.59e-04	2.50e-04	1.22e-04
$r = 0.85$	N	512	0.0726	0.0639	0.0628	4.96e-04	2.18e-04	1.07e-04
		1024	0.0728	0.0640	0.0629	4.97e-04	2.19e-04	1.07e-04
$r = 0.875$	N	512	0.0637	0.0551	0.0542	4.35e-04	1.88e-04	9.25e-05
		1024	0.0643	0.0551	0.0542	4.39e-04	1.88e-04	9.25e-05

Table 4.2: Results for synthetic test 1, when r is varied. The errors e_{total} and $\|T(\lambda) - \mu\|_{l^2_{\text{norm}}}$ presented are computed for v and v_T at the particle's initial locations.

was 2.2×10^{-16}). In fact, we can understand the effect of both errors through the following inequalities (for one particle), which build on (3.2):

$$|v_1 - v_{T_1}| \leq \left| v_1 - \frac{\mu_1 - \lambda_1}{\Delta t} \right| + C(r, \sigma) \left| \frac{\mu_1 - \lambda_1}{\Delta t} \right| \leq \mathcal{O}(\Delta t) + C(r, \sigma) \|v_1\|_\infty,$$

where $C(r, \sigma)$ is the bound given in Theorem 3.2.5, and v_1 and v_{T_1} are the first components of respectively the target vector field v and the approximate vector field v_T (a similar bound can be obtained for the second components of the vectors). We see that we are in a regime where the velocity error is dominated by the effect of σ and r alone, since the term in $\mathcal{O}(\Delta t)$ becomes negligible for Δt small enough, which is the case here. We will present a different regime for which the finite differences error is significant in the next numerical test.

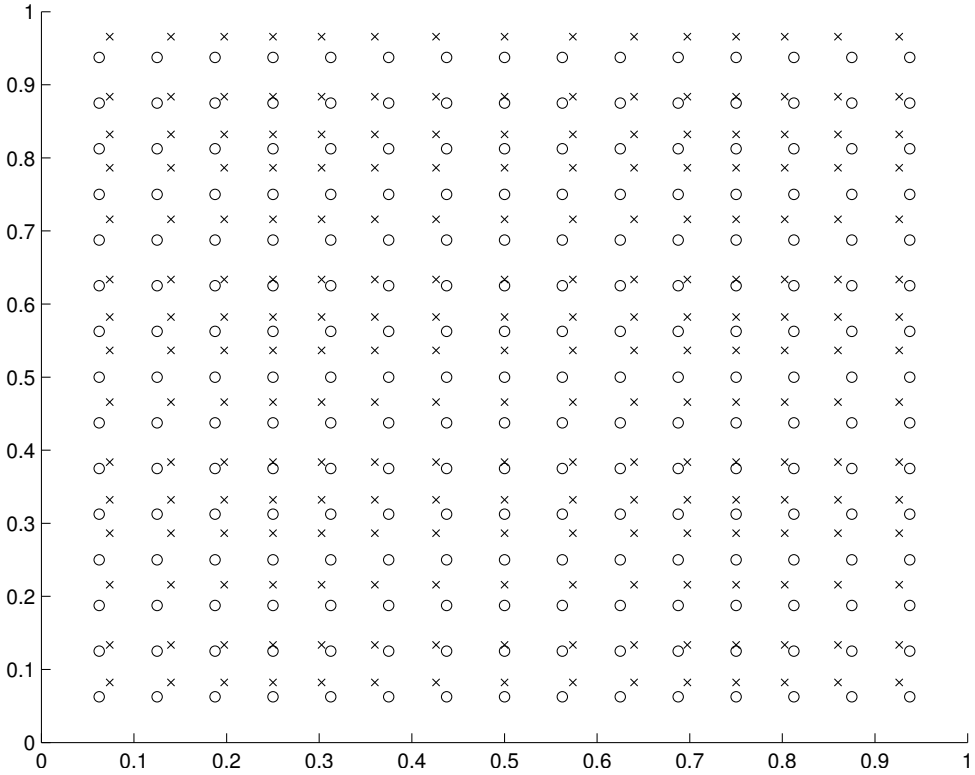
Let us now look at results for which only r is varying, in Table 4.2. The relative error also decreases as r increases and as Δt decreases. In the second part of this table, we display the total error in the final position of the particles, that is the $\|\cdot\|_{l^2_{\text{norm}}}$ norm of the vector containing the errors $\mu - T(\lambda)$ for every particle. We observe that this error in the final position also decreases as r increases and as Δt decreases, as expected. Note finally that all the results obtained in Table 4.1 and Table 4.2 were obtained using a multiresolution approach, where for example in Table 4.1, the final solution obtained for $\sigma = 1/96$ was taken to be the starting solution for $\sigma = 1/144$. In every case, we took $\tau_i = 3$ and $\tau_f = 1$.

$N_p = 16^2$, $r = 0.8$			e_{total}			e_{dir}		
			Δt			Δt		
			0.01	0.005	0.0025	0.01	0.005	0.0025
$\sigma = 1/96$	N	512	0.2513	0.1957	0.1803	0.1685	0.1156	0.1018
		1024	0.2523	0.1959	0.1803	0.1697	0.1157	0.1018
$\sigma = 1/144$	N	512	0.1218*	0.0865	0.0743	0.0522*	0.0321	0.0240
		1024	0.1296	0.0867	0.0744	0.0591	0.0322	0.0240
$\sigma = 1/192$	N	512	0.0888*	0.0563	0.0450	0.0376*	0.0200	0.0122
		1024	0.0980	0.0593	0.0454	0.0400	0.0206	0.0123

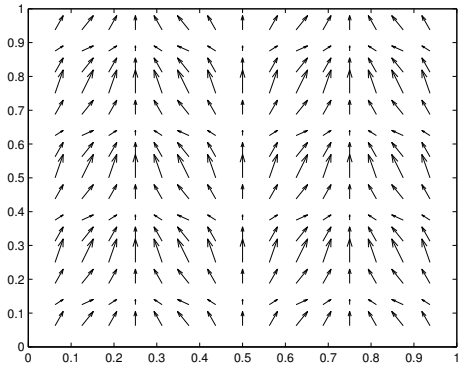
Table 4.3: Results for synthetic test 2, when σ is varied. The errors e_{total} and e_{dir} presented are computed for v and v_T at the particle's initial locations. If a * is displayed, it means that the algorithm did not converge to the given tolerance (it stagnated).

Test 2

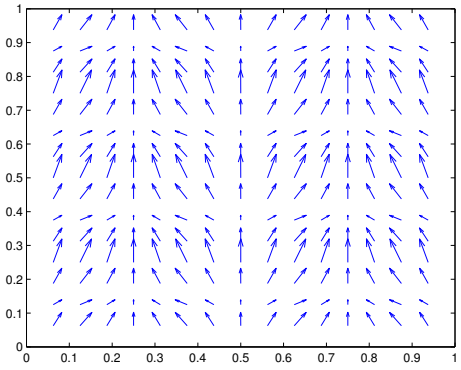
For the second test, we consider the synthetic flow given by the array of vortices in (4.6). We present the results of similar experiments as the ones presented in the previous test, in Table 4.3 and Figure 4.5. We observe a similar behavior for e_{total} as for the shear experiment. However, several differences arise. First, e_{dir} now decreases with Δt as opposed to the previous case. This can be explained by the fact that the finite differences relative error is now 0.0821, 0.0406 and 0.0202 for $\Delta t = 0.01$, $\Delta t = 0.005$ and $\Delta t = 0.0025$, respectively. Observe also that e_{total} is close to this finite differences error so the algorithm is giving good results. The error in direction e_{dir} is now a bigger portion of the relative error, but as σ and Δt decrease, this portion decreases significantly. Second, in the case where $N = 512^2$ and $\Delta t = 0.01$, the algorithm did not converge anymore for $\sigma = 1/144$ and $\sigma = 1/192$. In both cases, we let the Newton algorithm evolve even if the GMRES algorithm did not converge to a tolerance of $\text{TOL}_{\text{GMRES}} = 0.1$ on the relative error for some iterates (the maximum number of iterations was set to 50 for GMRES). We stop the algorithm when this relative residual in the GMRES algorithm becomes bigger than 0.99, that is when the new computed solution is almost the same as the one computed in the previous iteration. This happened after $n = 27$ Newton iterations for $\sigma = 1/144$ and after $n = 24$ for $\sigma = 1/196$. In the former case, we reduced the errors to $\|f_n - f\|_{l^2_{\text{norm}}} = 0.0041$, $\|u_n - u_{n-1}\|_{l^2_{\text{norm}}} = 7.0084 \times 10^{-8}$ and $\|f_n - f\|_{l^2_{\text{norm}}}/\|f\|_{l^2_{\text{norm}}} = 3.2496 \times 10^{-6}$ whereas we got $\|f_n - f\|_{l^2_{\text{norm}}} = 0.0327$,



(a) Initial and final distributions of particles. The circles represent the initial positions and the crosses the final positions. Note that the standard deviations are not represented on this plot.



(b) Target vector field.



(c) Approximate vector field.

Figure 4.5: Results for synthetic test 2, when $N_p = 16^2$, $\sigma = 1/192$, $r = 0.8$, $N = 1024^2$ and $\Delta t = 0.01$. The vector fields were scaled in order to better visualize these results. The maximum magnitude is 2.7584 for the target vector field and 2.5722 for the approximate vector field.

$M = 32^2$ $N = 1024^2$	e_{total}			$\|u_n - u_{n-1}\|_{l^2_{\text{norm}}}$		
	Δt			Δt		
	0.01	0.005	0.0025	0.01	0.005	0.0025
$\sigma = 1/192$, $r = 0.8$	0.9558*	0.2752	0.2129	1.92e-08	7.43e-12	3.82e-12
$\sigma = 1/192$, $r = 0.85$	0.9625*	0.2544	0.1878	2.32e-07	3.77e-12	1.60e-11
$\sigma = 1/288$, $r = 0.8$	1.0005*	0.2138*	0.0852	2.74e-06	6.48e-08	1.08e-11
$\sigma = 1/288$, $r = 0.85$	1.0011*	0.2039*	0.0680	1.70e-06	1.81e-06	2.68e-10

$N_p = 32^2$ $N = 1024^2$	$\|f_n - f\|_{l^2_{\text{norm}}}$			$\frac{\|f_n - f\|_{l^2_{\text{norm}}}}{\|f\|_{l^2_{\text{norm}}}}$		
	Δt			Δt		
	0.01	0.005	0.0025	0.01	0.005	0.0025
$\sigma = 1/192$, $r = 0.8$	0.0012	6.50e-07	7.13e-07	6.14e-07	1.82e-13	2.20e-13
$\sigma = 1/192$, $r = 0.85$	0.0677	8.67e-07	5.00e-07	0.0018	3.02e-13	1.00e-13
$\sigma = 1/288$, $r = 0.8$	0.4408	0.0655	7.10e-07	0.0408	9.01e-04	1.06e-13
$\sigma = 1/288$, $r = 0.85$	0.5505	0.1312	9.46e-07	0.0578	0.0033	1.70e-13

Table 4.4: Results for test 2 for the case $N_p = 32^2$ and $N = 1024^2$. If a * is displayed, it means that the algorithm did not converge to a tolerance of $\text{TOL}_{\text{Newton}} = 10^{-6}$ in the stopping criterion (it stagnated).

$\|u_n - u_{n-1}\|_{l^2_{\text{norm}}} = 3.0946 \times 10^{-07}$ and $\|f_n - f\|_{l^2_{\text{norm}}}/\|f\|_{l^2_{\text{norm}}} = 1.2206 \times 10^{-4}$ for the latter case. Nevertheless, as we can see in the table, the relative error e_{total} is quite close to the one for $N = 1024^2$. Therefore, we realize that a tolerance of $\text{TOL}_{\text{Newton}} = 10^{-6}$ on the stopping criterion (4.5) might be too stringent and that acceptable results could be obtained with less iterations. We also point out that varying the tolerance originally set to $\text{TOL}_{\text{GMRES}} = 10^{-1}$ for the GMRES algorithm does not prevent this stagnation from occurring.

The only parameter we have not varied so far is the number of tracers N_p. In Table 4.4, we present some results for $N_p = 32^2$. We fix the numerical grid size to $N = 1024^2$. From the table, we see that the procedure did not converge in any case when $\Delta t = 0.01$ or for two cases for $\Delta t = 0.05$. For these cases, the optimal solution is further from the zero initial guess in the Newton algorithm and a bigger grid size would be required to obtain convergence. To see how far the algorithm was from reaching the tolerance of $\text{TOL}_{\text{Newton}} = 10^{-6}$, we also presented in Table 4.4 the value of some errors involved for the last iterate. Note that the smaller values of σ

tested were selected to ensure that individual particles do not overlap in the density distribution and that they always remain within the tracking range.

Test 3

In this third test, we still consider densities given by our analytical model (3.1), but we now venture to the boundary of the tracking regime. We select another shear on $\Omega = [0,1]^2$, this time given by

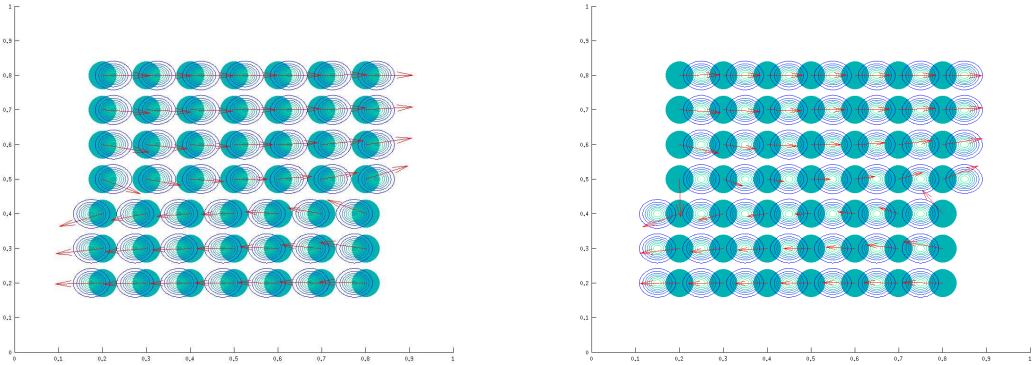
$$v(x_1, x_2) = \begin{cases} (\gamma, 0) & \text{if } x_2 \geq 0.5 \\ (-\gamma, 0) & \text{if } x_2 < 0.5 \end{cases}$$

for a given constant parameter γ. This field is obviously not continuous, but we will use it to approximate a continuous horizontal shear with a sharp transition between the two interfaces.

We can observe from Figure 4.6 the result of the OT algorithm applied to the example under study. We see that if the particles do not move further than the tracking regime (flow displacement smaller than mean distance between particles) and if we stay far enough from the shear's transition zone, then the OT algorithm recovers almost perfectly the given vector field. Note that this remains true for the "in between" regime, that is when the final positions are exactly in between two initial positions (Figure 4.6 b). When the shear is stronger and for the same time interval, particles exit the tracking range, and mismatches occur almost everywhere. Actually, a vertical transfer of mass now occurs at the edges since the vertical distance between particles is now smaller than the horizontal distance.

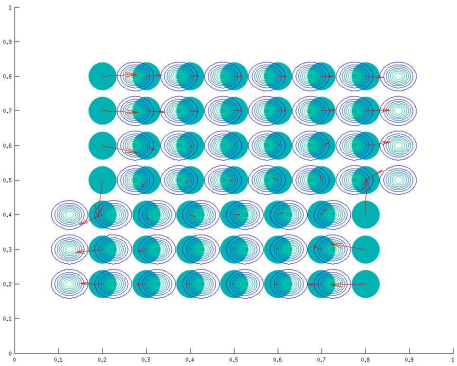
Let us now make two observations on the errors made by the method for the "in between" regime of Figure 4.6 b). First, the least accurate vectors are located at the shear's transition zone (especially at the edges). At these positions, the optimal solution to the L^2 OT problem splits the mass of the Gaussian distributions into pieces, distributing it towards neighboring target positions of particles. Because these masses correspond to physical tracers, this solution does not make physical sense. It would therefore be appropriate to remove some of these vectors from the final approximation (at the very least the ones at the edges).

In addition, we also observe in Figure 4.6 b) that there is an error even in the vectors far from the shear's transition zone. These vectors' directions are quite accurate, but their length is always smaller than the lengths of the target vectors. This

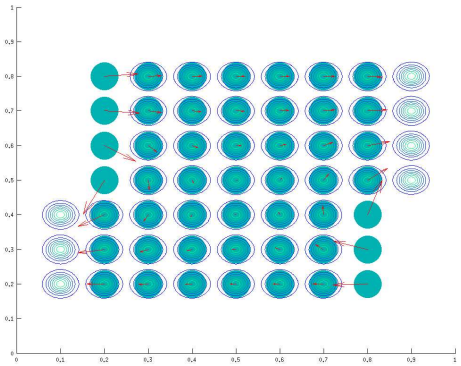


(a) $\gamma = 0.25$

(b) $\gamma = 0.50$



(c) $\gamma = 0.75$



(d) $\gamma = 1.00$

Figure 4.6: Results for test 3 for different values of the shear strength γ. The full circles represent the initial positions of particles whereas the transparent circles represent their final positions.The resulting field obtained via our OT algorithm is displayed at the particles' initial locations.

is due to the OT map which does not send the center of a Gaussian directly to its target center in the tracking range, as predicted by Theorems 3.2.1 and 3.2.5. In this particular case, this error could be corrected by simply assigning the image of this center to the brightest neighboring pixel.

Finally, we mention that while the detailed trajectories and flow properties of the OT map and those associated with the actual flow field are both available for synthetic examples, the use of the L^2 OT method to recover an approximate representation of the later should aim at precisely matching the initial position with the final position of the particles given by successive PIV images and not per se at predicting the trajec-

tories of the particles. In fact, the OT and the fluid flow can behave very differently even in cases where the effect of pressure can be neglected. Simple shear flows of the type presented here can illustrate this fact. Indeed, notice that a constant shear (of the form $v(x_1, x_2) = (\gamma(x_2), 0)$ for example, which is by construction a solution to the incompressible Euler equations for fluid flows) satisfies the first equation in the system (2.9). The second equation becomes

$$\frac{\partial \rho}{\partial t} + \gamma(x_2)\frac{\partial \rho}{\partial x_1} = 0$$

and signifies that the initial density is simply transported horizontally according to $\gamma(x_2)$. However, since the optimal map T is known to be the gradient of a convex function, we also know that v is a potential flow and thus cannot accurately represent a shear flow. Therefore, it is not possible with the OT method to go beyond the simple first-order accurate in time approximation corresponding to the matching problem.

4.4 Post-Processing Filters

We now use the observations from the previous section's numerical experiments on synthetic data to build two post-processing filters designed to reduce some of the errors presented. First, the shear experiment in test 3 shows that the OT solution will sometimes split the mass associated to one particle into several parts to be distributed among neighboring particles. This naturally leads to a post-processing filter where we identify and remove the vectors associated with a particle whose mass is being split.

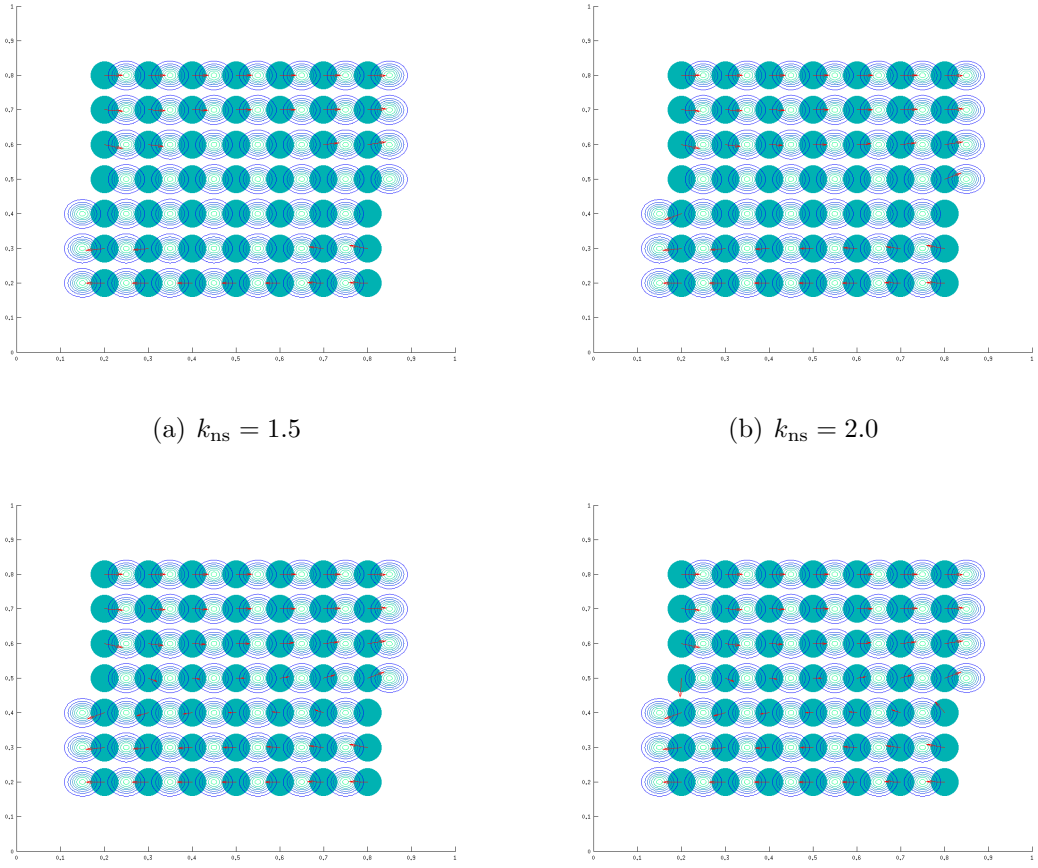
No Split Filter

Let λ_i be the initial position of a particle on the numerical grid, let Λ_{λ_i} be the set of all 8 neighboring pixels and let $k_{ns} > 0$ be a parameter. Then, if there exists $\mathbf{x} \in \Lambda_{\lambda_i}$ for which

$$k_{ns}|\mathbf{x} - \lambda_i| < |T(\mathbf{x}) - T(\lambda_i)|, \tag{4.7}$$

the vector $v_i = \dfrac{T(\lambda_i) - \lambda_i}{\Delta t}$ is rejected.

The positive parameter k_{ns} determines the strength of the filter and numerical experiments indicate that it should typically be taken in between 1 and 10. Note here that mass splitting is possible because we are considering continuous densities



(a) $k_{\text{ns}} = 1.5$

(b) $k_{\text{ns}} = 2.0$

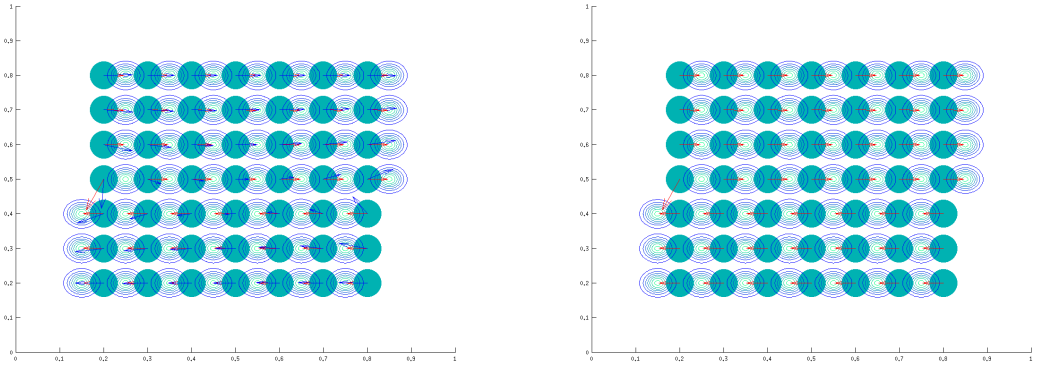
(c) $k_{\text{ns}} = 10.0$

(d) $k_{\text{ns}} = 20.0$ (filter has no effect and no vectors are removed)

Figure 4.7: The No Split filter applied to experiment b) in Figure 4.6, for different values of k_{ns} (strength of filter).

(not to be confused with mass splitting of Dirac Delta distributions, which is not allowed by the push-forward constraint imposed on the transport problem). Figure 4.7 shows the effect of this filter on experiment b) displayed in Figure 4.6. We see on this figure that this simple filter effectively removes spurious vectors for the shear experiment of test 3. We shall also observe its effectiveness on more realistic data sets in the subsequent sections.

The other natural post-processing filter motivated by previous observations consists in changing the predicted target position of tracers to the position corresponding to the brightest surrounding pixel. Provided the right assignment is achieved, this would ideally help correcting the error due to the transport map sending the center



(a) Superposition of both fields

(b) Corrected field only.

Figure 4.8: PTPC filter applied to the shear example in Figure 4.6 b).

of a Gaussian before the target center, in the presence of noise, as predicted by theorems 3.2.1 and 3.2.5 and demonstrated by the numerical tests in the previous section.

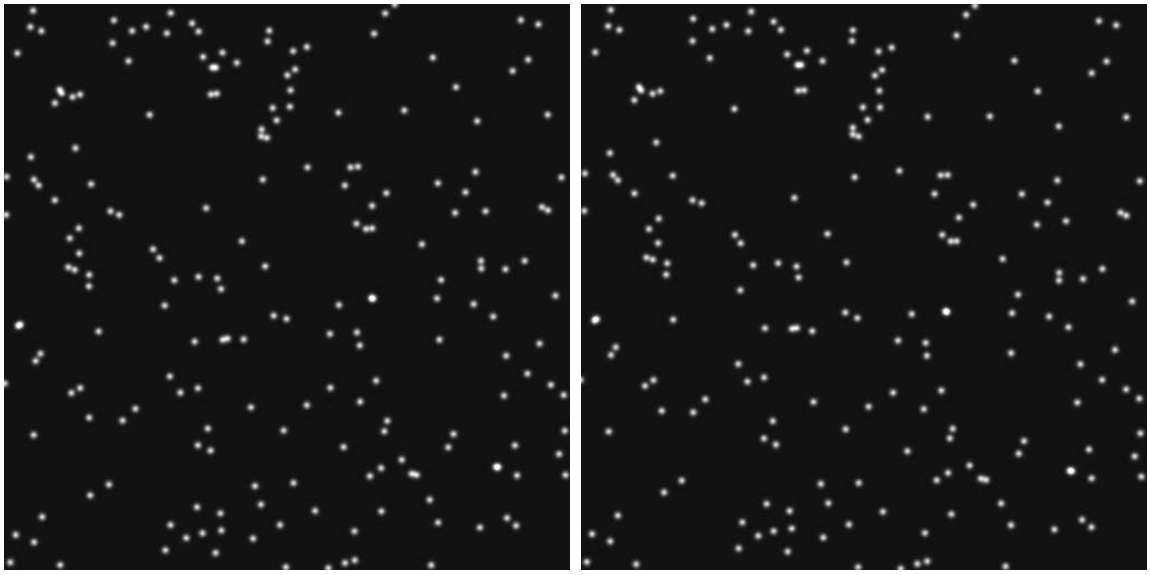
Particle Target Position Correction (PTPC) Filter

If λ is the position of a bright particle and $T(\lambda)$ its predicted final position and if Γ is the set of all pixels close to $T(\lambda)$, in the sense that

$$|y - T(\lambda)|_{l^1} < c \tag{4.8}$$

for $y \in \Gamma$ and c a constant, then, we select $y \in \Gamma$ such that the final density $g(y)$ is the greatest and we assign it to the final position of the particle at λ.

Let us first test this strategy on the example b) of Figure 4.6. The results are presented in Figure 4.8 for $c = 40$ (the total image is 512×512 pixels). We see that this correction eliminates the error made by the transport map for every particle except one. For that tracer, the brightest neighboring pixel to the image by the transport map does not correspond to the same initial tracer and the algorithm leads to a mismatch. This error will be more frequent when a larger number of tracers is considered in the next experiments. However, in this specific case, we can already eliminate this vector with the No Split filter as previously shown.



(a) f (b) g

Figure 4.9: Synthetic images created with the vortex (4.9).

4.5 Synthetic Data with Filters

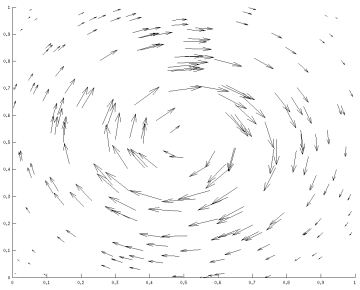
Test 4

In order to further validate the effect of the filters we presented, we run another test involving synthetic densities of the type (3.1). We take a fixed number of particles N_p with equal masses and randomly distributed initial positions in $\Omega = [0,1]^2$. We let them evolve according to the velocity field given by

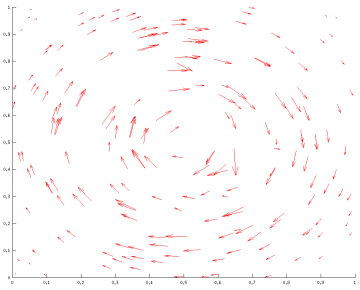
$$v(x_1, x_2) = \Big(x_2 - 0.5, 0.5 - x_1\Big) \exp\Big(- 10((x_1 - 0.5)^2 + (x_2 - 0.5)^2)\Big). \qquad (4.9)$$

We then use their final positions after $\Delta t = 0.25$ units of time to create a second density with Gaussian distributions. The initial and final densities are thus taken as f and g (200 tracers were seeded to create the densities). They are displayed in Figure 4.9. We applied the transport algorithm with different filters on these initial and final images and the results are presented in Figures 4.10 a), b), c) and d).

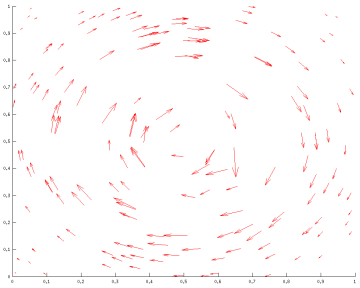
The velocity field a) shows the result of a perfect match between initial and final positions, i.e. the only error is the error due to the finite differences approximation. In b), we see the field obtained with the OT algorithm. The initial position of particles was not assumed to be known, and every pixel was considered to be a particle if it was brighter than a certain brightness threshold (0.8 in this case). This simple strategy



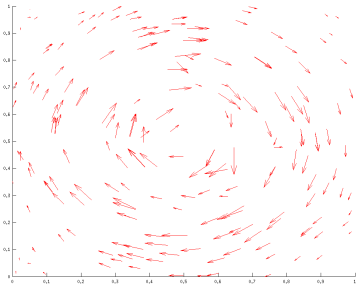
(a) Field obtained with a perfect match (the only error is due to finite differences).



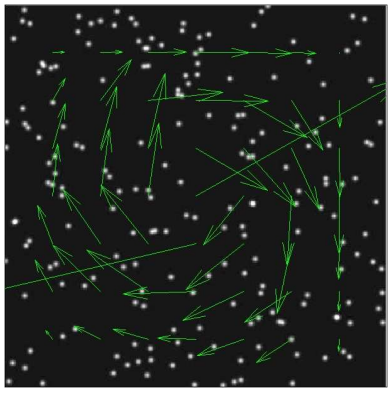
(b) Field obtained with the OT algorithm.



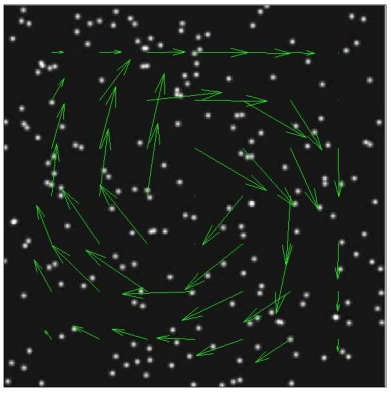
(c) OT field with the No Split filter and $k_{\text{ns}} = 5$.



(d) OT field with the PTPC filter and $c = 3$.



(e) Field obtained with the DCC algorithm superposed to the initial density.



(f) Field obtained by filtering the field in e) with a velocity threshold set at 2 standard deviations away from the mean velocity in either components.

Figure 4.10: Results of the OT and DCC algorithms for Test 4. For the DCC algorithm, the interrogation window size selected is 128×128 pixels, the step size is 64×64 pixels and a Gaussian 2X3 points subpixel approximation was also employed.

Fields Considered	l^2_{norm} Error	l^∞ Error	N_p
Finite Differences and OT	4.92×10^{-3}	3.37×10^{-2}	198
Finite Differences and OT with $k_{ns} = 5$	4.51×10^{-3}	2.90×10^{-2}	168
Finite Differences and OT with $c = 3$	4.40×10^{-3}	3.53×10^{-2}	198

Table 4.5: l^2_{norm} and l^∞ norms of the error between the fields displayed in Figures 4.10 a), b), c) and d). N_p represents the number of vectors used in each case.

Parameter Values		k_{ns}				
		No Filter	5.0	3.0	1.5	1.2
c	No Filter	5.4×10^{-3}	5.2×10^{-3}	4.6×10^{-3}	4.4×10^{-3}	4.2×10^{-3}
	3	5.3×10^{-3}	5.1×10^{-3}	4.5×10^{-3}	4.3×10^{-3}	4.2×10^{-3}
	5	5.2×10^{-3}	5.0×10^{-3}	4.4×10^{-3}	4.3×10^{-3}	4.2×10^{-3}
	9	5.1×10^{-3}	4.9×10^{-3}	4.4×10^{-3}	4.3×10^{-3}	4.2×10^{-3}
N_p		198	168	119	101	31

Table 4.6: l^2_{norm} errors between the exact field (4.9) and the OT field for the given parameter values. N_p represents the number of vectors used in each case.

to detect particles will be explained in more details in the next sections. By doing so, we obtain a vector for 198 of the 200 particles initially seeded. Note that some particles are not individually detected because they are too close to another particle. The field c) shows the result obtained by applying the No Split filter to field b) with the parameter $k_{\text{ns}} = 5$ (30 vectors are removed). Finally, we see in d) the field given by applying the PTPC filter to b).

To quantify the improvement done by the two filters considered, let us first compare the l^2_{norm} and l^∞ norms of the errors between the fields obtained and the "perfect match" field at the particle's initial positions. The results are presented in Table 4.5. We see that the No Split filter effectively removes some bad vectors. In addition, even if the PTPC filter does not improve the quality of every single vector (the l^∞ norm increases), it decreases the global l^2_{norm} error significantly. To look more specifically at the effect of the parameters on the l^2_{norm} error, we present in Table 4.6 this error for different values of the parameters k_{ns} and c, but this time for the exact field given by (4.9) and not the finite differences field. We see that the No Split filter significantly reduces the l^2_{norm} error, at the expense of removing more and more vectors. On the other hand, the PTPC filter does not improve the error as much, but does not remove any vectors. It has a greater effect when the No Split filter is not too strong, but when the value of c is greater than 9, the l^2_{norm} error starts increasing in some cases due to mismatches. Therefore, a mild application of both filters is desirable. Note finally

Fields Considered	l^2_{norm} Error	l^∞ Error	N_p
Exact Field and DCC	1.22×10^{-2}	5.41×10^{-1}	49
Exact Field and DCC with filtering	4.20×10^{-3}	9.49×10^{-2}	45

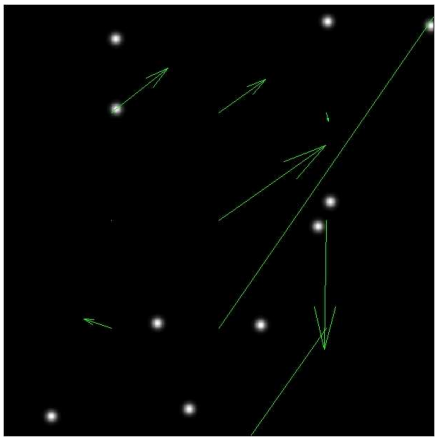
Table 4.7: l^2_{norm} and l^∞ norms of the error between the fields displayed in Figures 4.10 e) and f). N_p represents the number of vectors used in each case. For the DCC algorithm to be effective, a reasonable number of particles per window needs to be used and thus less vectors are produced when compared to the OT-PIV method.

that the No Split filter has to be applied before the PTPC filter because the latter modifies the direction of vectors, and the No Split filter relies on these directions to remove spurious vectors.

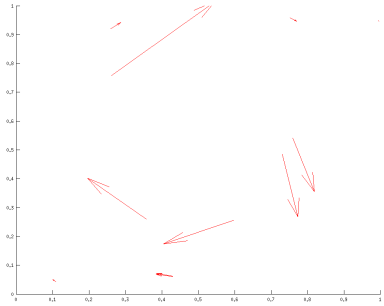
4.6 Comparison with Cross-Correlation at Low Seeding Density

We can compare the errors in the previous section with the ones obtained using a typical cross-correlation algorithm, namely the Direct Cross-Correlation (DCC) algorithm of the PIVlab toolbox in MATLAB [51, 52]. Results with this algorithm for the images under study in this section are presented in Figures 4.10 e) and f). Table 4.7 gives the l^2_{norm} and l^∞ errors in between the exact field (4.9) and the fields of Figures 4.10 e) and f). The reader should keep in mind that both methods give rise to different numerical grids, and thus the norms are computed on their respective grids and should be considered as approximations of the underlying continuous L^2_{norm} and L^∞ norms. However, for interpolation purposes, it is worth noticing that the DCC algorithm needs a coarser grid than the OT algorithm to achieve a similar error for the current experiment. Another observation arising from this comparison is that the OT algorithm gives less spurious vectors than the DCC before post-processing is applied to either of them. Also, the l^2_{norm} errors are better for the OT method than the DCC before any filtering is applied, but when both methods' respective filters are used, the l^2_{norm} errors become almost the same.

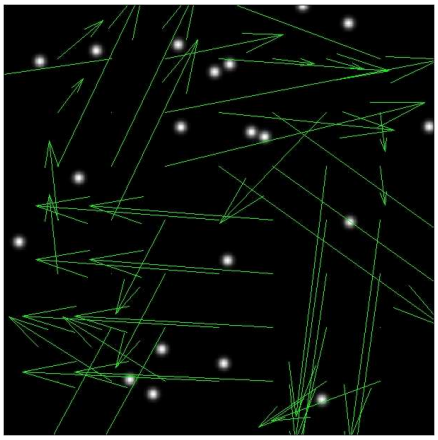
In addition, cross-correlation algorithms are not always accurate for a region at low seeding density, or equivalently for high-resolution fields. The OT procedure appears to perform better in those cases. In order to demonstrate this, we have generated 3 images with randomly placed tracers of equal weights, and we let them evolve in a preselected field, a vortex similar to (4.9). We again used the DCC algorithm of



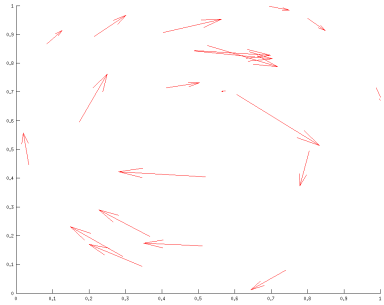
(a) 10 tracers, results of DCC



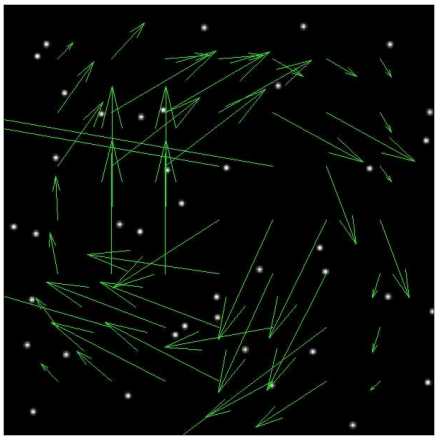
(b) 10 tracers, results of OT



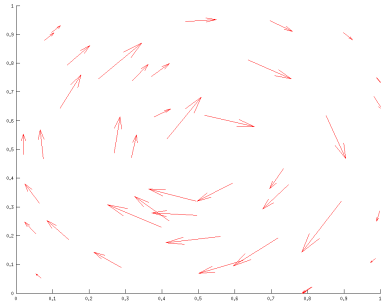
(c) 20 tracers, results of DCC



(d) 20 tracers, results of OT



(e) 40 tracers, results of DCC



(f) 40 tracers, results of OT

Figure 4.11: Results of the comparison experiment with cross-correlation at low seeding density.

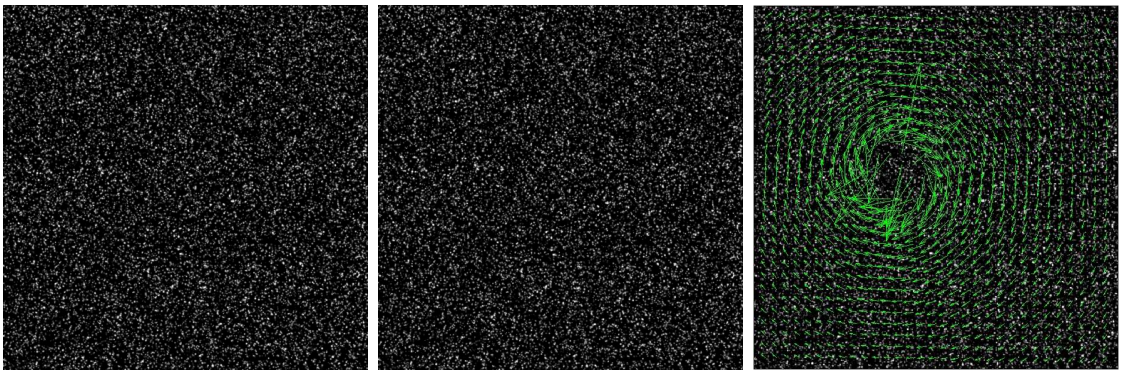
MATLAB's PIVlab toolbox as a cross-correlation algorithm to compare with [51, 52]. The results are presented in Figure 4.11. We see that in every case, the OT procedure produced less spurious vectors than the cross-correlation method employed. These errors are a direct result of the use of interrogation windows: particles moving from one window to another are more likely to cause a mismatch at low seeding density, even when multiple passes to move the location of the interrogation windows are employed. Due to the low seeding density, these errors remained present even when the size of the interrogation window was varied for the current experiment. When the seeding density is increased, we see that less spurious vectors are produced by the cross-correlation method because the effect of one particle crossing windows is diminished by the other particles which remain in the same window (possibly only after the window has been moved). Note that to ensure the fairness of this qualitative comparison, no post-processing was applied to any of the fields presented in Figure 4.11 (for both OT and DCC).

4.7 Realistic Data

We now present a strategy to apply our technique to more realistic PIV data and display the results of numerical experiments. By more realistic, we mean that we directly take PIV grayscale images similar to the ones in Figure 2.2 as densities f and g instead of defining them with our model (3.1).

As discussed in Chapter 1, this strategy does not consist in first identifying meaningful particles from the images and then solving the discrete assignment problem. What we do instead is to solve the continuous OT problem directly with f and g and then select a subset of this OT field associated with particle's locations. The identification of particles is thus performed after running the algorithm. The main reason behind this choice is the difficulty in solving the discrete OT problem for large numbers of particles. Indeed, if N_p is the number of particles involved, then the discrete OT problem scales as $\mathcal{O}(N_p^3)$ whereas the continuous method employed here uses $\mathcal{O}(N\text{Log}N)$ operations, where N is the numerical grid size (more details are given in [39]). It should nonetheless be noted that N increases rapidly with the dimension of the problem.

Let us now describe the strategy employed in more details. First, when there is too much noise coming from out-of-plane loss of particles in the images, we smooth them with a cutoff in frequency space used as a low-pass filter. More specifically,



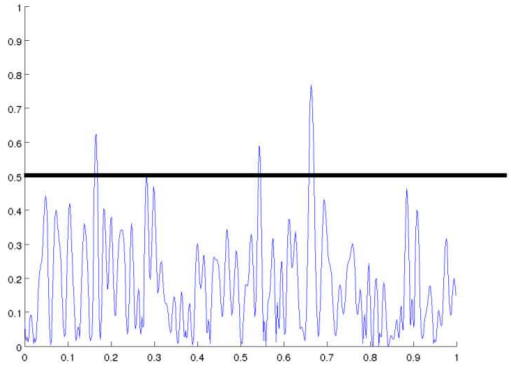
(a) Initial image (b) Final image (c) Results of DCC algorithm

Figure 4.12: Images B3 for Test 5. Figure c) displays the result of the multi-pass DCC algorithm of the PIVlab applied to a) and b) (the window size is 32 pixels and the step size is 16 pixels). A Gaussian 2X3 points subpixel approximation was also employed.

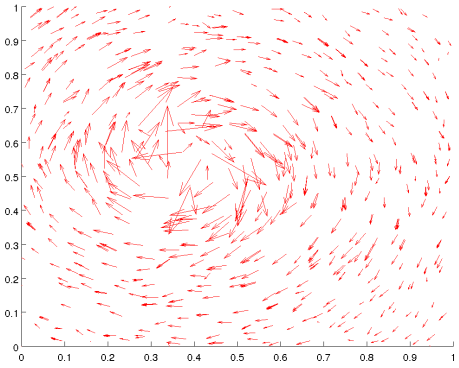
we cut the higher frequencies of both images by computing the Discrete Fourier Transform (DFT) of the images using the FFT algorithm. We then truncate the data in frequency space by keeping only the frequencies located inside a square centered at the origin with sides of length twice a certain proportion α_t of the Nyquist frequency [41].

After applying the OT algorithm to the two (possibly smoothed) densities, we obtain a vector field where every vector corresponds to a grid point on the numerical grid. However, many of these grid points do not correspond to a particle's location and for general fluid flows, the vectors obtained using OT would not necessarily be a good approximation of the velocity field at that location as discussed in Section 2.3 and Section 4.3. Moreover, according to the results of Section 3.2.3 and Section 3.2.4, the vectors obtained at particle locations should be more faithful for the brighter particles. Let us therefore introduce the post-processing parameter $\beta \in [0, 1]$ which we take as a brightness threshold used to identify the brighter particles on the initial image. We then filter the OT vector field to keep only the vectors associated to a point on the image with brightness level higher than β, because these points will correspond to brighter particles on the image. The value of β will have to be different depending on the strength of the smoothing employed and depending on the accuracy desired for the vectors obtained in the approximate field. We will also apply the No Split and the PTPC filters to the approximate fields.

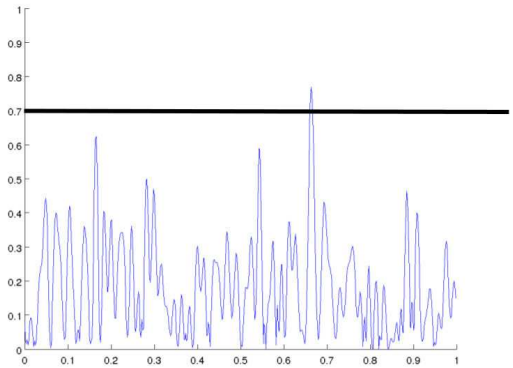
Let us now present the results of a series of numerical experiments on two test



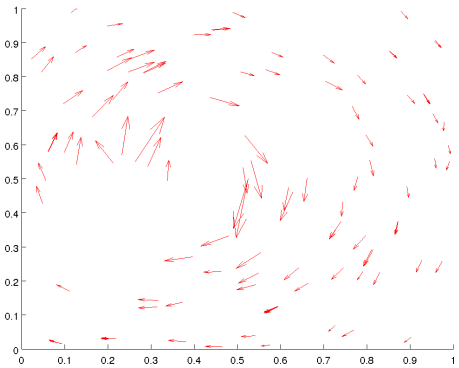
(a) Cross-section of Figure 4.12 a) with $\alpha_t = 0.25$ and $\beta = 0.5$.



(b) Result of the OT algorithm applied to the images with $\alpha_t = 0.25$ and $\beta = 0.5$



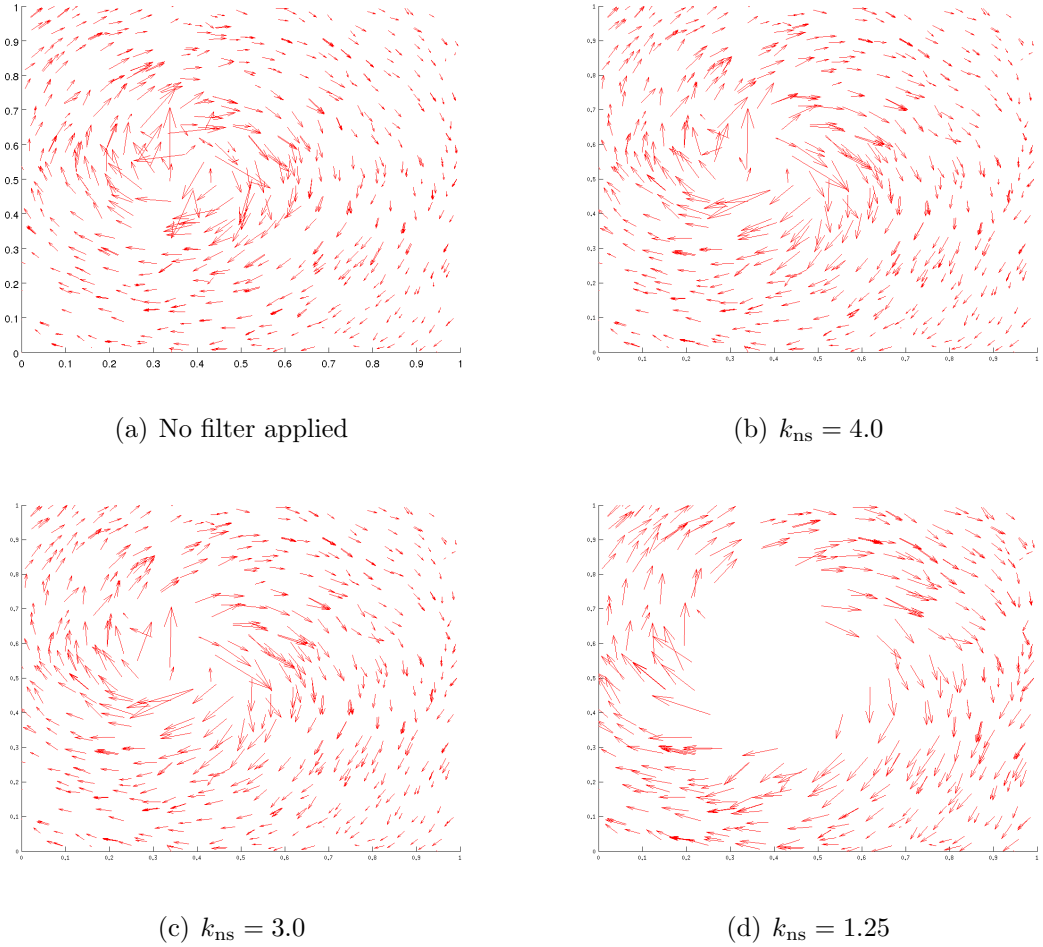
(c) Cross-section of Figure 4.12 a) with $\alpha_t = 0.25$ and $\beta = 0.7$.



(d) Result of the OT algorithm applied to the images with $\alpha_t = 0.25$ and $\beta = 0.7$

Figure 4.13: Results for Test 5. The OT algorithm was applied to the smoothed images of Figure 4.12 with $\alpha_t = 0.25$ and different values of β.

cases. For the first case, we selected images from the 2001 PIV challenge case B [9, 45]. These images were created synthetically to simulate PIV-like images generated by particles evolving in a strong vortex, with typical out-of-plane loss of particles and varying particle sizes. Several sets of images were created with different seeding densities and particle sizes to provide tests for PIV algorithms. In Test 5, we focus on the case B3 which corresponds to a medium seeding density and to small particles. Then, in Test 6, we compare results for all the different scenarios of the 2001 PIV Challenge B (cases B1 to B6). Finally, in Test 7, we use our method on a set of real PIV images showing a slightly turbulent air flow with small water droplets [60].



(a) No filter applied

(b) $k_{\text{ns}} = 4.0$

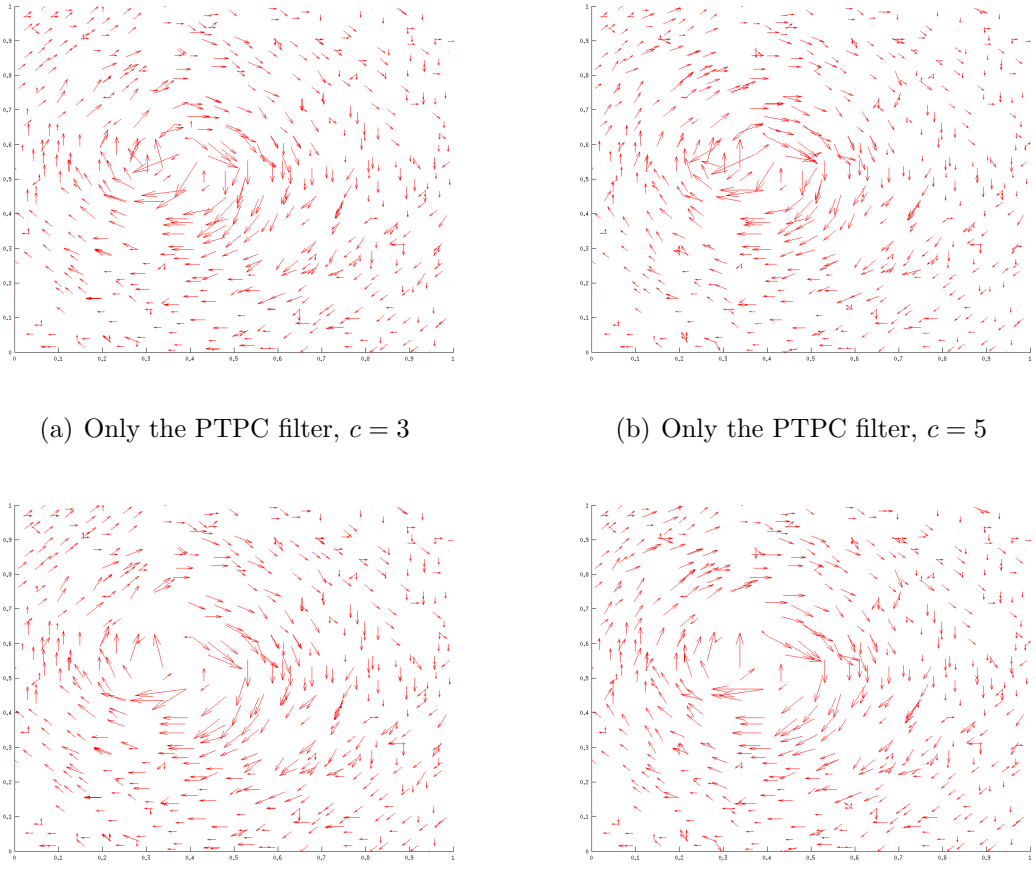
(c) $k_{\text{ns}} = 3.0$

(d) $k_{\text{ns}} = 1.25$

Figure 4.14: Results for Test 5. The No Split filter was applied to Figure 4.13 b) (which is also displayed here), for different parameter values.

Test 5

We consider for this first realistic test a vortex created to simulate real PIV-like images which contain different brightness intensity levels and out-of-plane loss of particles, as proposed by the 2001 PIV Challenge case B3 [9, 45]. The case B3 actually corresponds to a medium seeding density with small particles. Figure 4.12 presents the two 512×512 pixels test images representing the initial and final distributions of particles in the synthetic fluid as well as the resulting field obtained using the DCC Cross-Correlation algorithm of the PIVlab toolbox in MATLAB [51, 52]. In Figure 4.13, we present the results of the OT algorithm for this case, for different values of β. When a higher brightness cutoff $\beta = 0.7$ is selected, fewer vectors remain in the final field, but these vectors more accurately follow the underlying target vector field.



(a) Only the PTPC filter, $c = 3$

(b) Only the PTPC filter, $c = 5$

(c) Both No Split and PTPC filters, $k_{\text{ns}} = 3.0$ and $c = 3$

(d) Both No Split and PTPC filters, $k_{\text{ns}} = 3.0$ and $c = 5$

Figure 4.15: Results for Test 5. The No Split and PTPC filters were applied to Figure 4.13 b), for different parameter values.

As this threshold is lowered to $\beta = 0.5$, more vectors are selected at the price of a loss in accuracy for some of them. Therefore, as anticipated, vectors corresponding to brighter particles are more accurate.

We now look at the effect of the No Split filter in Figure 4.14. We see that higher values of the filter's strength k_{ns} result in several spurious vectors effectively removed. However, increasing the strength of the filter (decreasing k_{ns}) removes most of the vectors close to the center of the vortex, which is understandable since we are getting near the stagnation point where particles are moving much faster and mismatches by the OT algorithm are thus more likely to occur. Notice that the cross-correlation method also experiences difficulties close to the vortex's center.

We also present the result obtained by the PTPC post-processing filter in Figure

2001 PIV Challenge B		
Case	Seeding Density	Particle Size
B1	High	Small
B2	High	Large
B3	Medium	Small
B4	Medium	Large
B5	Low	Small
B6	Low	Large

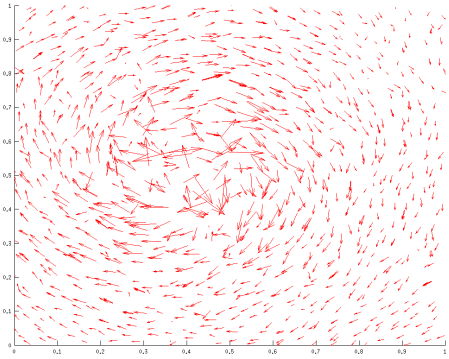
Table 4.8: 2001 PIV Challenge case B description.

4.15. As we have seen previously, this filter improves the overall quality of the approximation, but introduces significant errors, especially for vectors on the outer edge of the vortex for particles whose displacement is on the order of one pixel. Indeed, as we associate the tracers' end positions with brightest neighboring pixels, we loose the subpixel precision given by the transport algorithm. This could potentially be fixed by a subpixel approximation similar to the ones employed in traditional cross-correlation techniques.

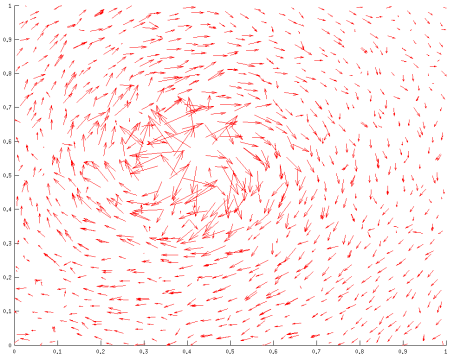
Test 6

The goal of the 2001 PIV challenge case B was to determine the accuracy of a PIV processing method applied to a strong vortex with respect to varying particle sizes and seeding density. The case B3 analyzed so far corresponds to medium particle density and small particles. Let us now try the OT method on the other cases, which are detailed in Table 4.7. For all cases, we present in Figure 4.16 ($\beta = 0.5$) and Figure 4.17 ($\beta = 0.7$) results for a weak cutting in frequency space ($\alpha_t = 0.35$), a weak value of k_{ns} and no PTPC filter. We then have in Figure 4.18 the results of a stronger No Split filter ($k_{\text{ns}} = 2$) and in Figure 4.19 the results of the PTPC filter.

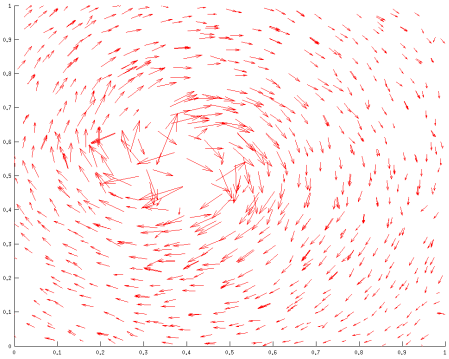
We see that in its current form, the algorithm performs best for low particle density and that it also gives good results in the case of a medium density, but more post-processing is required. The results do not appear to be better for smaller particles than for larger particles, except for the particle end-position correction filter which gives better results in the cases with larger particles when $c = 5$. Note that if the parameter c is reduced to 3, the results for images with smaller particles are better, but still not as good as for larger particles (as previously mentioned, a subpixel approximation would be even better).



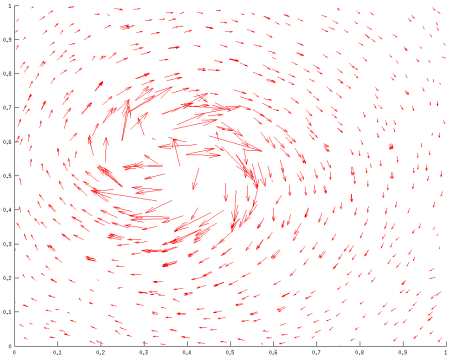
(a) Case B1



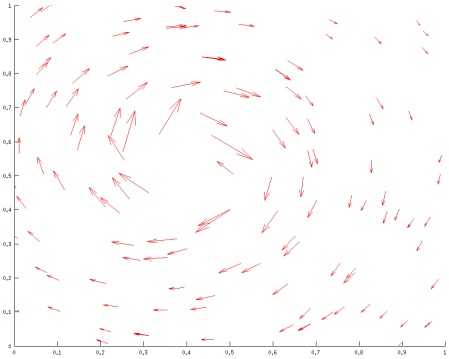
(b) Case B2



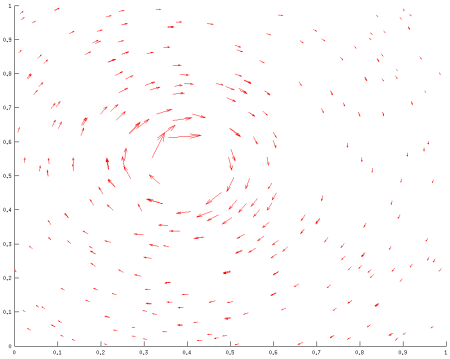
(c) Case B3



(d) Case B4

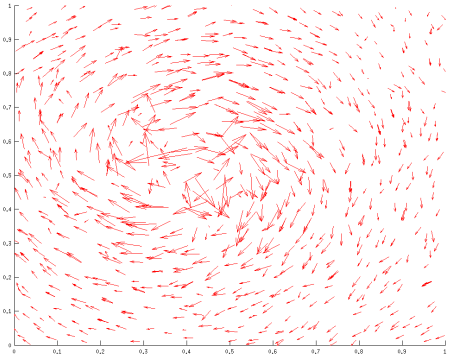


(e) Case B5

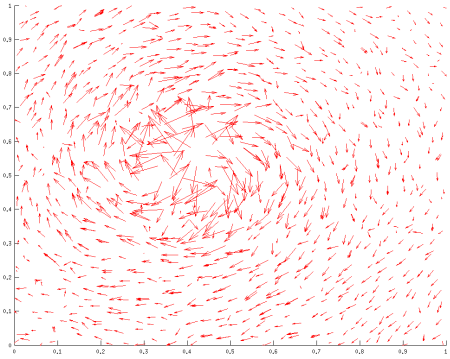


(f) Case B6

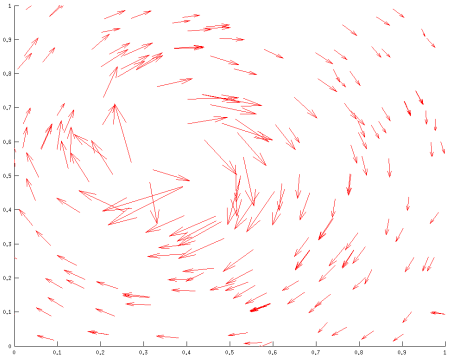
Figure 4.16: Results of Test 6 for $\alpha_t = 0.35$, $\beta = 0.5$, $k_{\text{ns}} = 5.0$ and no PTPC filter.



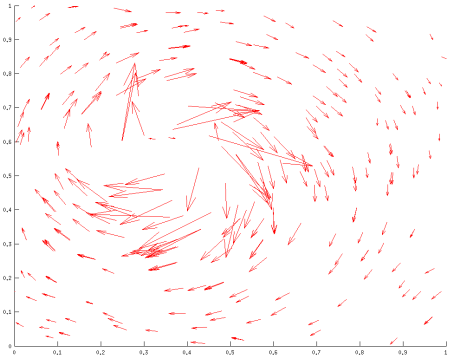
(a) Case B1



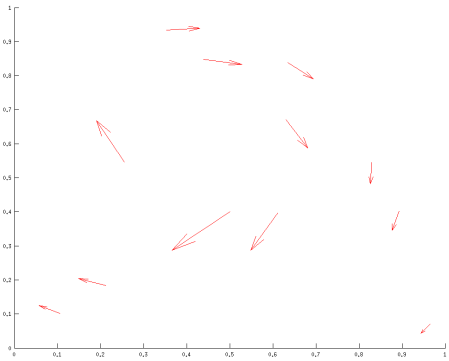
(b) Case B2



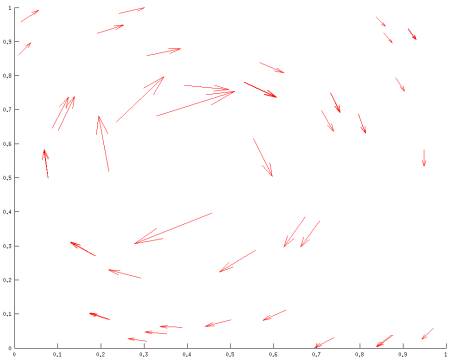
(c) Case B3



(d) Case B4

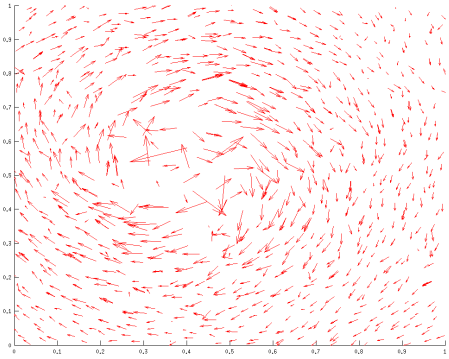


(e) Case B5

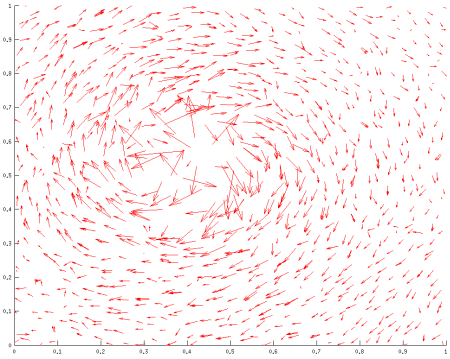


(f) Case B6

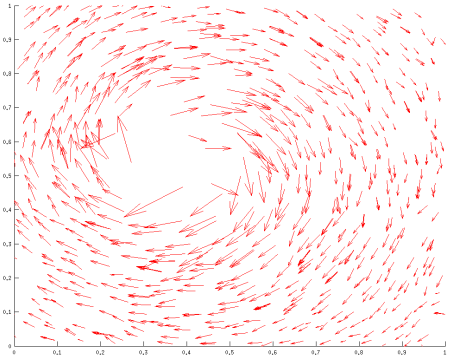
Figure 4.17: Results of Test 6 for $\alpha_t = 0.35$, $\beta = 0.7$, $k_{\text{ns}} = 5.0$ and no PTPC filter.



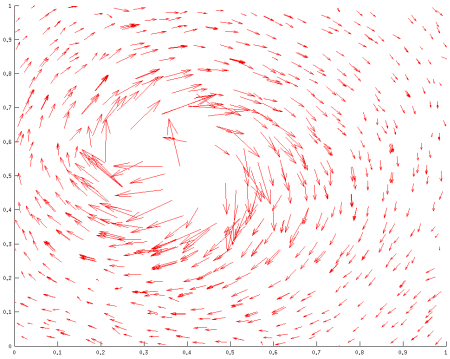
(a) Case B1



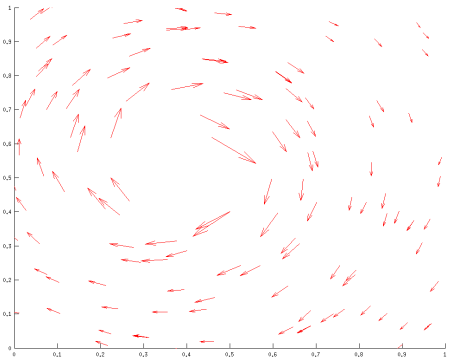
(b) Case B2



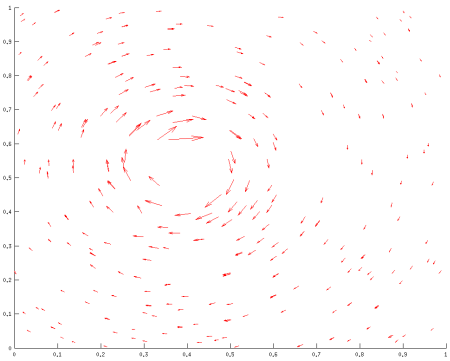
(c) Case B3



(d) Case B4

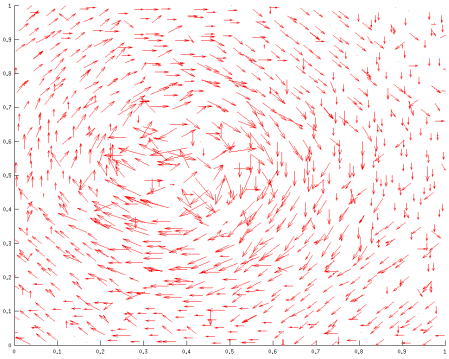


(e) Case B5

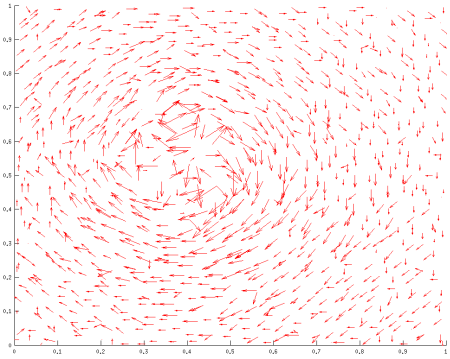


(f) Case B6

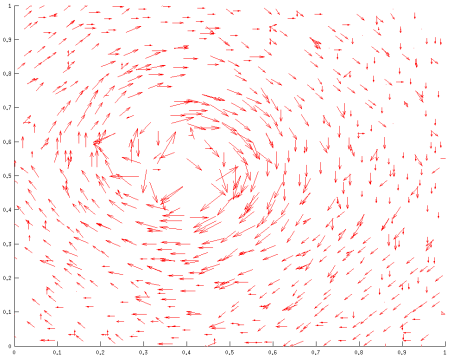
Figure 4.18: Results of Test 6 for $\alpha_t = 0.35$, $\beta = 0.5$, $k_{\text{ns}} = 2.0$ and no PTPC filter.



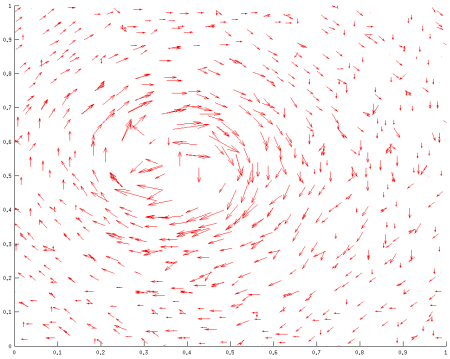
(a) Case B1



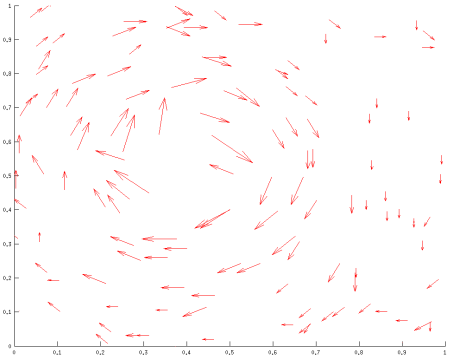
(b) Case B2



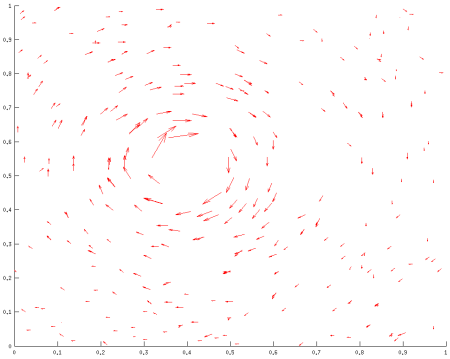
(c) Case B3



(d) Case B4

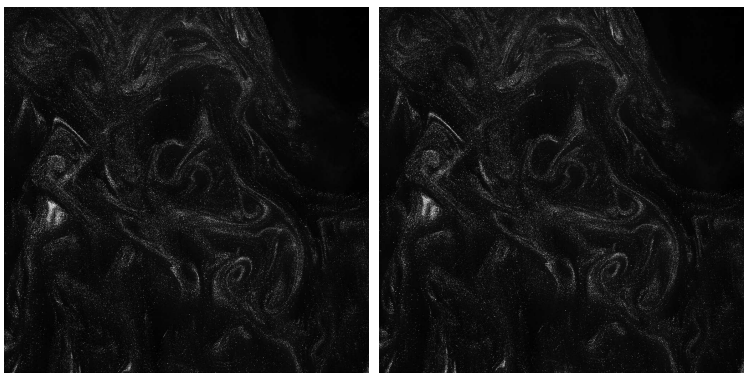


(e) Case B5

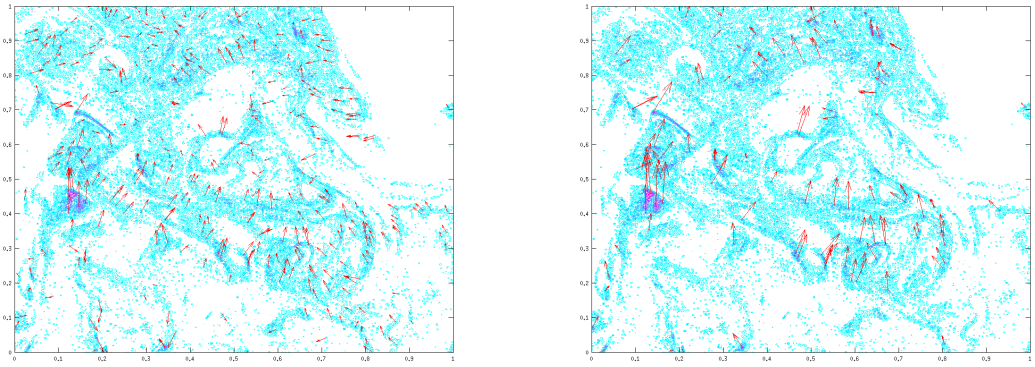


(f) Case B6

Figure 4.19: Results of Test 6 for $\alpha_t = 0.35$, $\beta = 0.5$, $k_{\text{ns}} = 5.0$ and $c = 5$.



(a) First image. (b) Second image. (c) Result of PIVlab's DCC algorithm.

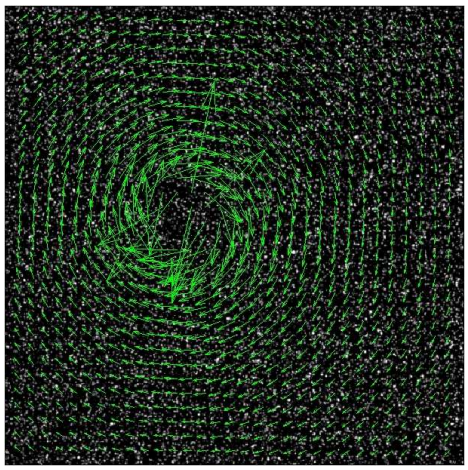


(d) $\beta = 0.20$, $k_{\text{ns}} = 2.0$ (e) $\beta = 0.35$, $k_{\text{ns}} = 2.0$

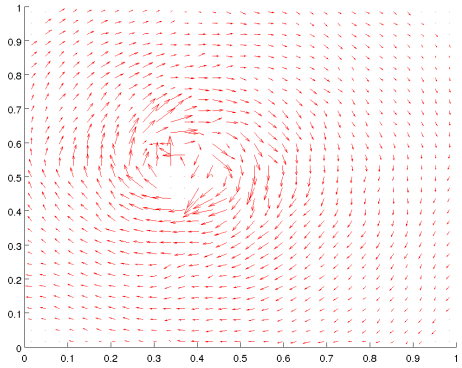
Figure 4.20: Results for Test 7 where the OT algorithm is applied to the first two images of package P3 of [60] with $\alpha_t = 0.35$.

Test 7

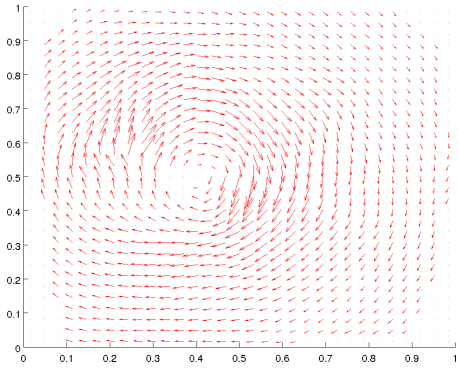
In this final test, we select data from a real laboratory experiment. The images we used are the first two in the sequence given in [60], which consists of PIV images showing a slightly turbulent air flow seeded with small water droplets. We observe in Figure 4.20 that the results are better for brighter particles, as expected. The vectors obtained are not too far from the field obtained using the cross-correlation algorithm, although larger errors are present close to the boundary due to the periodic boundary conditions employed in the resolution of the transport algorithm. This boundary issue could be dealt with by embedding the physical domain containing particles in a larger numerical domain, thus padding the boundaries with a region containing no particles. Note that the current algorithm does not handle vanishing densities, and therefore a little bit of noise would have to be added to the buffer region to get convergence.



(a) Results of DCC algorithm



(b) Linear interpolation of Figure 4.13 b) on a uniform grid



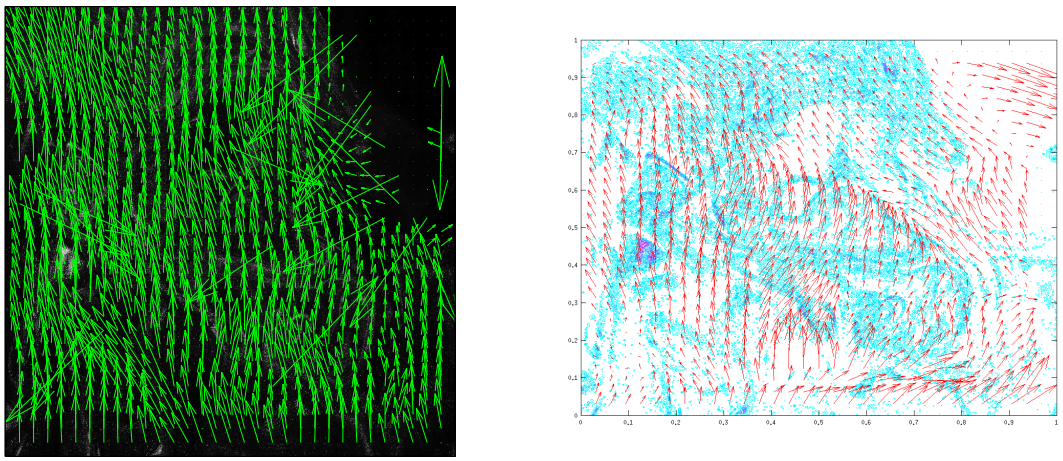
(c) Linear interpolation of Figure 4.13 d) on a uniform grid

Figure 4.21: Results of the linear interpolations of Figure 4.13 b) and d) with a Delaunay triangulation, along with the DCC results for the same images.

However, algorithms to solve the L^2 OT problem in the case where f and g are allowed to vanish in parts of the domain have been recently developed [3] and could be applied in this situation.

4.8 Linear Interpolation

The vector fields obtained in the previous numerical tests with the OT method are all non-uniformly sampled because the brighter particles are not necessarily uniformly distributed. However, it is sometimes useful to have access to an approximate field defined everywhere on the domain in order to help visualize the quality of the results



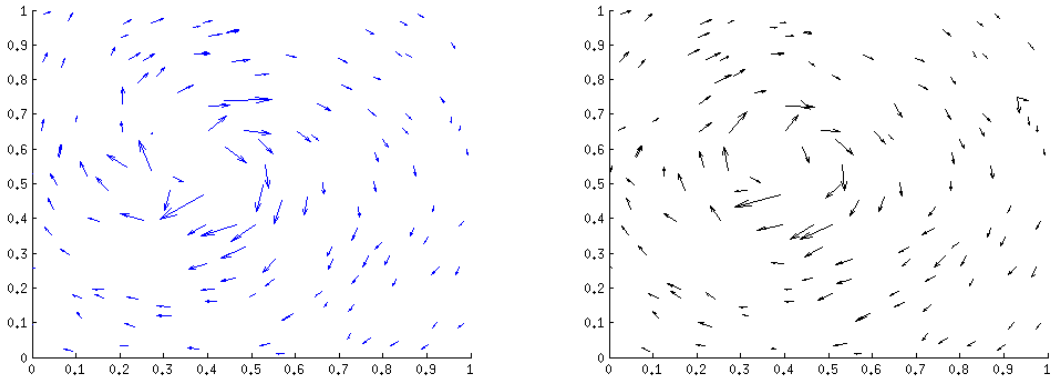
(a) Result of PIVlab's DCC algorithm.

(b) Linear interpolation of Figure 4.13 e) on a uniform grid.

Figure 4.22: Results of the linear interpolations of Figure 4.20 with a Delaunay triangulation, along with the DCC results for the same images.

or to use a multi-pass algorithm akin to the ones used with typical PIV methods (we will present such a method in the next section). Therefore, in order to extend the fields obtained to a uniform grid defined on the whole domain, we present here a first simple method to perform this interpolation. We use MATLAB's TriScatteredInterp function which builds the Delaunay triangulation associated with the data points. This triangulation creates a surface which we use to approximate each component of the vectors in between the locations corresponding to the selected brighter particle images.

Figure 4.21 shows the velocity fields obtained by interpolating linearly the OT fields of Figure 4.13. We see from the results that the center of the vortex is close to the one predicted by the cross-correlation algorithm, but that the strength of the vectors close to the center of the vortex are better captured by the DCC method. We can also see in Figure 4.22 the results of the interpolation of Figure 4.20 e). Again, upon visual observations, the results appear close to what is given by cross-correlation. Note that we will introduce in Chapter 5 a better technique to extend non-uniform fields to the whole fluid domain.



(a) First pass (b) Second pass

Figure 4.23: Result of the multipass algorithm applied to PIV challenge 2001 B3 with $\alpha_t = 0.35$, $\beta = 0.7$, $k_{\text{ns}} = 3.0$ and no PTPC correction. For the second pass, the image was divided into a uniform grid of 4×4 interrogation windows.

4.9 Multi-Pass Algorithm

As we observed so far, the optimal map T transporting f onto g in the L^2 OT problem provides a good approximation of the particles' true displacements, as long as the time interval Δt between the two snapshots is not too large. Indeed, if Δt is small enough so that the average particle displacement is no greater than about half of the average distance between particles (in other words, if we are in the realm of PTV - the tracking range), then a direct computation of $(T(x) - x)/\Delta t$ yields a good approximation of the fluid's velocity field at the particles' locations. In addition, if Δt is larger and the particles are displaced over longer distances, it is still possible to follow the brighter particles with the OT map.

However, in order to correct mismatches and possibly to be able to track particles that are not as bright over longer distances, one approach would be to use a multiresolution method akin to the ones typically employed for cross-correlation algorithms. In fact, as discussed in Section 2.2, modern PIV methods based on cross-correlation usually use some kind of multiresolution method, where for example several passes of the algorithm are used to "offset" the interrogation windows in the direction of the flow's motion [36]. In our case, we could use the OT fields obtained previously as a first pass, split the images into interrogation windows, displace the windows according to the first pass and reapply the transport algorithm on the individual windows. More precisely, we could use the following procedure:

Multipass Algorithm

1. Compute the OT field and corresponding post-processing filters on the given images.

2. Interpolate the non-uniform field on the whole domain, using for example the procedure of Section 4.8.

3. Split the two images into interrogation windows.

4. Displace the center of each interrogation window in the target image according to the field obtained in 1.

5. Compute the OT field and corresponding post-processing filters on each pair of initial and displaced final interrogation window.

6. Create a final field by adding the windows displacement vectors to the ones obtained in 5.

Let us test this strategy on one of the previously tested examples, namely the vortex provided by the 2001 PIV challenge case B3. Results for $\alpha_t = 0.35$, $\beta = 0.7$, $k_{\text{ns}} = 3.0$ and no PTPC correction are presented in Figure 4.23. The second pass was obtained by splitting the images into a 4×4 uniform grid of interrogation windows and by applying the multiresolution approach discussed here. We observe from the images that the second pass corrects several mismatches and appears to improves the quality of the approximation. These results show that it is feasible to employ similar strategies for the OT method than the ones that are commonly used with cross-correlation for PIV. More involved strategies could also be designed, where for example the brightness threshold β is lowered at every pass, thus obtaining a field defined at more points in the domain. However, a more detailed analysis of the effect of these multiple passes is beyond the scope of this work.

Chapter 5

Effective Filtering with Navier-Stokes Equations

5.1 Optimization Problem

The vector fields obtained with Optimal Transport in Chapter 4 provide a good approximation of the underlying flow velocity field on the non-uniform grid given by the particle locations. However, for points outside theses locations, the field obtained with the OT method is not physically accurate (see Section 2.3 and Section 4.3). In addition, the velocities given by the OT field are equivalent to first-order in time finite differences approximations of the velocities on the interval $[0, \Delta t]$ between two laser pulses. It is therefore natural to seek an extension of v_T on all of $\Omega \times [0, \Delta t]$ that would provide a better approximation of the target flow. While the interpolation technique of Section 4.8 does provide a way to extend the non-uniform field to the whole physical domain, the resulting field remains stationary on $[0, \Delta t]$ and may not be physically accurate.

Many methods have been developed to extend discretely sampled fields in the context of fluid flows [21, 22, 24, 34, 38, 43, 44, 49, 56]. Some of these methods use triangulations or tessellations [43] or splines [24, 44]. Other combine Computational Fluid Dynamics (CFD) techniques such as Direct Numerical Simulations (DNS) with PIV or PTV measurements [49]. Given that the discrete field under study is likely to contain noisy data, one may formulate this problem in the context of data assimilation. Out of all the various techniques this field suggests [22], variational approaches have given successful procedures to perform such extensions and filtering by using

the Navier-Stokes equations [21, 34, 38, 56]. While some of these methods have been developed to handle entire sequences of images [34, 38], to be consistent with the rest of this work, we focus only on the time interval $[0, \Delta t]$ corresponding to the capture of two successive images. In addition, as opposed to [34, 38], we do not wish to use information given by the PIV images directly: we want to rely solely on the approximated stationary vector field v_T in order to make the technique applicable to a potentially wider range of problems. In [56], the authors have considered a similar situation where a corrupted velocity field is available only at one point in time. However, they assumed the flow was quasi-stationary on $[0, \Delta t]$ and thus neglected the time derivatives. Our goal here is to design a method which allows non-stationary flows and which also performs well when only a few vectors are available (i.e. when the non-uniform grid on which v_T is defined is sparse - this is the case when only the brighter particles are selected in the OT-PIV procedure).

Let us now describe our method. Consider $v_T\big|_P$ be the OT field restricted to the bright particle locations $P = \{p_1, p_2, \ldots, p_{N_p}\} \subset \Omega$. Ideally, the resulting field $v_E(x,t)$ obtained by extending $v_T\big|_P$ to the entire physical domain $\Omega \subset \mathbb{R}^2$ on the time-interval $[0, \Delta t]$ would be a solution of the incompressible Navier-Stokes equation on Ω:

$$\begin{cases} \dfrac{\partial v}{\partial t} + (v \cdot \nabla)v = -\nabla p + \nu \Delta v \\ \\ \operatorname{div}(v) = 0 \\ \\ v(x,0) = v_0(x),\ x \in \Omega,\ t \in [0, \Delta t], \end{cases} \tag{5.1}$$

where p is the fluid's pressure, ν the viscosity and $v_0(x)$ the initial velocity field. However, there might be many solutions of (5.1) passing through (at some time $t^\star$ in $[0, \Delta t]$) the prescribed vectors $v_T(p_i)$ given by $v_T\big|_P$ at the brighter particle's locations p_i, especially if the non-uniform grid given by these locations is sparse. One might therefore want to enforce a constraint to select a solution v_E which does not add too many artificial features to the field. A natural choice is to pick a solution with a small kinetic energy

$$\int_{[0,\Delta t]} \int_\Omega |v(x,t)|^2 \, \mathrm{d}x \, \mathrm{d}t. \tag{5.2}$$

Also, given that $v_T\big|_P$ still possibly contains erroneous vectors which were not corrected or removed by the post-processing filters introduced in Chapter 4, it might be too stringent of a condition to perfectly match v_E and $v_T\big|_P$ at some time(s) $t^\star$ on $[0, \Delta t]$.

We thus choose to minimize the total error

$$\sum_{i=1}^{N_p} \left| v\left(p_i, t^\star\right) - v_T(p_i) \right|^2 . \tag{5.3}$$

To summarize, we are looking for a spatial and temporal extension $v_E(x,t)$ of $v_T\big|_P$ on $\Omega \times [0, \Delta t]$ which solves (5.1) and minimizes both (5.2) and (5.3).

Let us therefore build an optimization problem to find such a vector field v_E. First, we point out that it is sufficient to minimize the initial kinetic energy of v_E since the kinetic energy (5.2) of the solution to the Navier-Stokes equations (5.1) is time-decreasing [29]. Also, one common and easier way to solve the Navier-Stokes equations in 2D is to consider the associated vorticity equation

$$\begin{cases} \dfrac{\mathrm{D}\omega}{\mathrm{D}t} = \nu \Delta \omega, \\ \omega(x,0) = \omega_0(x),\ x \in \Omega,\ t \in [0, \Delta t], \end{cases} \tag{5.4}$$

where $\omega = \nabla \times v$ is the vorticity of the field [29]. The velocity v is then determined from the vorticity, using the Biot-Savart law [29]

$$v(x,t) = \int_\Omega K_2(x-y)\omega(y,t)\,\mathrm{d}y,$$

where K_2 is the 2D Biot-Savart Kernel on Ω, and the pressure is determined by the Poisson equation

$$-\Delta p = \mathrm{tr}(\nabla v)^2 = \sum_{1 \le i,j \le 2} \frac{\partial v_i}{\partial x_j} \frac{\partial v_j}{\partial x_i}.$$

We also point out that, if we let ψ be the stream function for the fluid, then the Biot-Savart law is derived from the relationship between the velocity and stream function

$$v = \nabla^\perp \psi = \left(-\frac{\partial \psi}{\partial x_2}, \frac{\partial \psi}{\partial x_1} \right)$$

as well as the relationship between the vorticity and the stream function $\omega = \Delta \psi$. For simplicity, we will therefore denote $v = \nabla^\perp \Delta^{-1} \omega$ in the next sections. Let us now introduce the following optimization problem:

Minimization Problem:

$$\text{Min } F[\omega_0] = k_e \int_\Omega |v_0(x)|^2 \, \mathrm{d}x + \sum_{i=1}^{N_p} \left| v\left(p_i, t^\star\right) - v_T(p_i) \right|^2 \tag{5.5}$$

for $\omega_0 \in \mathcal{C}^\infty(\Omega)$. Here, v_0 and v depend on ω_0 through $v = \nabla^\perp \Delta^{-1} \omega$ and (5.4), and k_e is a constant.

Essentially, the goal here is to find, among all suitable initial vorticities in Ω, the one that will give the velocity field v_E which minimizes F. Note that we will present reasons as to why F is a strictly convex function of ω_0 for small Δt and for every $k_e \in (0, \infty)$ in the next section, but for now we assume that F has a unique minimizer. On top of making the functional strictly convex for small $t^\star$ as we will see in the next section, the parameter k_e allows flexibility in this solution by controlling the size of the kinetic energy with respect to the error with the prescribed field. Indeed, a solution corresponding to a small k_e should be very faithful to $v_T\big|_P$ but may have too much kinetic energy and thus may present too turbulent eddies. On the other hand, a solution associated with a larger k_e will have less kinetic energy, at the expense of allowing v_E to be further from v_T at the particle locations. While the time $t^\star$ at which we want the vector fields to be close to each other may appear arbitrary, as we will select $t^\star = \Delta t/2$ for numerical experiments, one can think of (5.3) as a midpoint approximation of the time-integral of this error term over the interval $[0, \Delta t]$. Also, by considering the minimization of the total error (5.3), one allows the optimal solution to not go through every single $v_T(p_i)$ and thus potentially smooth out some of the noise. Finally, it is worth mentioning that v_E could be used to extend (and possibly filter) other vector fields: it is not limited to extending the OT field v_T. One could definitely use this procedure to extend a field obtained with a cross-correlation algorithm which, while defined on a uniform grid of interrogation windows, remains a discrete vector field.

5.2 Convexity and Linearization

Before we attempt to solve problem (5.5), let us first comment on the convexity of F. We introduce the following notation:

$$F_1(\omega_0) = \int_\Omega |v_0(x)|^2 \, \mathrm{d}x = \int_\Omega |\nabla^\perp \Delta^{-1} \omega_0(x)|^2 \, \mathrm{d}x,$$
$$F_2(\omega_0) = \sum_{i=1}^{N_p} |v\,(p_i, t^\star) - v_T(p_i)|^2 = \sum_{i=1}^{N_p} \left|\nabla^\perp \Delta^{-1} \omega\,(p_i, t^\star) - v_T(p_i)\right|^2.$$

Assume also that Ω is a closed and bounded subset of $\mathbb{R}^2$. We quickly see that the function $F_1(\omega_0)$ is strictly convex since $\nabla^\perp \Delta^{-1}$ is a linear operator [14] and since F_1 is strictly convex with respect to v_0. In addition, F_2 is convex when $t^\star = 0$ for similar reasons, but it is not necessarily strictly convex. Indeed, there may be more than one vector field v solving the Navier-Stokes equations and exactly equal to v_T at all the points p_i, especially for small N_p. The function F is therefore strictly convex when $t^\star = 0$ as it is the sum of a convex function F_2 and strictly convex function $k_e F_1$. However, the situation is not as clear when $t^\star > 0$. To investigate the convexity of F in this case, denote the solution of (5.4) by $\omega(t) = S_t \omega_0$ where S_t is the flow operator associated with the solution of (5.4). Note that S_t satisfies $S_0 = \mathrm{Id}$, and let

$$[F(\omega_0)](t) = k_e F_1(\omega_0) + \sum_{i=1}^{N_p} \left|\nabla^\perp \Delta^{-1} S_t\, \omega_0(p_i) - v_T(p_i)\right|^2.$$

The strict convexity of F at $t = 0$ means that the Hessian of $F(\omega_0)(0)$ is strictly positive definite. We argue that this remains true for $F(\omega_0)(t)$ for t small enough because of the continuity of the Hessian $D^2_{\omega_0} F_2(\omega_0)(t)$ with respect to t.

Indeed, the chain rule implies that ij-th entry of this matrix

$$\frac{\partial^2}{\partial x_i \partial x_j}[F_2(\omega_0)(t)](x)$$

depends on the first and second derivatives of F_2 with respect to $\nabla^\perp \Delta^{-1} S_t \omega_0$ as well as those of $\nabla^\perp \Delta^{-1} S_t \omega_0$ with respect to ω_0. Since the former derivatives are clearly continuous, we focus on the latter. Using again the chain rule, we have the following

(formal) computations:

$$\frac{\partial}{\partial x_i}[\nabla^\perp \Delta^{-1} S_t \omega_0](x) = D_\omega[\nabla^\perp \Delta^{-1}\omega(x,t)]\frac{\partial \omega(x,t)}{\partial x_i},$$
$$\frac{\partial^2}{\partial x_i \partial x_j}[\nabla^\perp \Delta^{-1} S_t \omega_0](x) = D^2_\omega[\nabla^\perp \Delta^{-1}\omega(x,t)]\frac{\partial \omega(x,t)}{\partial x_j}\frac{\partial \omega(x,t)}{\partial x_i} + D_\omega[\nabla^\perp \Delta^{-1}\omega(x,t)]\frac{\partial^2 \omega(x,t)}{\partial x_i \partial x_j}.$$

The derivatives $D_\omega \nabla^\perp \Delta^{-1}\omega(x,t)$ and $D^2_\omega \nabla^\perp \Delta^{-1}\omega(x,t)$ both depend continuously on ω. In addition, the regularity theory [29, 50] of the 2D Navier-Stokes equations (5.1) ensures that

$$t \to \omega(x,t), \quad t \to \frac{\partial \omega(x,t)}{\partial x_i} \quad \text{and} \quad t \to \frac{\partial^2 \omega(x,t)}{\partial x_i \partial x_j}$$

are all continuous functions of t for t close to 0, as long as ω_0 is smooth enough (choosing $\omega_0 \in \mathcal{C}^\infty(\Omega)$ as in the next section is definitely sufficient [50]). We therefore conclude that the Hessian $D^2_{\omega_0} F(\omega_0)(t)$ remains strictly positive definite for a small time. This argument justifies the strict convexity of F for small t. However, it is possible that the value of $t^\star = \Delta t/2$ selected is larger than the time t required to maintain this convexity. In that case, taking a larger value of k_e may help the functional F retain its strict convexity with respect to ω_0 by putting more weight on the strictly convex function $k_e F_1$ in the sum giving F. The addition of $k_e F_1$ to F_2 can also be seen as adding a penalty term to the error functional F_2 in the context of penalty methods, ensuring solvability of the optimization problem at the expense of allowing less kinetic energy in the final solution. We can thus claim that F is strictly convex for k_e large enough and Δt small enough.

Next, in order to obtain the numerical solution of this optimization problem within reasonable time, we will employ a steepest descent algorithm. We therefore need to linearize the functional F. Consider a small variation $\omega_0 + \epsilon h$ of ω_0. For F_1, we have

$$\begin{aligned} F_1[\omega_0 + \epsilon h] &= \int_\Omega |\nabla^\perp \Delta^{-1}(\omega_0 + \epsilon h)|^2 \, \mathrm{d}x \\ &= \int_\Omega |\nabla^\perp \Delta^{-1}\omega_0|^2 \, \mathrm{d}x + 2\epsilon \int_\Omega \left\langle \nabla^\perp \Delta^{-1}\omega_0, \nabla^\perp \Delta^{-1} h \right\rangle \mathrm{d}x + \mathcal{O}(\epsilon^2) \end{aligned}$$

where $< \cdot, \cdot >$ is the l^2 inner product in $\mathbb{R}^2$. From this, we obtain the linearization of

F_1 at ω_0 in direction h:

$$DF_1(\omega_0)(h) = 2\int_\Omega \left\langle \nabla^\perp \Delta^{-1}\omega_0, \nabla^\perp \Delta^{-1} h \right\rangle \, \mathrm{d}x. \tag{5.6}$$

For F_2, we have

$$\begin{aligned} F_2[\omega_0 + \epsilon h] &= \sum_{i=1}^{N_p} \left| \nabla^\perp \Delta^{-1} S_{t^\star}(\omega_0 + \epsilon h)(p_i) - v_T(p_i) \right|^2 \\ &\approx \sum_{i=1}^{N_p} \left| \nabla^\perp \Delta^{-1} \left(\omega_0(p_i) + \epsilon h(p_i) + t^\star \frac{\partial}{\partial t} S_t(\omega_0 + \epsilon h)(p_i) \Big|_{t=0} \right) - v_T(p_i) \right|^2, \end{aligned}$$

where in the second line we used a first order in time Taylor expansion to approximate $S_{t^\star}$. Note that this approximation is reasonable since the $\mathcal{O}({t^\star}^2)$ term missing is small in our case due to the small Δt. Using (5.4) to replace the time derivative, we reach

$$\begin{aligned} &\sum_{i=1}^{N_p} |\nabla^\perp \Delta^{-1}\omega_0(p_i) + \epsilon \nabla^\perp \Delta^{-1} h(p_i) \\ &\quad + t^\star \nabla^\perp \Delta^{-1} \Big(\nu \Delta(\omega_0 + \epsilon h) - (\nabla^\perp \Delta^{-1}(\omega_0 + \epsilon h) \cdot \nabla)(\omega_0 + \epsilon h) \Big)(p_i) - v_T(p_i)|^2 \\ &= \sum_{i=1}^{N_p} |\nabla^\perp \Delta^{-1}\omega_0(p_i) + t^\star \nabla^\perp \Delta^{-1} \nu \Delta \omega_0(p_i) \\ &\qquad\qquad - t^\star \nabla^\perp \Delta^{-1}(\nabla^\perp \Delta^{-1}\omega_0(p_i) \cdot \nabla \omega_0(p_i)) - v_T(p_i) \\ &\quad + \epsilon \Big(\nabla^\perp \Delta^{-1} h(p_i) + t^\star \nu \nabla^\perp \Delta^{-1} \Delta h(p_i) - t^\star \nabla^\perp \Delta^{-1}(\nabla^\perp \Delta^{-1} h(p_i) \cdot \nabla \omega_0(p_i)) \\ &\qquad\qquad - t^\star \nabla^\perp \Delta^{-1}(\nabla^\perp \Delta^{-1}\omega_0(p_i) \cdot \nabla h(p_i)) \Big) + \mathcal{O}(\epsilon^2)|^2. \end{aligned}$$

Expanding the norm as an inner product and gathering the linear terms in ϵ then

yields the following formula:

$$
\begin{aligned}
&DF_2(\omega_0)(h) = \\
&2\sum_{i=1}^{N_p} \Bigg\langle \nabla^\perp \Delta^{-1} \Big(\omega_0(p_i) + t^\star \nu \Delta \omega_0(p_i) - t^\star \nabla^\perp \Delta^{-1} \omega_0(p_i) \cdot \nabla \omega_0(p_i) \Big) - v_T(p_i), \\
&\qquad \nabla^\perp \Delta^{-1} \Big(h(p_i) + t^\star \nu \Delta h(p_i) - t^\star \nabla^\perp \Delta^{-1} h(p_i) \cdot \nabla \omega_0(p_i) \\
&\qquad\qquad - t^\star \nabla^\perp \Delta^{-1} \omega_0(p_i) \cdot \nabla h(p_i) \Big) \Bigg\rangle. \qquad (5.7)
\end{aligned}
$$

Finally, we obtain

$$DF(\omega_0)(h) = k_e DF_1(\omega_0)(h) + DF_2(\omega_0)(h), \qquad (5.8)$$

where $DF_1(\omega_0)(h)$ and $DF_2(\omega_0)(h)$ are defined by (5.6) and (5.7), respectively.

5.3 Numerical Algorithm

To obtain a numerical solution of the aforementioned minimization problem (5.5), we use an expansion of the initial vorticity in radial basis functions

$$\omega_0(x) = \sum_{i=1}^{N_b} \alpha_i \frac{e^{-|x-b_i|^2/\epsilon^2}}{\pi \epsilon^2} \qquad (5.9)$$

akin to the one used in the context of vortex blob methods [26]. Here, b_i is the center of blob i, N_b is the number of blobs and $\epsilon > 0$ is a parameter controlling the size of the blobs. However, as opposed to using a Lagrangian discretization of (5.4) which is typical of vortex methods, we instead impose periodic boundary conditions on Ω (to be consistent with the boundary conditions used so far in this work) and use Fourier transforms with a finite differences scheme to solve the resulting first-order in time ODE in Fourier space. When the domain Ω is the two-dimensional torus T^2, the vorticity formulation of (5.1) becomes

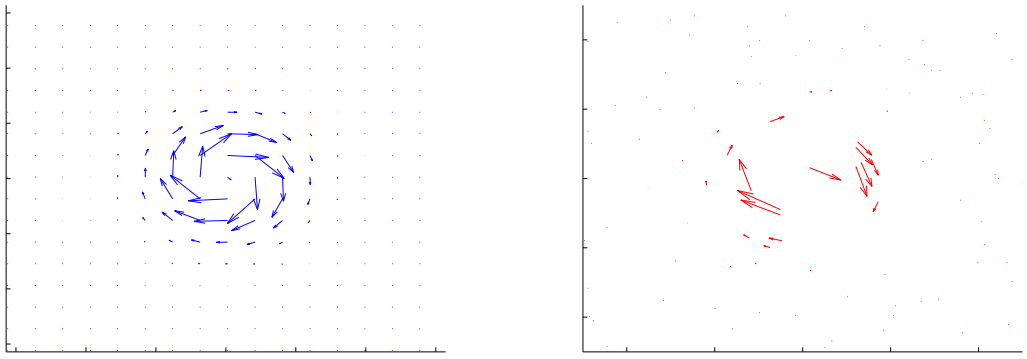
$$
\begin{cases}
\dfrac{\partial \omega}{\partial t} + [(\overline{v_0} + \widetilde{v}) \cdot \nabla] \omega = \nu \Delta \omega, \\
\omega(x, 0) = \omega_0(x),\ x \in \Omega,\ t \in [0, \Delta t]
\end{cases} \qquad (5.10)
$$

where $\omega = \operatorname{curl} \widetilde{v}$, $\overline{v_0} = \int_\Omega v_0 \, \mathrm{d}x$ and $\widetilde{v}$ is recovered from the periodic version of the Biot-Savart law

$$\widetilde{v}(x,t) = \sum_{k \neq 0} \frac{(-k_2, k_1)^t}{2\pi i |k|^2} e^{2\pi i x \cdot k} \widehat{\omega}(k,t) \tag{5.11}$$

and $\widehat{\omega}$ is the Fourier transform of the vorticity ω [29].

Let us now give the specific details of the discretization employed. First, we embed Ω in a larger numerical domain Ω_n to avoid boundary effects, and we impose the periodic boundary conditions on Ω_n instead of Ω. We lay a uniform grid of the centers b_i for the Gaussian blobs in Ω. We then consider (5.5) as an unconstrained minimization problem in $\mathbb{R}_{N_b}$ and use MATLAB's fminunc quasi-Newton algorithm to find the minimum of F with respect to the weights α_i in (5.9). This algorithm uses the gradient information supplied by (5.8) as well as an approximation of the Hessian of F with the BFGS method to iterate in the direction of steepest descent [12]. At each step, given a set of weights $\{\alpha_i\}$, ω_0 is reconstructed on a uniform numerical grid on Ω_n of size $N = N_1 \times N_2$ through (5.9). Then, v_0 is recovered with the periodic Biot-Savart law (5.11) using the FFT algorithm (all the computations of $\nabla^\perp \Delta^{-1}$ required in the derivative of F are done this way). Once v_0 is obtained, the periodic Navier-Stokes equations in vorticity form (5.10) are solved in Fourier space with the FFT and the trapezoidal method for the resulting ODE to get ω and v on the time-domain $[0, \Delta t]$, which is itself discretized with N_t points. After all this, F_2 is computed at $t^\star = \Delta t/2$, all the integrals are approximated by a 2D Simpon's method, and F as well as its derivative are computed with the same techniques. Finally, we use the stopping criterion provided by the fminunc package, which stops the iterations when either the size of the objective function F or the size of a step becomes smaller than the specified tolerance $\mathrm{TOL}_{\mathrm{fminunc}}$. Also, as we are looking for an optimal solution with a minimum kinetic energy, it is natural to take $\omega_0 = 0$ as an initial guess in the quasi-Newton algorithm.



(a) Target Vector Field. (b) Random selection of 100 vectors in a).

Figure 5.1: Fields for Test 1.

5.4 Numerical Experiments

We now conduct several numerical experiments to analyze the behavior of the minimizer of (5.5). In all the following tests, we run the algorithm using a spatial grid size of $N = N_1 \times N_2 = 256 \times 256$ and a temporal grid size of $N_t = 100$, using a fixed $\Delta t = 0.1$. The viscosity is set to $\nu = 1 \times 10^{-6}\, m^2/s$, which is roughly the (kinematic) viscosity of distilled water at 20 degrees Celsius. We will however vary the other parameters (k_e, ϵ, and N_b) to study their effect on the resulting fields. We also stop the algorithm when $\mathrm{TOL}_{\mathrm{fminunc}} = 10^{-6}$. The computations were all performed on a personal laptop computer of type Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz with a parallel implementation using all 4 cores to compute the components of the gradient of F at every step. The computing times for a typical experiment presented in this section (when $N_b = 64$) are around 15 minutes. As most of the work goes into computing the gradient at every step and that each component of the gradient is independent of the others, using more than 4 cores would greatly improve the performances of the current implementation. Also, we point out that our code has not been professionally optimized to minimize the computing time. Finally, in what follows, the images displayed all correspond to the optimal vector fields v_E at time $t^\star = \Delta t/2$ and the various norms are computed with the same fields.

k_e	1000	100	10	1	0.1	0
F_1	0.0000142	0.000763	0.00632	0.0111	0.0128	0.0130
F_2	0.3770	0.2220	0.0428	0.0218	0.0208	0.0208
l^2_{norm} error	0.0688	0.0576	0.0384	0.0392	0.0436	0.0444
e_{total}	0.9485	0.6642	0.2994	0.3101	0.3830	0.3986
l^∞ error	0.3011	0.2838	0.2346	0.2113	0.2228	0.2249

Table 5.1: Results for Test 1 when $\epsilon = 0.05$ and $N_b = 64$. The errors presented are computed for every grid point in Ω.

Test 1

For the first numerical experiment, we consider the vector field on the interval $[0,1]^2$ given by

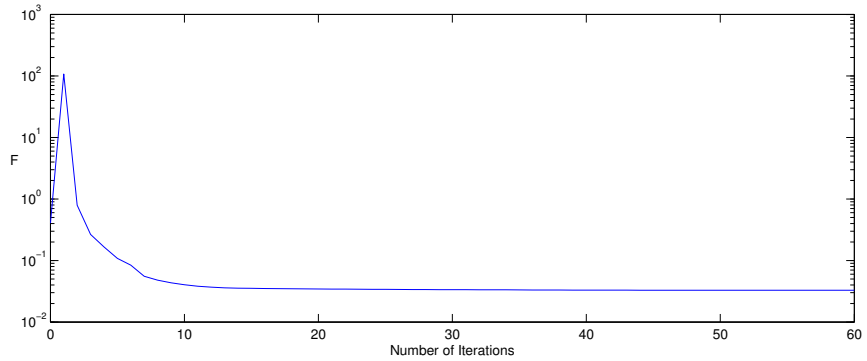
$$v(x_1, x_2) = \Big(x_2 - 0.5, 0.5 - x_1\Big) \exp\Big(- 2((x_1 - 0.5)^2 + (x_2 - 0.5)^2)\Big), \qquad (5.12)$$

which is similar to the field (4.9) used in Test 4 of Chapter 4 (we made the size of the vortex bigger in order to better display the properties of the method under study here). This vector field cannot be recovered exactly with a single term in the expansion (5.9) since its curl is given by

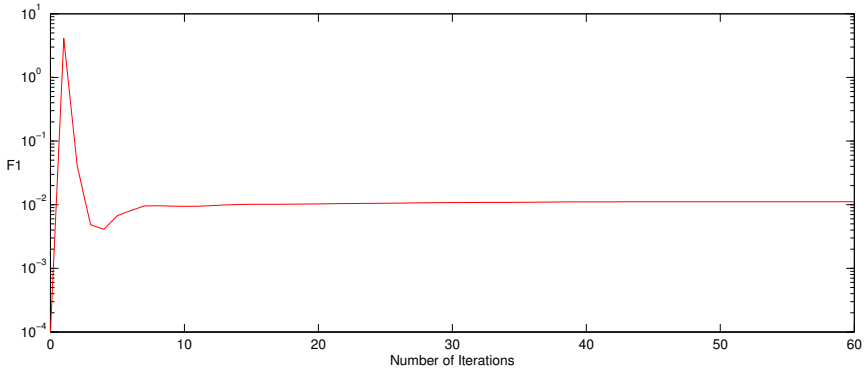
$$\nabla \times v = \Big(4(x_1 - 0.5)^2 + 4(x_2 - 0.5)^2 - 2\Big) \exp\Big(- 2((x_1 - 0.5)^2 + (x_2 - 0.5)^2)\Big).$$

In addition, even though it is divergence free, this field is not a solution of (5.10). We randomly select 100 vectors in (5.12) in order to simulate a vector field on a non-uniform grid. The resulting field is displayed in Figure 5.1. Figure 5.2 then displays the details of the iterations when the other parameters are $k_e = 0.1$, $\epsilon = 0.05$ and $N_b = 64$. We observe that for the first few iterations, both kinetic energy F_1 and error with prescribed field F_2 are decreasing. After that, the algorithm slightly increases F_1 while decreasing F_2. This behavior is fairly typical, especially for small k_e.

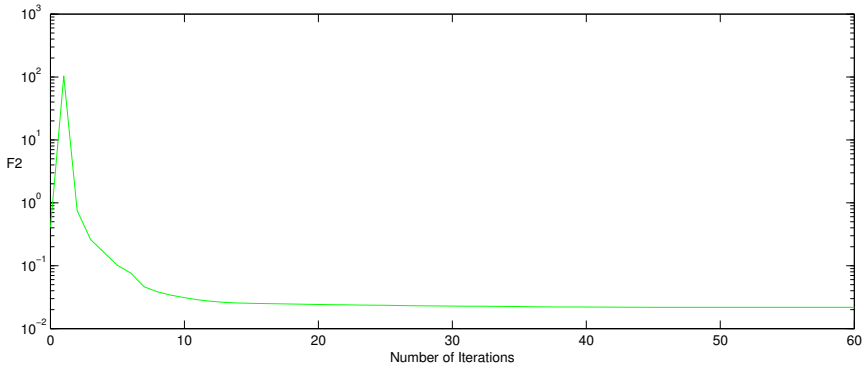
Let us now analyze the effect of the parameter k_e. Figure 5.3 and Table 5.1 present the results of the algorithm when ϵ and N_b are fixed to $\epsilon = 0.05$ and $N_b = 64$ but k_e is varied. We see that for small values of k_e, the velocity field obtained displays artificial features (at the bottom right of the target vortex) which disappear for $k_e = 10$. Indeed, the algorithm allows more kinetic energy for small k_e in order to decrease F_2 as much as possible. We also observe that it is not desirable to select k_e too large since the optimal solution becomes very close to 0, which corresponds to a field with



(a) Value of F as a function of the number of iterations.



(b) Value of F_1 as a function of the number of iterations.



(c) Value of F_2 as a function of the number of iterations.

Figure 5.2: Iterations details for Test 1 with $k_e = 0.1$, $\epsilon = 0.05$ and $N_b = 64$.

almost no kinetic energy. These claims are quantified with the norms given in Table 5.1. The first two rows of this table give the values of F_1 and F_2 obtained for different values of k_e. The last three rows give the error between the target field (5.12) and the field obtained by the algorithm at time $t^\star = \Delta t/2$ in weighted norm l^2_{norm}, relative l^2 norm e_{total} and l^∞ norm, which were defined in Section 4.1. As k_e varies from

ϵ	1.0	0.5	0.2	0.1	0.05	0.02
F_1	0.0119	0.00894	0.00998	0.00986	0.0111	0.0142
F_2	0.3450	0.2180	0.00898	0.00910	0.0218	0.0300
l^2_{norm} error	0.0668	0.0580	0.0196	0.0252	0.0392	0.0548
e_{total}	0.9024	0.6768	0.0776	0.1262	0.3101	0.6034
l^∞ error	0.2947	0.2593	0.1130	0.1050	0.2113	0.8124

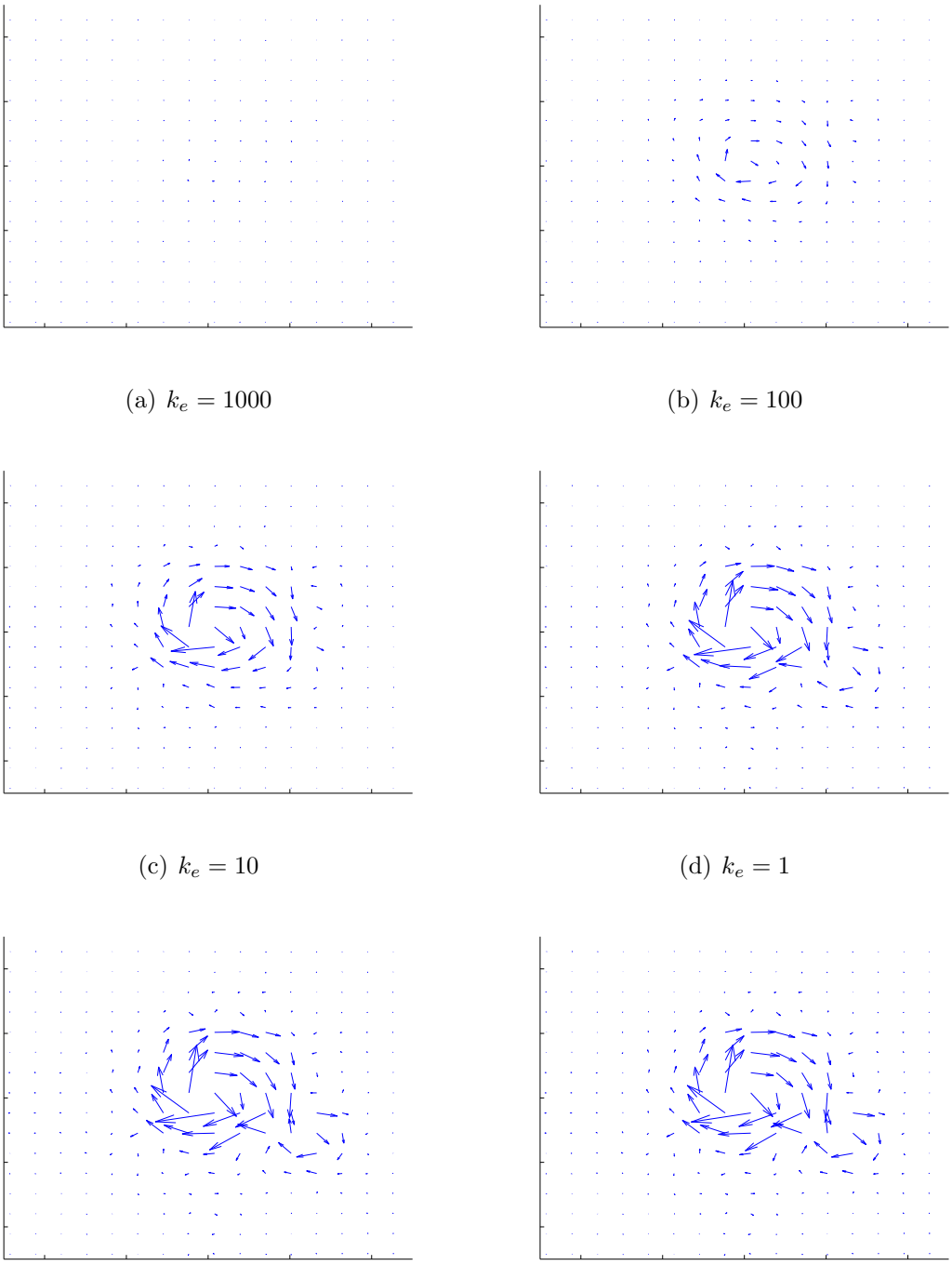
Table 5.2: Results for Test 1 when $k_e = 1$ and $N_b = 64$. The errors presented are computed for every grid point in Ω.

N_b	4	16	64	256
F_1	0.00733	0.00884	0.0107	0.0113
F_2	0.1820	0.0695	0.00869	0.000423
l^2_{norm} error	0.0470	0.0281	0.0260	0.0193
e_{total}	0.4438	0.1582	0.1363	0.0752
l^∞ error	0.1704	0.1015	0.1025	0.0999

Table 5.3: Results for Test 1 when $k_e = 0.1$ and $\epsilon = 0.1$. The errors presented are computed for every grid point in Ω.

1000 to 0, we see that the kinetic energy F_1 steadily increases while the error with the prescribed field F_2 steadily decreases. The l^2_{norm} error is best when $k_e = 10$, and significantly increases when k_e is increased or reduced. A similar behavior is displayed for the l^∞ error, which is best at $k_e = 1$. These results confirm the usefulness of k_e, not only to make the functional strictly convex, but also to prevent the formulation of artificial features for sparse grids. Note that some residual error is expected since v in (5.12) is not a solution of the Navier-Stokes equations (5.10). In addition, the solutions presented for $k_e = 0$ may not be unique, but initiating $\omega_0 = 0$ possibly helps creating a bias towards a field with a small kinetic energy.

Now that we analyzed the effect of k_e, let us fix $k_e = 1$ and $N_b = 64$ and vary ϵ. Figure 5.4 presents the vector fields obtained for different values of ϵ. The behavior observed is similar to the behavior of k_e: taking ϵ too small or too large decreases the quality of the approximation. Indeed, by taking ϵ to be too large, the algorithm cannot fully recover the size of the target vortex, and by taking ϵ too small, the algorithm allows smaller variations in the field which are not physical in this case. This observation from the images is confirmed in Table 5.2: the best field obtained is when $\epsilon = 0.2$ for the l^2_{norm} error and $\epsilon = 0.1$ for the l^∞ error. The relative l^2 norm e_{total} is worse for the biggest ϵ tested whereas the l^∞ norm is worse for the smallest ϵ tested. In addition, F_1 and F_2 also become larger as ϵ increases or decreases from its best values.



(a) $k_e = 1000$

(b) $k_e = 100$

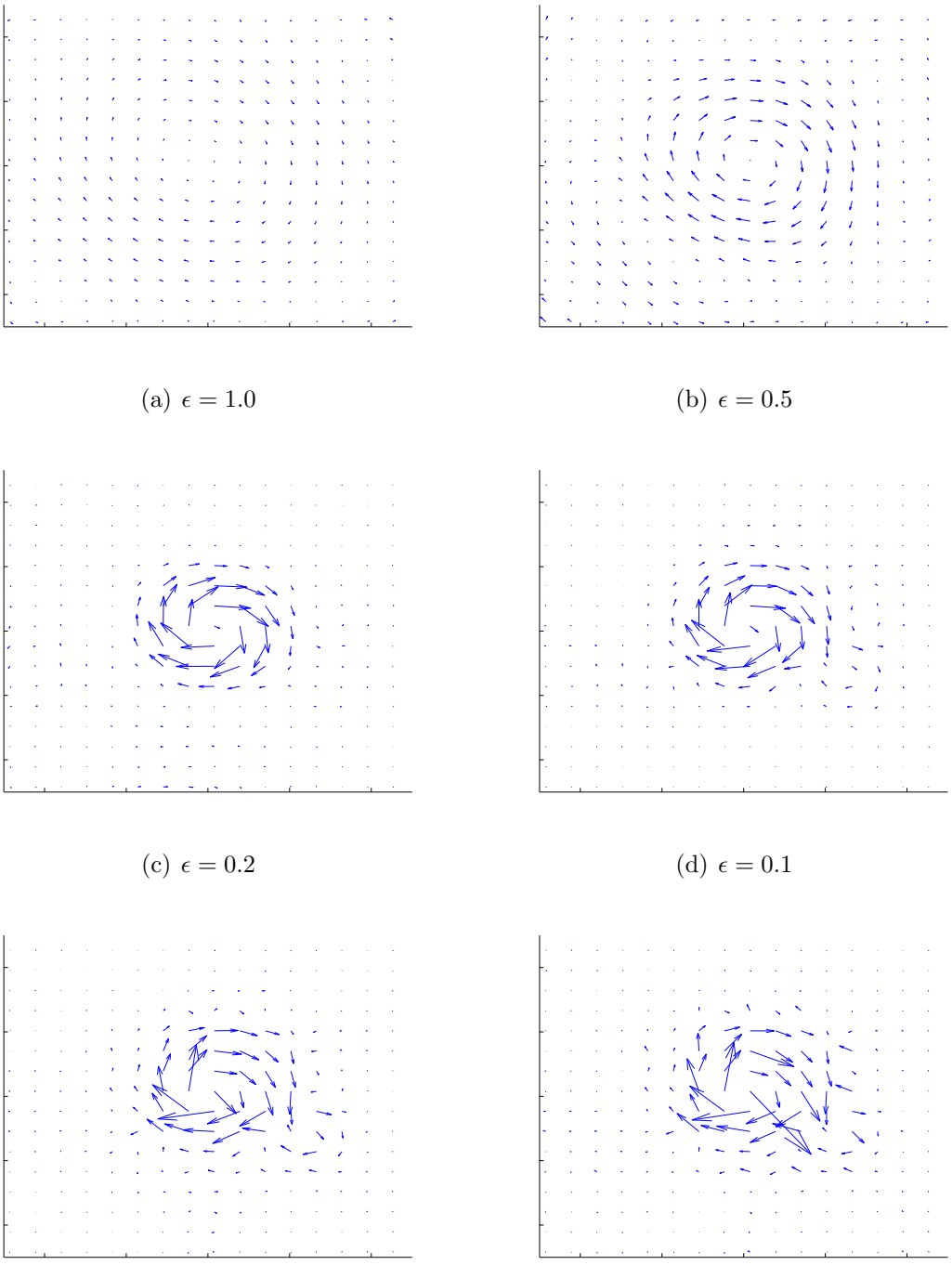
(c) $k_e = 10$

(d) $k_e = 1$

(e) $k_e = 0.1$

(f) $k_e = 0$

Figure 5.3: Results for Test 1 when $\epsilon = 0.05$ and $N_b = 64$.



(a) $\epsilon = 1.0$

(b) $\epsilon = 0.5$

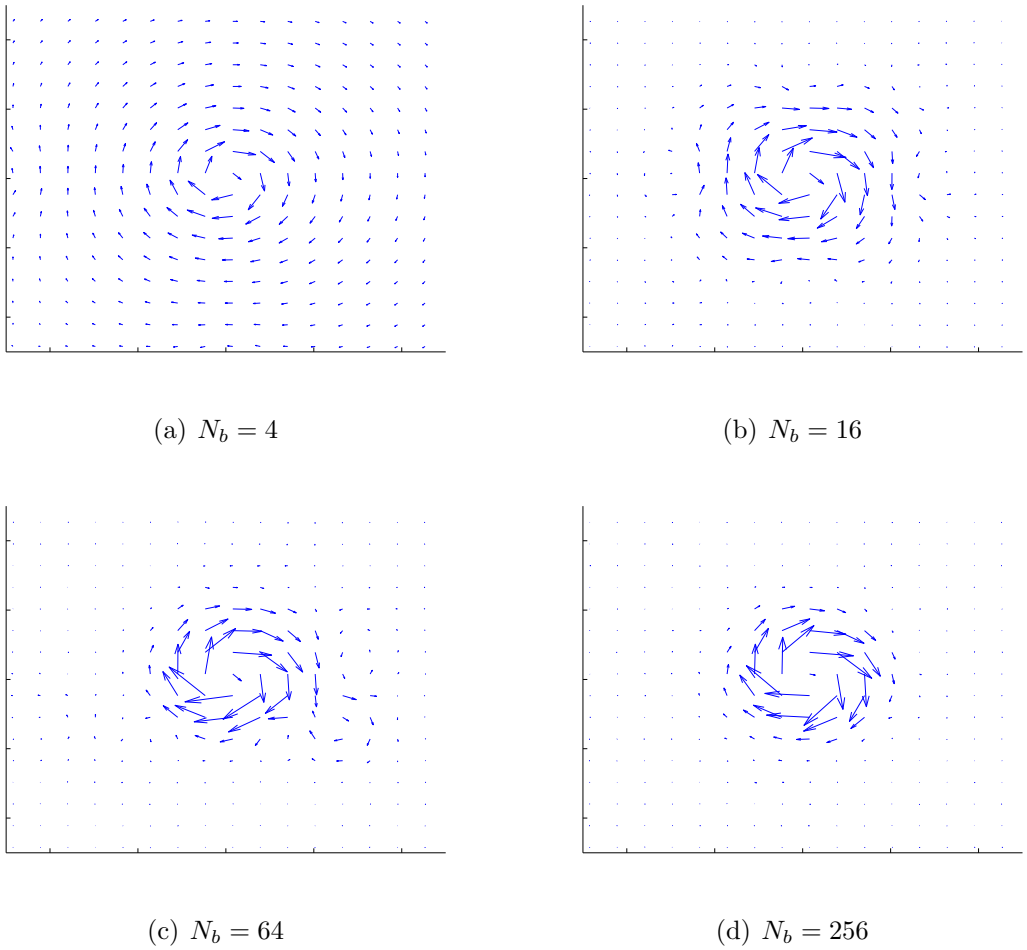
(c) $\epsilon = 0.2$

(d) $\epsilon = 0.1$

(e) $\epsilon = 0.05$

(f) $\epsilon = 0.02$

Figure 5.4: Results for Test 1 when $k_e = 1$ and $N_b = 64$.

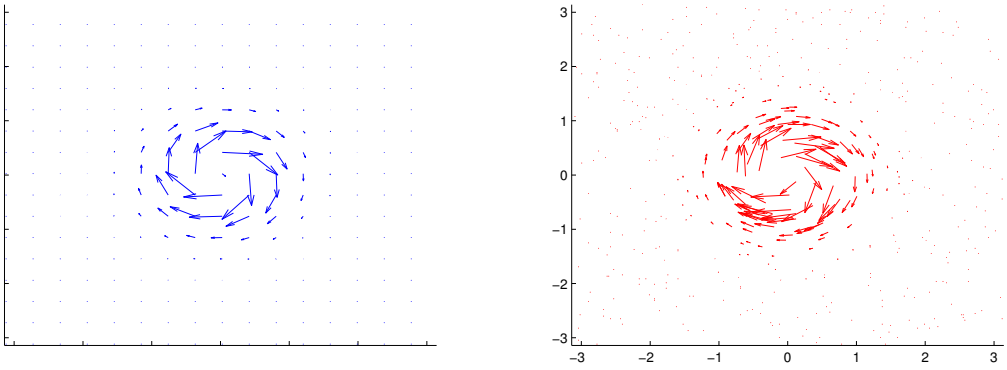


(a) $N_b = 4$ (b) $N_b = 16$

(c) $N_b = 64$ (d) $N_b = 256$

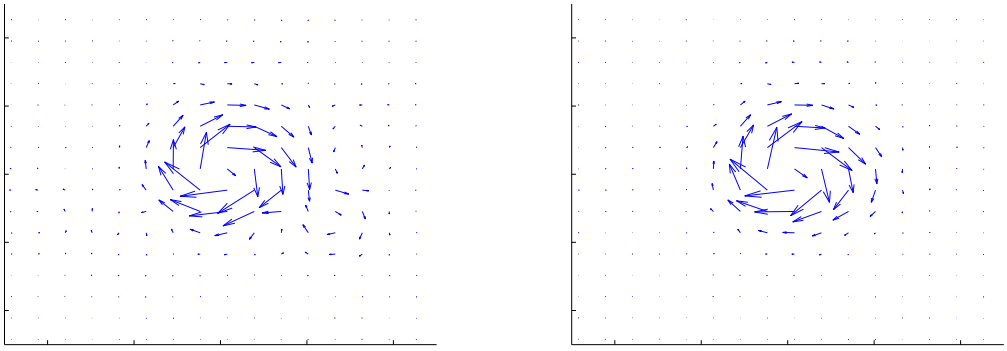
Figure 5.5: Results for Test 1 when $k_e = 0.1$ and $\epsilon = 0.1$.

Finally, we fix $k_e = 0.1$, $\epsilon = 0.1$ and vary N_b. The results for grids $N_b = 2 \times 2$, $N_b = 4 \times 4$, $N_b = 8 \times 8$ and $N_b = 16 \times 16$ are presented in Figure 5.5 and Table 5.3. We see on the images that the quality of the approximation increases as N_b gets larger, which is to be expected. The size of F_2 roughly decreases by a factor 10 for each increase in N_b. The results are best when $N_b = 256$, but this comes at the expense of a much longer computational time required to obtain the approximate field. Indeed, this time jumps from about 15 minutes for $N_b = 64$ to about 6.5 hours for $N_b = 256$ (still on the personal laptop of specifications described earlier).



(a) Target Field (b) Selection of 500 vectors

Figure 5.6: Fields for Test 2.



(a) Test 1 result. (b) Test 2 result.

Figure 5.7: Comparison of results for 100 and 500 randomly selected vectors (Tests 1 and 2), for the parameters $\epsilon = 0.1$, $k_e = 0.1$ and $N_b = 64$.

	100 vectors	500 vectors
F_1	0.0107	0.0113
F_2	0.00869	0.1071
l^2_{norm} error	0.0260	0.0173
e_{total}	0.1363	0.0604
l^∞ error	0.1025	0.0683

Table 5.4: Norm comparisons of results for 100 and 500 randomly selected vectors (Tests 1 and 2), for the parameters $\epsilon = 0.1$, $k_e = 0.1$ and $N_b = 64$. The errors presented are computed for every grid point in Ω.

Test 2

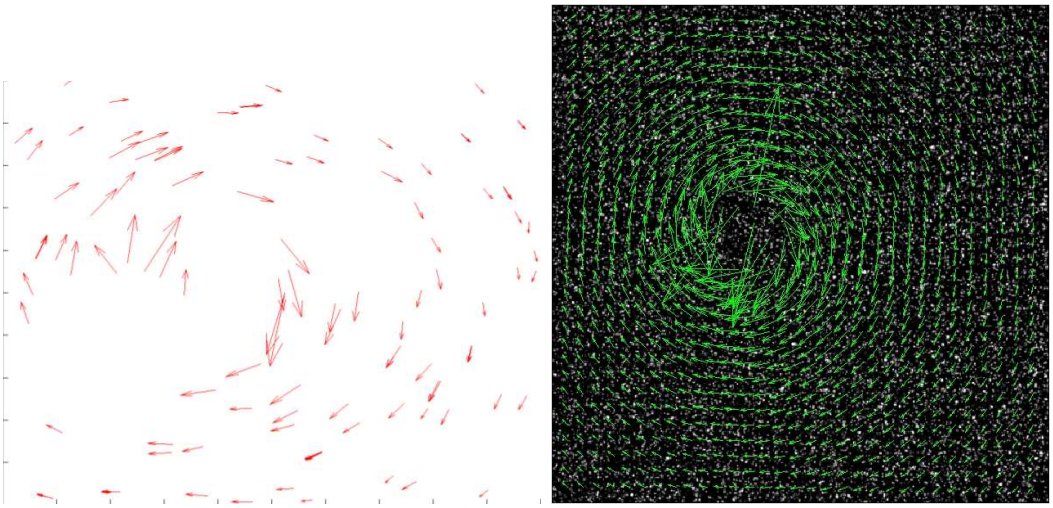
For our second numerical test, we keep the same vector field (5.12) but we now randomly select 500 vectors instead of 100. The resulting field is displayed in Figure 5.6. We then select the parameters $\epsilon = 0.1$, $k_e = 0.1$ and $N_b = 64$, and we compare the results obtained to the results of Test 1 for the same values of the parameters. Figure 5.7 and Table 5.4 give the details of this comparison. One obvious observation arising is that the l^2_{norm} and l^∞ norms are better when 500 vectors are given in the initial field. As expected, the more vectors given, the better the field obtained approximates the target. Starting from 500 vectors, the vector field obtained with the current algorithm is now only 6% away from the target field. Also, we see that F_2 increases from Test 1 to Test 2, but this simply has to do with the fact that the sum in F_2 has many more terms to it, combined with the fact that v is not a solution of the Navier-Stokes equations (5.10).

Test 3

For the third numerical experiment, we choose as a starting point one of the fields obtained with our OT-PIV algorithm on the data set given by the PIV Challenge 2001 B3 (see Test 5 in Section 4.7). Figure 5.8 displays the vector field given by our algorithm for selected values of the parameters. We only present here one example of results for one set of parameters because in Tests 1 and 2 we analyzed in details the effect of those parameters and we found that they were quite similar for this new test. The field presented in Figure 5.8 d) is consistent with what one would expect out of this example. We can also see that the magnitude of the vectors close to the vortex's center are better captured with the method presented in this chapter than for the linear interpolation method of Section 4.8. However, this comparison is now limited to qualitative statements as we do not have access to the real target velocity field.

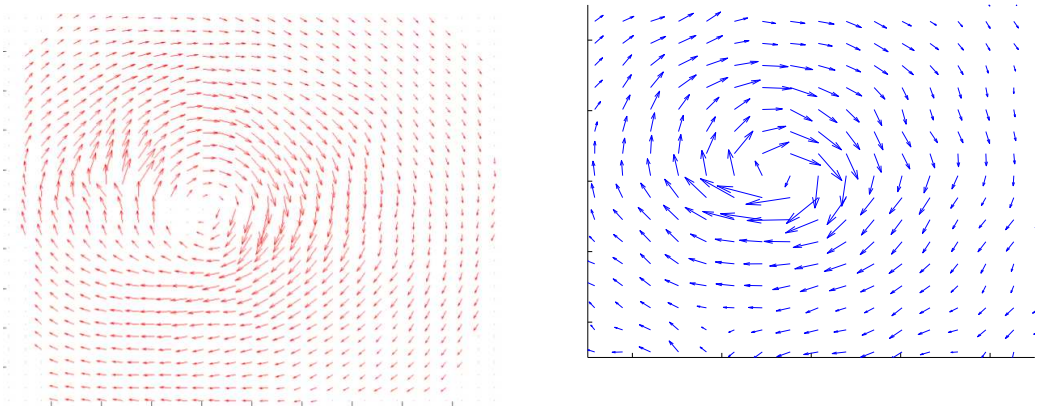
Test 4

In this final test, we apply the procedure to one of the scattered OT-PIV field obtained using the real PIV dataset of Test 7 in Section 4.8. The results are presented in Figure 5.9. This time, we display the fields associated with three different values of ϵ to show an interesting feature of the algorithm. For $\epsilon = 3$, we see that the general direction of the flow is recovered, but none of the finer structures are present. For $\epsilon = 0.8$, the flow recovered is fairly close to the one given in b) and c) and the finer variations are displayed. However, the flow given by $\epsilon = 0.3$ gives too small features



(a) Field obtained with OT method given in Figure 4.13 d).

(b) Result of PIVlab's DCC algorithm.

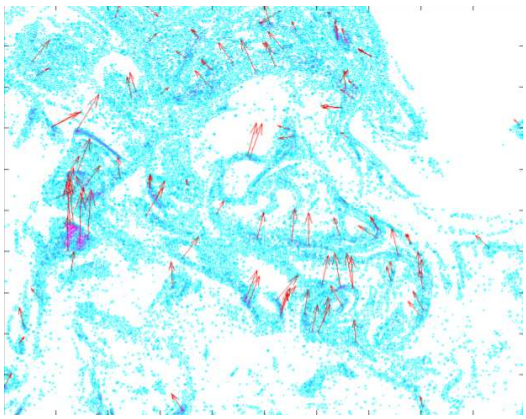


(c) Linear interpolation with the method of Section 4.8.

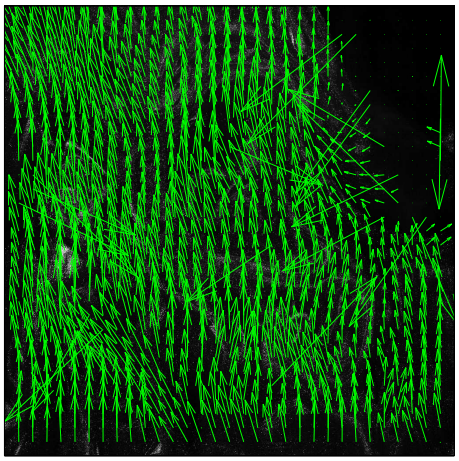
(d) Result of test 3.

Figure 5.8: Results of Test 3. The parameters selected for d) are $\epsilon = 0.2$, $k_e = 0.1$ and $N_b = 256$.

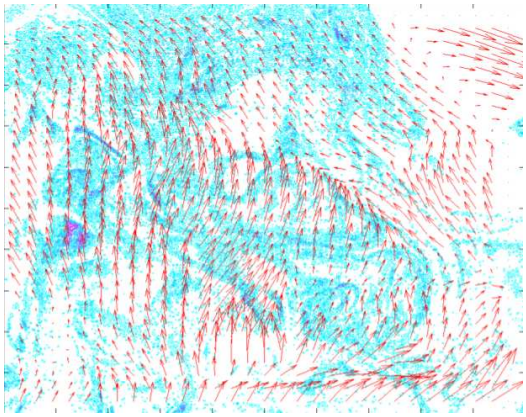
which do not seem to be accurate in the target flow. Keep in mind that we do not have access to the exact solution as this dataset is taken from a real experiment, but we can nonetheless observe that varying ϵ may help to visualize different regimes of the target velocity field. Also, even though a large value of N_b was selected for Test 3 and Test 4, good results can also be obtained for $N_b = 64$.



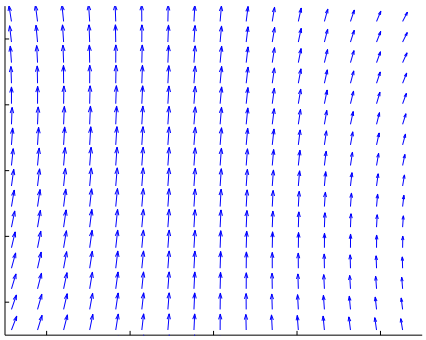
(a) Field obtained with OT method given in Figure 4.20 e)



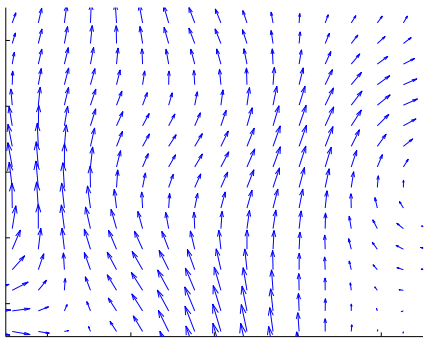
(b) Result of PIVlab's DCC algorithm.



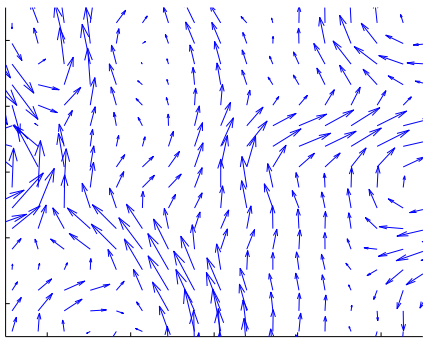
(c) Linear interpolation with the method of Section 4.8.



(d) Result of test 4 for $N_b = 256$, $k_e = 0.1$ and $\epsilon = 3$.



(e) Result of test 4 for $N_b = 256$, $k_e = 0.1$ and $\epsilon = 0.8$.



(f) Result of test 4 for $N_b = 256$, $k_e = 0.1$ and $\epsilon = 0.3$.

Figure 5.9: Results of Test 4.

Chapter 6

Conclusions

We presented in this work a new PIV method based on the theory of optimal mass transportation to retrieve the flow velocity of a fluid. While traditional PIV algorithms are typically based on cross-correlations between the successive images recorded [23, 36], our results indicate that when augmented with proper post-processing filters, the OT-PIV method yields equally valid results in many cases, and even better in some. We also effectively extended the velocity fields obtained with this method to physically relevant fields solving the Navier-Stokes equations. The OT technique presented has several advantages: it can be used on problems requiring any number of physical dimensions; it is a global approach and thus there is no need to split the domain into several interrogation windows; it can be applied to virtually any distribution of tracers and it provides velocity vectors for individual particles and not just for groups of several particles. While these advantages are significant, the method also comes with some drawbacks. However, we have effectively shown in this work how to correct these drawbacks in order to improve the results obtained.

Indeed, with the presence of background noise in the image, we found from the theory and the numerical tests that the optimal map does not send the particle's initial location exactly to its target location. To correct this behavior, we designed the Particle Target Position Correction (PTPC) filter which reassigns the predicted final position to the brightest neighboring pixel, within a radius of c pixels. The parameter c depends on the resolution of the images at hand (the number of pixels per particle), and in order to avoid creating too many mismatches, it should be selected so that the maximum correction corresponds to the radius of a particle, in pixels. One future improvement on this filter would be to combine it with a subpixel approximation procedure typical to cross-correlation algorithms [36] so that the precision of the

recovered position goes beyond a pixel's size.

Because the light distribution around a particle is closer to a Gaussian distribution than it is to a Dirac-delta function, the mass associated with a single particle can be split or spread around two or more particle locations. We thus introduced the No Split filter which consists in discarding particles whose masses are spread by the transport map over a region exceeding a certain limit determined by the parameter k_{ns}. This improves the flow in regions of high densities or rapid rotations prone to mismatching between particles by the OT map. Concerning the value of the parameter k_{ns}, we saw that it mostly depends on the seeding density and the viscosity of the fluid. For lower densities within the tracking range, a lower value of k_{ns} (close to 1.5 for example) can be selected as we do not expect particles to be influencing each other too much, unless there is a clear mismatch. For higher densities, we saw from numerical tests that taking k_{ns} too low removes too many vectors in regions of greater turbulence, and thus a value closer to 3 or 4 is desirable. The viscosity of the fluid can also play a role in determining the size of k_{ns}. For example, smaller viscosities may lead to more turbulent fluids and require a higher k_{ns} to obtain a reasonable number of vectors (the opposite statement should also hold true for higher viscosities).

By analyzing both theory and experiments, we showed in addition that, in the case of uneven particle brightness, the OT map performs better on the brighter particles than it does on the faded ones. Using only the locations of particles whose brightness in the initial image exceeds a certain threshold β to construct the flow velocity drastically improves the accuracy of the later. For images at lower seeding densities, a lower β can be safely employed. This is the case when the particles mostly remain within the tracking range, which we can validate by verifying whether the average displacement is smaller than the average distance between particles. It could also be possible to employ a nonuniform β which could be taken as a threshold on the relative brightness of a particle with respect to its immediate neighbors. However, even when particles are quite isolated from each other, if the noise level in the pictures is high when compared to the brightness of a particle, the parameter β cannot be taken too low as there might be significant mass exchanges between noise and particles (β should be at least greater than a multiple of $1 - r$). The parameter β essentially provides a trade-off between accuracy and quantity of information, and more involved strategies involving this parameter should be investigated in the future.

Given the continuum nature of the transport flow, one is also tempted to extend this approximation outside the particle locations. However, as we pointed out in

this work, the flow field associated with the OT problem solves pressureless and inviscid transport equations which makes it far from physically relevant flows solving the Navier-Stokes equations. We therefore introduced an optimization problem to find the vector field that is the closest to the OT field and that solves the Navier-Stokes equations while minimizing its total kinetic energy. To do so, we assumed that initial vorticity in the OT field was given by a sum of Gaussian blobs of fixed width determined by a parameter ϵ. The minimal kinetic energy constraint was enforced as a penalty term on the error with the prescribed field, with corresponding penalty parameter k_e. The interplay of the parameters k_e and ϵ also contributed to filtering the flow from erroneous approximated vectors. We found that larger values of ϵ effectively smoothed the OT field by limiting the presence of small-scale features, while larger values of k_e made the problem strictly convex and helped reducing the presence of artificial features for sparse grids when smaller values of ϵ are employed. As for the parameter k_{ns}, one might be able to determine the size of k_e a priori from the seeding density and the fluid's viscosity, with a higher k_e possibly required for lower viscosities and lower densities. We also point out that this method could in fact be applied to any discrete vector field, not just ones given by our Optimal Transport procedure. For example, it could be used jointly with cross-correlation algorithms which give vector fields defined on a grid of interrogation windows.

Due to the mass preservation constraint inherent to the Optimal Transport problem, the OT-PIV method in its current stage is limited in 2D-PIV to flows that are not too turbulent, so that there is not too much noise coming from out-of-plane loss of particles. We have however demonstrated that this drawback can be alleviated by applying a low-pass filter before the OT algorithm. To extend the OT-PIV method to highly turbulent flows where high seeding densities are necessary, multi-pass algorithms similar to the ones employed in traditional cross-correlation methods could be employed. We quickly investigated this possibility in the current work, but this should be further explored. In addition, this would not be an issue anymore if the method is applied to 3D-PIV datasets, which are now more common thanks to modern technical advances. We believe it would definitely be worth implementing the different algorithms designed in this work in dimension 3 in the future in order to test the method on these datasets. In the longer run, the very general framework provided by the Optimal Transport could provide even more accurate methods. The theory of Optimal Transportation is rapidly evolving and future theoretical results or more advanced numerical algorithms might be directly applicable to the PIV problem.

Bibliography

[1] R.J. Adrian. Twenty years of particle image velocimetry. *Experiments in Fluids*, 39:159–169, 2005.

[2] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84:375–393, 2000.

[3] J.-D. Benamou, B. Froese, and A. Oberman. Numerical Solution of the Optimal Transportation Problem Using the Monge-Ampère Equation. *Journal of Computational Physics*, 260:107–126, 2014.

[4] D. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals Operations Research*, 14(1):105–123, 1988.

[5] Y. Brenier. Décomposition polaire et réarrangement monotone de champs de vecteurs. *Comptes Rendus de l'Académie des Sciences de Paris Ser. I*, 305(19):805–808, 1987.

[6] M. Burger, M. Franek, and C.-B. Schonlieb. Regularized regression and density estimation based on optimal transport. *Applied Mathematics Research eX*, 2:209–253, 2012.

[7] L.A. Caffarelli. The regularity of mappings with a convex potential. *Journal of the American Mathematical Society*, pages 99–104, 1992.

[8] L.A. Caffarelli. Boundary regularity of maps with convex potentials–ii. *Annals of mathematics*, pages 453–496, 1996.

[9] PIV Challenge. `http://www.pivchallenge.org`.

[10] R. Chartrand, B. Wholberg, K. Vixie, and E. Bollt. A gradient descent solution to the Monge-Kantorovich problem. *Applied Mathematical Sciences*, 3(21–24):1071–1080, 2009.

[11] D. Cordero-Erausquin. Sur le transport de mesures périodiques. *Comptes Rendus de l'Académie des Sciences de Paris Ser. I*, 329:199–202, 1999.

[12] J.E.Jr. Dennis and J.J. Moré. Quasi-newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.

[13] G.E. Elsinga, F. Scarano, B. Wieneke, and B.W. van Oudheusden. Tomographic particle image velocimetry. *Experiments in Fluids*, 41:933–947, 2006.

[14] L.C. Evans. *Partial Differential Equations*. American Mathematical Society, 1998.

[15] S. Ferradans, N. Papadakis, J. Rabin, G. Peyr, and J.-F. Aujol. *Regularized Discrete Optimal Transport*. Springer Berlin Heidelberg, 2013.

[16] D. Goldfarb. Efficient dual simplex algorithms for the assignment problem. *Mathematical Programming*, 33(2):187–203, 1985.

[17] E. Haber, T. Rehman, and A. Tannenbaum. An efficient numerical method for the solution of the l^2 optimal mass transfer problem. *SIAM Journal on Scientific Computing*, 32(1):197–211, 2010.

[18] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent. Optimal mass transport for registration and warping. *International Journal of Computer Vision*, 60(3):225–240, 2004.

[19] Douglas P Hart. The elimination of correlation errors in piv processing. In *9th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, pages 13–16, 1998.

[20] D.P. Hart. Super-resolution piv by recursive local-correlation. *Journal of Visualization*, 3(2):187–194, 2000.

[21] D. Heitz, E. Mémin, and C. Schnörr. Variational fluid flow measurements from image sequences: synopsis and perspectives. *Experiments in fluids*, 48(3):369–393, 2010.

[22] E. Kalnay. *Atmospheric modeling, data assimilation, and predictability.* Cambridge university press, 2003.

[23] R.D. Keane and R.J. Adrian. Theory of cross-correlation analysis of PIV images. *Applied Scientific Research*, 49:191–215, 1992.

[24] M. Kohler and A. Krzyzak. Nonparametric estimation of non-stationary velocity fields from 3D particle tracking velocimetry data. *Computational Statistics and Data Analysis*, 56(6):1566–1580, 2012.

[25] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[26] A. Leonard. Vortex methods for flow simulation. *Journal of Computational Physics*, 37(3):289–335, 1980.

[27] W.L. Lim, Y.T. Chew, T.C. Chew, and H.T. Low. Steady flow dynamics of prosthetic aortic heart valves: a comparative evaluation with piv techniques. *Journal of biomechanics*, 31(5):411–421, 1998.

[28] G. Loeper and F. Rapetti. Numerical solution of the Monge-Ampère equation by a Newton's algorithm. *Comptes Rendus de l'Académie des Sciences de Paris*, I(340):319–324, 2005.

[29] A. J. Majda and A. L. Bertozzi. *Vorticity and incompressible flow*, volume 27. Cambridge University Press, 2002.

[30] R.J. McCann. A Convexity Principle for Interacting Gases. *Adv. Math.*, 128:153–179, 1997.

[31] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, pages 666–704, 1781.

[32] O. Museyko, M. Stiglmayr, K. Klamroth, and G. Leugering. On the application of the MongeKantorovich problem to image registration. *SIAM J. Imaging Sci.*, 2:1068–1097, 2009.

[33] J. Nogueira, A. Lecuona, and P.A. Rodriguez. Limits on the resolution of correlation piv iterative methods. fundamentals. *Experiments in fluids*, 39(2):305–313, 2005.

[34] N. Papadakis and É. Mémin. Variational assimilation of fluid motion from image sequence. *SIAM Journal on Imaging Sciences*, 1(4):343–363, 2008.

[35] J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. *Scale Space and Variational Methods in Computer Vision*, pages 435–446, 2012.

[36] M. Raffel, C.E. Willert, S.T. Wereley, and J. Kompenhans. *Particle Image Velocimetry: A Practical Guide*. Springer Berlin Heidelberg, 2007.

[37] T. Rehman, E. Haber, G. Pryor, J. Melonakos, and A. Tannenbaum. 3d nonrigid registration via optimal mass transport on the gpu. *Medical Image Analysis*, 13(6):931–940, 2009.

[38] P. Ruhnau, A. Stahl, and C. Schnörr. Variational estimation of experimental fluid flows with physics-based spatio-temporal regularization. *Measurement Science and Technology*, 18(3):755, 2007.

[39] L.-P. Saumier, M. Agueh, and B. Khouider. An Efficient Numerical Algorithm for the L^2 Optimal Transport Problem with Periodic Densities. *IMA Journal of Applied Mathematics*, 80(1):135–157, 2015.

[40] L.-P. Saumier, B. Khouider, and M. Agueh. Optimal transport for particle image velocimetry. *Communications in Mathematical Sciences*, 13(1):269–296, 2015.

[41] L.-P. Saumier, B. Khouider, and M. Agueh. Optimal transport for particle image velocimetry: Real Data and Post-Processing Algorithms. *SIAM Journal on Applied Mathematics*, 75(6):2495–2514, 2015.

[42] L.P. Saumier. An Efficient Numerical Algorithm for the L^2 Optimal Transport Problem with Applications to Image Processing. M.sc. thesis, University of Victoria, 2010.

[43] W.E. Schaap and R. van de Weygaert. Continuous fields and discrete samples: reconstruction through delaunay tessellations. *Astronomy & astrophysics*, 363(3):L29–L32, 2000.

[44] G.R. Spedding and E.J.M. Rignot. Performance analysis and application of grid interpolation techniques for fluid flows. *Experiments in Fluids*, 15(6):417–430, 1993.

[45] M. Stanislas, K. Okamoto, and C.J. Kahler. Main results of the first international PIV challenge. *Measurement Science and Technology.*, 14:R63–R89, 2003.

[46] M. Stanislas, K. Okamoto, C.J. Kähler, and J. Westerweel. Main results of the second international piv challenge. *Experiments in Fluids*, 39(2):170–191, 2005.

[47] M. Stanislas, K. Okamoto, C.J. Kähler, J. Westerweel, and F. Scarano. Main results of the third international piv challenge. *Experiments in Fluids*, 45(1):27–71, 2008.

[48] J. Strain. Fast Spectrally-Accurate Solution of Variable-Coefficients Elliptic Problems. *Proceedings of the AMS*, 122(3):843–850, 1994.

[49] T. Suzuki. Reduced-order kalman-filtered hybrid simulation combining particle tracking velocimetry and direct numerical simulation. *Journal of Fluid Mechanics*, 709:249–288, 2012.

[50] R. Temam. *Navier-Stokes equations and nonlinear functional analysis*, volume 66. 1995.

[51] W. Thielicke. *The flapping flight of birds - analysis and application.* Ph.d. dissertation, Rijksuniversiteit Groningen, 2014.

[52] W. Thielicke and E.J. Stamhuis. Pivlab - time-resolved digital particle image velocimetry tool for matlab (version: 7.10). `http://dx.doi.org/10.6084/m9.figshare.1092508`, 2014.

[53] E.D. Tytell. Do trout swim better than eels? challenges for estimating performance based on the wake of self-propelled bodies. In *Animal Locomotion*, pages 63–74. Springer, 2010.

[54] C. Villani. *Topics in Optimal Transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, 2003.

[55] C. Villani. *Optimal Transport:Old and New.* Springer-Verlag, 2009.

[56] A. Vlasenko and C. Schnörr. Physically consistent and efficient variational denoising of image fluid flow estimates. *Image Processing, IEEE Transactions on*, 19(3):586–595, 2010.

[57] J. Westerweel. *Digital particle image velocimetry Theory and application*. Ph.d. dissertation, Delft University, 1993.

[58] J. Westerweel, D. Dabiri, and M. Gharib. The effect of a discrete window offset on the accuracy of cross-correlation analysis of digital piv recordings. *Experiments in fluids*, 23(1):20–28, 1997.

[59] D.J. White, W.A. Take, and M.D. Bolton. Soil deformation measurement using particle image velocimetry (piv) and photogrammetry. *Geotechnique*, 53(7):619–631, 2003.

[60] B. Wieneke LAVISION. Sequence of experimental time-resolved PIV images showing a slightly turbulent air flow with small water droplets with a diameter of about 5 micrometer. `http://fluid.irisa.fr/data-eng.htm`.

[61] C. Willert, M. Raffel, J. Kompenhans, B. Stasicki, and C. Kähler. Recent applications of particle image velocimetry in aerodynamic research. *Flow Measurement and Instrumentation*, 7(3):247–256, 1996.

[62] Y. Zemel. Optimal Transportation: Continuous and Discrete. M.sc. thesis, EPFL, 2012.