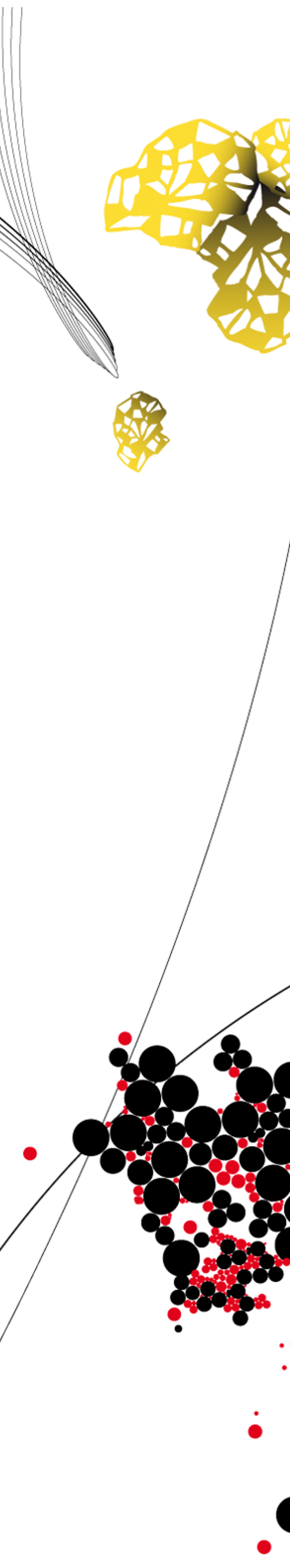


UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Combined Modelling of Optimal Transport and Segmentation Revealing Vascular Properties

Yoeri E. Boink

MSc Thesis
September 2016

Assessment committee:
prof. dr. Stephan van Gils
prof. dr. Hans Zwart
dr. Christoph Brune

Supervisor:
dr. Christoph Brune

Applied Analysis Research Group
Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

For studying vascular structures in 4D biomedical imaging, it is of great importance to automatically determine the velocity of flow in video sequences, for example blood flow in vessel networks. In this thesis new optimal transport models focusing on direction and segmentation are investigated to find an accurate displacement between two density distributions. By incorporating fluid dynamics constraints, one can obtain a realistic description of the displacement. With an a-priori given segmentation of the network structure, transport models can be improved. However, a segmentation is not always known beforehand. Therefore, in this work a joint segmentation-optimal transport model has been described. Other contributions are the ability of the model to allow for inflow or outflow and the incorporation of anisotropy in the displacement cost. For the problem, a convex variational method has been used and primal-dual proximal splitting algorithms have been implemented. Existence of a solution of the model has been proved. The framework has been applied to synthetic vascular structures and real data, obtained from a collaboration with the applied mathematics and the hospital in Cambridge.

Keywords: optimal transport, segmentation, anisotropy, fluid dynamics, variational methods, convex optimisation, vascularity.

Acknowledgements

This thesis is the result of over a half year of research in the group of Applied Analysis at the University of Twente. I have really enjoyed my time working on this project, which is due to many people, whom I cannot mention all. I want to thank some of them specifically.

First of all a big thanks for dr. Christoph Brune. Christoph inspired me to do both my internship and my final project in the very interesting field of mathematical imaging. For my final project, we have had many interesting, inspiring and sometimes long discussions. The resulting work would not have been possible without his great knowledge and the enormous amount of time he decided to spend on supervising me. I am happy to say that he will continue to be my supervisor during my PhD-research.

Next I would like to thank dr. Carola-Bibiane Schönlieb, who has been my supervisor during my stay at the University of Cambridge. Not only was it great to be a visiting member of her group, she also continued to help me during my final project and showed interest in my progress.

I would also like to acknowledge prof. dr. Stephan van Gils for his valuable comments during my research and for reading my thesis carefully in the past weeks.

I thank Ruben Heersink and Nick Luiken for being able to listen to my sometimes incomprehensible questions, giving me good feedback and drinking litres of coffee together.

Finally, I am very grateful to my parents, my family and my friends for supporting me, showing interest and giving me a great time by distracting me from my studies when that had to be done. Especially I want to thank Evelien for always listening to me and being there for me.

Contents

1 Introduction

With the use of video imaging techniques, the movement or flow of certain objects can be captured. In the field of biomedical imaging, it is of great importance to automatically determine the velocity of flow in video sequences, for example blood flow in vessel networks. Optimal transport models are capable of doing this job: a velocity estimation is obtained while finding the interpolation between two given images, that can be seen as two density functions. One can think of an example where one wants to transport a heap of sand of a certain shape to another heap of sand with a different shape, that might also be on a different location (Figure 1).

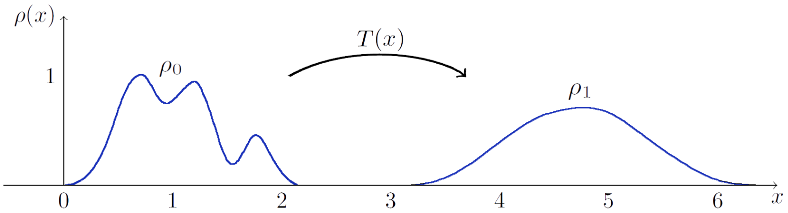


Figure 1: With optimal transport, one can find the map T that maps ρ_0 to ρ_1.

Originally, finding the optimal transport is a problem in measure theory, where one tries to find a transport map T mapping ρ_0 to ρ_1. This transport map minimises a certain cost functional for the transport. When an L^2-Wasserstein distance is chosen as cost functional, the problem to solve reads

$$d_p(\rho_0, \rho_1)^p = \inf_{T \in \mathcal{T}} \int_\Omega \|T(x) - x\|^p \, \rho_0(x) \, \mathrm{d}x, \tag{1.1}$$

where $\mathcal{T}$ is the set of all possible transport maps. In their famous paper, Benamou and Brenier [6] showed that for the L^2-Wasserstein distance, the problem can be set in a fluid dynamics framework. In this new formulation, solving the optimal transport problem is equivalent to solving a constrained convex minimisation problem in a continuous framework. Benamou and Brenier implemented the transport for both periodic and zero Dirichlet boundary conditions, where mass conservation is ensured. Papadakis *et al.* [36] took this fluid dynamics formulation and implemented it in several proximal splitting algorithms. They also implemented an optimal transport model for an H^{-1}-Wasserstein distance. In a follow-up paper, Hug *et al.* [21] extended the L^2-Wasserstein model with an anisotropic cost functional, therefore being able to model transport with a directional preference. Moreover, they incorporated several fluid dynamics constraints, to ensure incompressible, translational or rigid transports. Maas *et al.* [27, 28] changed the mass-conservation set to a set in which mass could be created or destroyed by taking singular sources and sinks into account.

In this thesis new optimal transport models for mathematical imaging are investigated. The goal is to acquire an interpolation and velocity estimation of flow in vascular structures in the field of biomedical imaging. We follow the approach of the fluid dynamics formulation to model optimal transport. Contributions are the use of edge-detecting tensors and structure tensors [33, 44] to obtain an anisotropy that is implemented in our models. We apply anisotropic regularisers for the transport, as well as various regularisers based on fluid dynamics constraints. Furthermore, we show several ways to incorporate segmentations in the models. In contrast to other models, our models allow for inflow or outflow, meaning that zero Dirichlet boundary conditions are not necessarily needed. Our last contribution is in analysis, where we prove the existence of a minimiser for a non-convex optimal transport model with flow regularisation.

In the field of mathematical imaging, it is becoming more and more popular to design so-called joint models. In these models, two imaging tasks are modelled in a joint fashion, to fulfil both tasks with the same model, instead of applying two models consecutively. Le Guyader and Vese [26] and Ozeré *et al.* [34, 35] developed several joint segmentation-registration models. In these models, a Chan-Vese type segmentation model [14] is combined with a non-linear (hyper)elasticity smoother. If the segmentation of the template image is available, both registration between template and reference image and segmentation of the reference image is performed together, using the same minimisation functional. In [19], Heldmann *et al.* developed a joint segmentation-registration model that allows for sliding motion at the interface of the two segments, based on a piecewise smooth deformation model. Droske and Rumpf [16] proposed a joint segmentation-registration model in which the two images to be registered have different modalities. Their model is based on morphological matching with elasticity regularisation. A joint multigrid segmentation-optical flow model has been derived by Mémin and Pérez [29], where the function to be minimised contains the the optical flow constraint and a discontinuity-preserving smoothness constraint. Preusser *et al.* [38] proposed to minimise a Mumford-Shah type energy functional in which not only a piecewise smooth image is searched for, but also the optical flow constraint is incorporated. This way, one searches for a smooth velocity field while denoising and segmenting the original images.

To our knowledge, a joint segmentation-optimal transport model has not been developed yet. Segmentation-optical flow models can only handle image sequences showing slow or linear flow. If a joint segmentation-optimal transport model would be available, fast and non-linear flows can be captured well. In this work, we give a first description of such a joint model. In a first implementation, the non-convex minimisation problem is solved by minimising three convex minimisation problems alternatingly.

This thesis is organised as follows. We start with a description of vascular structures and some motivational examples hereof in chapter 2. In chapter 3 the concept of optimal transport is explained and our two main optimal transport models are presented. We explain how to incorporate anisotropy and fluid dynamics constraints into the models. Moreover, we prove the existence of a minimiser for the non-convex minimisation problem. Chapter 4 starts with an overview of well known segmentation methods, after which we explain how to incorporate a segmentation in our earlier derived models. In chapter 5, a joint segmentation-optimal transport model is described. Then we explain how proximal splitting methods work in chapter 6, followed by the implementation of our two models in the modified Primal Dual Hybrid Gradient method. This includes the derivation of all operators and the choice of our numerical discretisation. In chapter 7, we show the results of applying our models to both synthetic and real data sets, followed by a discussion on these results. We conclude by a summary and outlook on further research in chapter 8.

2 Motivation

In this thesis, we will derive and implement a model for the interpolation and velocity estimation of flow in vascular structures. In order to do this properly, we need to know what kind of images we use as input data. In this chapter, we will first examine the class of images on which we will apply the model later on. Next we will give three concrete examples of realistic test cases that we will use in chapter 7.

In the sequel, we assume that we have two images in which an object or structure has undergone a movement between the first and the second image. Think of a road, where the cars have moved in the time between the two images were taken (Figure 2). We will always look at images in which there is an obvious movement or transport between the first and the second image; this movement may be either small or large.

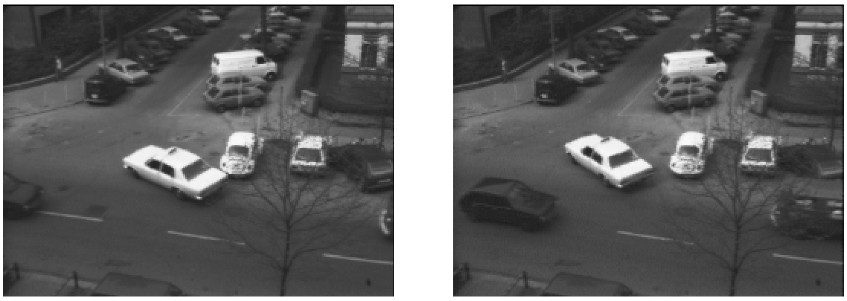


Figure 2: Two frames in the taxi-cab sequence (KOGS/IAKS Universität Karlsruhe).

2.1 Vascular structures

In biology, vascular structures are very common. Very often, these structures are meant to transport something in an organism from one place to the other. Think of blood flowing through blood vessels and arteries, lymph moving through the lymphatic system or nutrients moving through branches of a tree. If sequential images (a video) of this movement are available, we can see the movement between two images. We separate different classes of image sequences that can be found:

1) Videos in which the moving part can be seen as coherent objects. Examples are blood cells in a blood vessel or cars on a highway.
    a) Only one object is moving.
    b) Multiple objects are moving at the same time:
        i) with the same speed and direction;
        ii) with different speed and directions.
2) Videos in which the moving part is a dense collection of small particles.
    a) The structure of the collection of particles stays intact. An example is when we look at a traffic jam from far up above. Since the cars cannot overtake, the structure of cars and trucks stays the same over time.
    b) The structure changes continuously while moving. An example is a river which has a different surface every time, because of the waves that influence each other.

As we can see, many types of image sequences containing vascular structures are available. We will not treat all of them, but it is good to know there is a wide variety of image sequences that have to be dealt with differently.

In general, we have a non-moving background, which often has a different average intensity than the moving object or structure. However, it is not always clear that we have a single foreground structure. For instance, think of the example of the car on a highway: the background consist of the landscape in which the highway is laid. The highway can be seen as the 'vascular structure' on which the car is moving. It is not clear if the highway also belongs to the foreground or if only the car belongs to the foreground. We have to keep this in mind, when we want to apply a segmentation to our problem in chapter 4: we have to decide what has to be segmented from what.

2.2 Application to real data

In chapter 7, we will apply our model to a number of real data sets, to see how it behaves for different kind of images. In the following, we will define these data sets, which solely consist of two-dimensional images, although the model that we will develop can also be applied to three-dimensional images.

2.2.1 Moving white blood cell

The first real data set contains an image sequence in which a white blood cell (leukocyte) is moving along a c-shaped path. The data was obtained in a project of the University of Münster. In this project, the movement of leukocytes from blood vessels to inflamed tissue is investigated. Because of a barrier of endothelial cell layers, the leukocytes can not move freely and therefore sometimes follow a curved path.

In Figure 3 we see the movement of one leukocyte, where the aforementioned barriers are not shown. If we assume that the preferred path is given by the c-shape, it is interesting to see if we can recreate this image sequence, when only taking the first and last frame into account. Moreover, we could interpolate such that we know the position of the cell at every time t during the movement. When classifying the data set to one of the classes proposed in section 2.1, we see that it belongs to class 1a: we are dealing with a single object that moves, without the vascular structure being visible. The background shows almost zero intensity, which will prove to be very handy in applying the model.

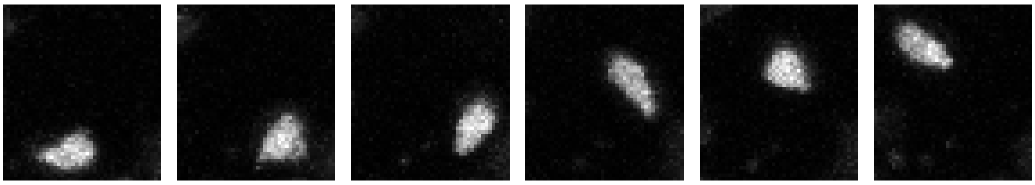


Figure 3: A sequence of six frames in which a cell is moving.

2.2.2 Haemoglobin microcirculation

The second real data set contains an image sequence in which the microcirculation of blood in the human eye is shown. This data set was obtained by collaboration with the applied mathematics and the hospital in Cambridge. In the image sequence, various flow types are visible: we see arteries pumping blood towards the eye and several types of blood vessels transporting the blood through the eye. For a more extensive description of this data set, we refer to [8].

An example of a close-up of a single image in the sequence is given in Figure 4. We are dealing

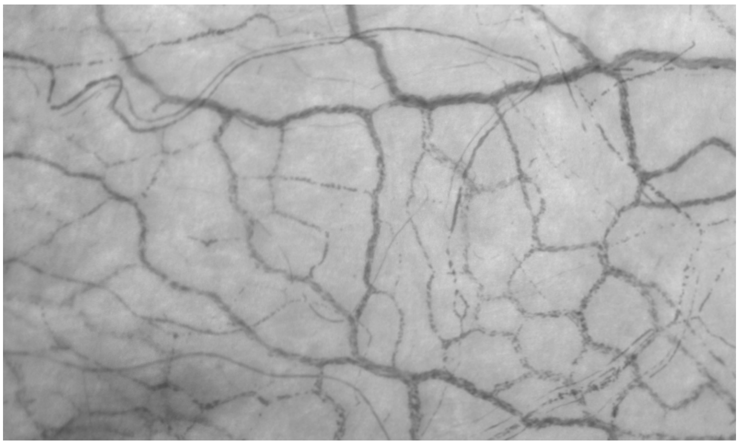


Figure 4: An example of a single frame from a haemoglobin video image.

with images in which the moving part is a dense collection of small particles, although in some parts of the image the heamoglobin has disintegrated into several distinct objects. Moreover, we see that in the main part of the image, the structure stays the same within consecutive frames. Therefore, we would best classify this data set as class 2a, if we take the classification from section 2.1.

In [8], an attempt has been made to obtain a segmentation between the blood vessels and the background of the image. It seemed that obtaining a segmentation was rather hard, when information about the motion was not taken into account. On the other hand, we can imagine that trying to find an estimate of the motion, without knowing a segmentation, can also be difficult, since the motion is then not restricted to the blood vessels. Therefore, in this thesis, we make a first attempt to obtain both in a joint matter.

2.2.3 Flow in human brain

In the last image sequence, we see the flow of blood in the human brain. The data was obtained by the Radboud University Medical Center by imaging a contrast agent in blood.

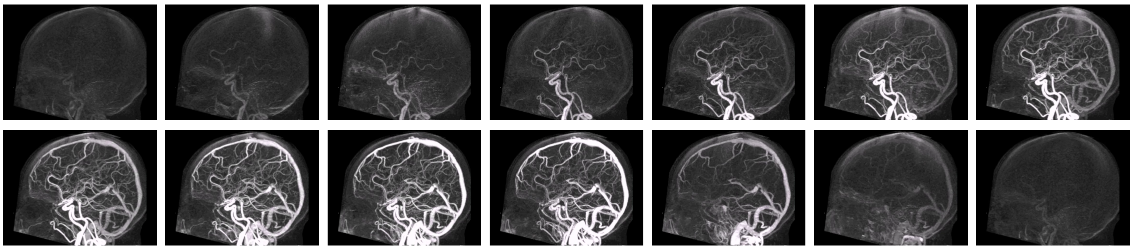


Figure 5: Consecutive inflow, spreading and outflow of blood in a human brain.

In Figure 5 we see the inflow of blood on a small part of the bottom boundary, after which it accumulates within the brain, followed by an outflow at the bottom boundary. The properties of this flow are similar to the haemoglobin flow: we have a collection of small particles moving through a fine vascular structure. A big difference is that we now have a much smoother flow than in the haemoglobin case; we do not see a specific structure within the blood. Consequently, we can classify it as class 2b in the classification from section 2.1.

3 Optimal transport

In this chapter, we will define two models, which we will use to find both an interpolation between two images, as well as a velocity field that belongs to this interpolation. First, we will explain the mass transport problem from the original statement by Monge. Next, we will set the transport problem in a fluid dynamics setup, which allows us to define our first model. Then we will add some regularity on our velocity field, which gives us our second model. For both models, we will give the option to incorporate anisotropy, in order to make our model suitable for the vascular applications we have in mind. Finally, there will be an analysis on the existence and uniqueness of a solution to our model.

3.0.1 Monge-Kantorovich problem

Optimal transport as a mass transfer problem has its origins in the end of the eighteenth century, as it was first formalised by Monge [30] in the year 1781. He considered the problem in which "the transport of excavated material is such that the sum of the products of the molecules by the space traversed is a minimum between the roads of any two points". In the years 1942 [22] and 1948 [23], Kantorovich treated the same problem in a modern way, leading to the nowadays called 'Monge-Kantorovich problem', which is defined as follows:

Consider two density functions $\rho_0(x) \geq 0$ and $\rho_1(x) \geq 0$, with $x \in \Omega \subset \mathbb{R}^d$, such that both densities have equal mass C:

$$\int_\Omega \rho_0(x)\,\mathrm{d}x = \int_\Omega \rho_1(x)\,\mathrm{d}x = C. \tag{3.1}$$

A map T transports ρ_0 to ρ_1 if, for all subsets $U \subset \Omega$, the following holds:

$$\int_U \rho_1(x)\,\mathrm{d}x = \int_{T(U)} \rho_0(x)\,\mathrm{d}x, \tag{3.2}$$

which reduces to

$$|\det(\nabla T(x))|\rho_1(T(x)) = \rho_0(x), \tag{3.3}$$

when T is a smooth one-to-one map. One can easily verify that for all densities ρ_0 and ρ_1 that are not dirac-measures, one can think of different transport maps that satisfy (3.3). For instance, if we take $\Omega = \mathbb{R}$, and take the transport of the rectangular function

$$\rho_0(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq 1, \\ 0 & \text{else} \end{cases}$$

to the function

$$\rho_1(x) = \begin{cases} 1 & \text{for } 2 \leq x \leq 3, \\ 0 & \text{else} \end{cases}$$

we can already state the following two transport maps:

- linear shift: $T_1(x) = x + 2$;
- linear shift with mirroring: $T_2(x) = -x + 3$.

Clearly, both transport maps satisfy (3.2) and (3.3), as can be seen in Figure 6. This means that a solution to the Monge-Kantorovich is in general not unique, and thus it makes sense to introduce a cost functional, which we will try to minimise. One such a cost functional is the L^p-Wasserstein distance ($p \geq 1$), defined by:

$$d_p(\rho_0, \rho_1)^p = \inf_{T \in \mathcal{T}} \int_\Omega \|T(x) - x\|^p \rho_0(x)\,\mathrm{d}x, \tag{3.4}$$

where $\mathcal{T}$ is the set of all possible maps that map ρ_0 to ρ_1. The map T^* that minimises (3.4) is called the optimal transport. For our made-up one-dimensional example with gaussians, and choosing $p > 1$, we see that we would prefer the linear shift without mirroring.

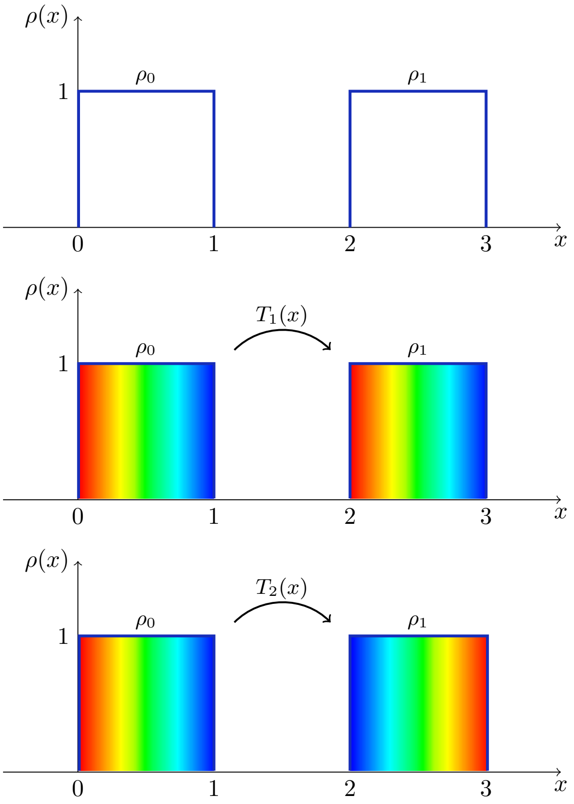


Figure 6: Two different transport maps that have the same ρ_0 and ρ_1. Colour-coding is used to emphasise the difference.

3.1 Model 1: isotropic and anisotropic transport

For the first model, we consider two types of transports: isotropic transport, where we do not take direction into account, and anisotropic transport, where we can deal with the information we might have about the expected direction of the transport.

3.1.1 Isotropic optimal transport

In 2000, Benamou and Brenier [6] showed that in the case of a L^2-Wasserstein distance and for $\Omega = \mathbb{R}^d$, solving problem (3.4) is equivalent to solving the following fluid dynamics problem:

$$\min_{(\rho,v)\in MC_{\rho v}} \frac{1}{2}\int_0^1 \int_{\mathbb{R}^d} \rho(x,t)|v(x,t)|^2 \,\mathrm{d}x\,\mathrm{d}t, \tag{3.5}$$
$$\text{where } MC_{\rho v} = \{(\rho, v) \mid \partial_t \rho + \operatorname{div}(\rho v) = 0,\ \rho(\cdot, 0) = \rho_0,\ \rho(\cdot, 1) = \rho_1,\ \rho \geq 0\},$$

where $\rho(x,t) \in \mathbb{R}$ and $v(x,t) \in \mathbb{R}^d$ are respectively the density and the velocity at time $t \in [0,1]$ and in space $x \in \mathbb{R}^d$. We see that solving (3.5) means finding a density and velocity in a time-space domain, that satisfies the continuity equation for mass (density), that satisfies our initial conditions ρ_0 and ρ_1 and that has minimal kinetic energy in the whole domain.

In order to extend this result to an arbitrary domain $\Omega \subset \mathbb{R}^d$, we need to take a look at the boundary conditions of our mass conservation set $MC_{\rho v}$. It can be shown [25] that the condition $\partial_t \rho + \mathrm{div}(\rho v)$ for all ρ and v, in combination with the boundary condition $\rho v \cdot n|_{\partial\Omega} = 0$, is enough to ensure mass conservation. However, we can think of examples in which we do have an inflow and/or outflow of density. In this case, this boundary condition is thus not a good choice. Let us consider two cases, that are likely to occur when handling transport problems with vascular applications:

1. The total density does not change between ρ_0 and ρ_1: $\int_\Omega \rho_0 \,\mathrm{d}x = \int_\Omega \rho_1 \,\mathrm{d}x$. We still might expect an inflow or outflow, so choosing $\rho v \cdot n|_{\partial\Omega} = 0$ is not an option. In this case we could enforce $\int_\Omega \rho(x,t)\,\mathrm{d}x = \int_\Omega \rho_0 \,\mathrm{d}x = \int_\Omega \rho_1 \,\mathrm{d}x$ for all t. An example where we see such behaviour is the one in Figure 4.
2. The total density increases or decreases between ρ_0 and ρ_1, because of an inflow or outflow. In many images, it is reasonable to assume the inflow or outflow constant, like the example in Figure 5. In this case, we could enforce $\int_\Omega \rho(x,t)\,\mathrm{d}x = t \cdot \int_\Omega \rho_0 \,\mathrm{d}x + (1-t) \cdot \int_\Omega \rho_1 \,\mathrm{d}x$. Note that this case does not fulfil the original Monge-Kantorovich equation (3.1), but in our new fluid dynamics setup, we can solve this problem.

There are many more cases that could be considered, but it is very hard to ensure the mass conservation if we do not have prior knowledge, such as specific boundary conditions. We will show in section 6.1 how we will deal with the ones discussed here. For the ease of readability, we will from now on only consider a mass conservation set, in which $\int_\Omega \rho_0 \,\mathrm{d}x = \int_\Omega \rho_1 \,\mathrm{d}x$ and inflow or outflow is prohibited.

Following [6] and [36], we can apply the change of variables $(\rho, m) := (\rho, \rho v)$, such that we use the momentum as a variable instead of the velocity, in order to obtain a convex optimization problem in (ρ, m):

$$\min_{(\rho,m)\in MC} \int_0^1 \int_\Omega J(\rho,m)\,\mathrm{d}x\,\mathrm{d}t,$$

$$\text{where } J(\rho,m) = \begin{cases} \frac{|m|^2}{2\rho} & \text{if } \rho > 0, \\ 0 & \text{if } (\rho,m) = (0,0), \\ \infty & \text{otherwise.} \end{cases}$$

$$\text{and } MC = \{(\rho,m) \mid \partial_t \rho + \mathrm{div}(m) = 0,\ \rho(\cdot,0) = \rho_0,\ \rho(\cdot,1) = \rho_1,\ m \cdot n|_{\partial\Omega} = 0,\ \rho \geq 0\}.$$

We write our mass-conservation constraint set as an indicator function $\iota_{MC}(\rho,m)$, which holds the value 0 if the variables are in the set MC and ∞ if they are not. This notation is useful, because we will make use of the proximal splitting method and for both $J(\rho,m)$ and $\iota_{MC}(\rho,m)$ proximal operators can now be defined (section 6.1). We have the following (unconstrained) convex minimisation problem:

$$\min_{(\rho,m)} \mathcal{J}(\rho,m) = \min_{(\rho,m)} \int_0^1 \int_\Omega J(\rho,m) + \iota_{MC}(\rho,m)\,\mathrm{d}x\,\mathrm{d}t, \tag{3.6}$$

$$\text{where } J(\rho,m) = \begin{cases} \frac{|m|^2}{2\rho} & \text{if } \rho > 0, \\ 0 & \text{if } (\rho,m) = (0,0), \\ \infty & \text{otherwise.} \end{cases}$$

$$\text{and } MC = \{(\rho,m) \mid \partial_t \rho + \mathrm{div}(m) = 0,\ \rho(\cdot,0) = \rho_0,\ \rho(\cdot,1) = \rho_1,\ m \cdot n|_{\partial\Omega} = 0,\ \rho \geq 0\}.$$

It is easy to see that our functional is indeed a convex one, although not continuous, and thus not differentiable everywhere. Therefore we can not rely on algorithms that make use of the gradient, such as the gradient descent method. Papadakis *et al.* [36] used the technique of proximal splitting to solve (3.6) for different initial conditions ρ_0 and ρ_1. We will follow this approach, on which we will elaborate in chapter 6.

Instead of an L^2-Wasserstein distance, one could also choose to use a general L^p-Wasserstein distance or an H^{-1}-Sobolev distance [36]. Different cost functionals lead to different minimisation problems, as can be seen in table 1. Moreover, the type of transport that is preferred differs for every cost functional. For instance, with an L^1-Wasserstein distance, sparse transports in space are preferred and sparse transports in time are possible, whereas sparsity is not preferred in the L^2-case. For a more extensive description on the use of different transport costs, we refer to appendix A.

L^2-Wasserstein	$\rho\|v\|^2$	$\frac{\|m\|^2}{\rho}$
L^1-Wasserstein	$\rho\|v\|$	$\|m\|$
L^p-Wasserstein for $p \geq 1$	$\rho\|v\|^p$	$\frac{\|m\|^p}{\rho^{p-1}}$
H^{-1}-Sobolev	$\rho^2\|v\|^2$	$\|m\|^2$
Interpolated L^2/H^{-1}	$\rho^{2-p}\|v\|^2$	$\frac{\|m\|^2}{\rho^p}$

Table 1: Overview of the effect of different cost functionals on the functional $J(\rho, m)$.

3.1.2 Anisotropic optimal transport

When we think of vascular structures, we see that flows are often oriented in a specific direction within a (small) region. If we know the 'preferred' direction of the flow, we might want to use this in the modelling of our transport. This brings us to a functional in which the cost of the transport depends on the direction of the flow. Hug *et al.* [21] proposed to extend (3.6) to the following anisotropic minimisation problem:

$$\min_{(\rho,m)} \mathcal{J}(\rho,m) = \min_{(\rho,m)} \int_0^1 \int_\Omega J_A(\rho,m) + \iota_{MC}(\rho,m)\,\mathrm{d}x\,\mathrm{d}t, \tag{3.7}$$

$$\text{where } J_A(\rho,m) = \begin{cases} \frac{m^T A m}{2\rho} & \text{if } \rho > 0, \\ 0 & \text{if } (\rho,m) = (0,0), \\ \infty & \text{otherwise.} \end{cases}$$

$$\text{and } MC = \{(\rho,m) \mid \partial_t\rho + \mathrm{div}(m) = 0,\ \rho(\cdot,0) = \rho_0,\ \rho(\cdot,1) = \rho_1,\ m \cdot n|_{\partial\Omega} = 0,\ \rho \geq 0\}.$$

Notice that for $A = \mathrm{Id}_d$ (identity matrix), where d is the space dimension of our image, (3.7) reduces to the isotropic equation (3.6). In chapter 4, we will explain different ways of obtaining a useful anisotropy matrix A. Existence and uniqueness of a solution to (3.7) have been shown in [21], when A only depends on x, not on t.

3.2 Model 2: flow regularised transport

Our second model will be an extension on our first model, in which we allow to incorporate a regulariser on the velocity of the flow. A motivation for regularising the velocity is given in section 3.2.1. In section 3.1.1, we made the change of variables from v to m. However, the regularisers that we will consider are a function of v. It is not possible to make the same change of variables, since the regularisers would not be convex anymore. This means that we need to make a coupling between ρ, m and v in order to use regularisers based on the velocity. We will follow the approach of [21], where the coupling term is a L^2-coupling which penalises the squared distance between m and ρv:

$$\mathcal{K}(\rho, m, v) = \frac{1}{2}\int_0^1 \int_\Omega |m - \rho v|^2 \,\mathrm{d}x \,\mathrm{d}v. \tag{3.8}$$

Considering the energy functional, the regulariser and the coupling term, we obtain the following minimisation problem:

$$\begin{aligned} &\min_{(\rho,m,v)} \big\{\mathcal{J}(\rho,m) + \lambda\mathcal{K}(\rho,m,v) + \mu\mathcal{R}(v)\big\} \\ &= \min_{(\rho,m,v)} \int_0^1 \int_\Omega J_A(\rho,m) + \iota_{MC}(\rho,m) + \lambda K(\rho,m,v) + \mu R(v) \,\mathrm{d}x \,\mathrm{d}t, \end{aligned} \tag{3.9}$$

$$\text{where } J_A(\rho,m) = \begin{cases} \frac{m^T A m}{2\rho} & \text{if } \rho > 0, \\ 0 & \text{if } (\rho,m) = (0,0), \\ \infty & \text{otherwise.} \end{cases}$$

$$MC = \{(\rho,m) \mid \partial_t\rho + \operatorname{div}(m) = 0,\ \rho(\cdot,0) = \rho_0,\ \rho(\cdot,1) = \rho_1,\ m\cdot n|_{\partial\Omega} = 0,\ \rho \geq 0\},$$

$$K(\rho,m,v) = \frac{1}{2}|m - \rho v|^2,$$

and $R(v)$ some convex regularisation functional,

where $\lambda, \mu \in \mathbb{R}$ are weighting parameters. Here we only considered the more general anisotropic model, since the isotropic model is simply a special case in which $A = \mathrm{Id}_d$. In section 3.3, we will prove existence of a solution to (3.9). Moreover, we will discuss the uniqueness of a solution.

3.2.1 Physical priors as regularisers

For the rest of this section, we will assume $\Omega \subset \mathbb{R}^2$, but it should be kept in mind, that in general, we can generalise this to a domain in any dimension.

As a motivation for using specific regularisers, we will consider the following example: Let $\Omega = [0,1]^2 \subset \mathbb{R}^2$, let ρ_0 be a gaussian with mean $(\frac{1}{5}, \frac{1}{5})$ and some variance σ_0^2 such that its value is very low at $\partial\Omega$. Let ρ_1 be a gaussian with mean $(\frac{4}{5}, \frac{4}{5})$ and variance σ_0^2. Remember from (3.5) that our cost in v is quadratic. In a simple case as this one, the optimal transport is such that we get a constant velocity directed from ρ_0 to ρ_1: if the density would first move quickly and then slowly, because of the quadratic penalisation, we would end up with a higher cost; if the density would not follow a straight line, the travelled distance would be larger, which needs a higher velocity and thus a higher cost. The movement is shown in Figure 7.

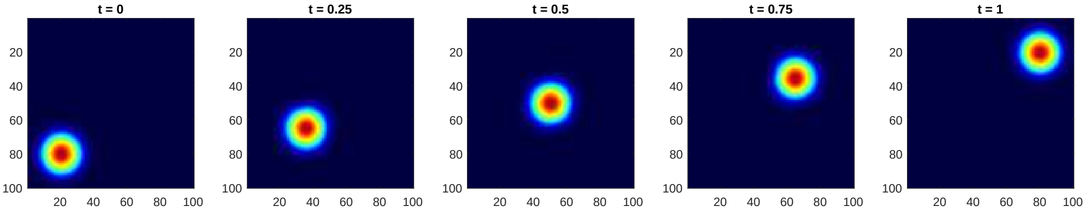


Figure 7: A Gaussian density moving from one corner to the other in a squared domain.

Now if we change our space domain to an L-shaped $\Omega = [0,1] \times [0, \frac{2}{5}] \cup [\frac{3}{5}, 1] \times [0,1] \in \mathbb{R}^2$, it is clear that a direct route is not available anymore, since part of this route is outside our domain. It is intuitive that every point in the gaussian density 'will try to find' a path that is as short as possible. This means that every point in the density will make a sharp turn at the corner of our domain. As a consequence, at $t = \frac{1}{2}$, there will be a very high density around the corner of our

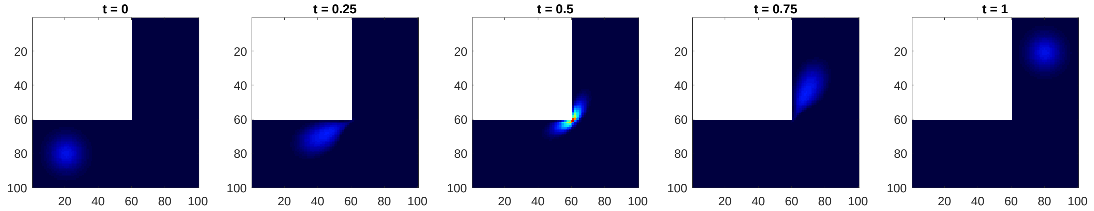


Figure 8: A Gaussian density moving from one corner to the other in a L-shaped domain.

domain; in fact, we will obtain a Dirac measure at the corner. This movement is shown in Figure 8.

In many applications, like the transport of a cell, such an accumulation of density in a small neighbourhood is not expected, so we might need a regulariser R to deal with this behaviour. An accumulation of density means that the transported structure is compressed during the transport, so we are looking for a flow where compression is prohibited or penalised. From the field of fluid dynamics, it is well known that an incompressible flow is one where $\operatorname{div}(v) = 0$ everywhere. Note that incompressibility does not mean that the structure of the density may not change at all. However, it does mean that if we made a histogram of the density distribution in space, it would stay exactly the same over time, since the structure can not compress or expand. An easy to implement regulariser would be an L^2-penalisation of the divergence of our flow v:

$$\mathcal{R}_D(v) = \frac{1}{2}\int_0^1 \int_\Omega |\operatorname{div}(v)|^2 \,\mathrm{d}x\,\mathrm{d}t \tag{3.10}$$

A second option for regularisation would be one where rigidity of the flow is preferred. When we, for example, model a car that moves from one place to the other, we know for certain that the car should not deform. However, it is allowed to translate or rotate. From [25] and [21], we know that for $\Omega \subset \mathbb{R}^2$, the condition $\nabla v + (\nabla v)^T = 0$ for every $x \in \Omega$ is enough to guarantee rigidity. Here ∇v is the Jacobian matrix of v:

$$\nabla v = \begin{pmatrix} \dfrac{\partial v_1}{\partial x_1} & \dfrac{\partial v_1}{\partial x_2} \\ \dfrac{\partial v_2}{\partial x_1} & \dfrac{\partial v_2}{\partial x_2} \end{pmatrix}$$

Again using an L^2-penalisation, we obtain the following regulariser:

$$\mathcal{R}_R(v) = \frac{1}{2}\int_0^1 \int_\Omega \left\| \frac{\nabla v + (\nabla v)^T}{2} \right\|^2 \mathrm{d}x\,\mathrm{d}t, \tag{3.11}$$

where the norm is induced by the pointwise product of the matrix entries.

Comparison between the regularisers

If we expand the integrands of the regularisers considered here, we obtain:

$$\|\operatorname{div}(v)\|^2 = \left(\frac{\partial v_1}{\partial x_1}\right)^2 + \left(\frac{\partial v_2}{\partial x_2}\right)^2 + 2\left(\frac{\partial v_1}{\partial x_1}\right)\left(\frac{\partial v_2}{\partial x_2}\right);$$

$$\left\| \frac{\nabla v + (\nabla v)^T}{2} \right\|^2 = \left(\frac{\partial v_1}{\partial x_1}\right)^2 + \left(\frac{\partial v_2}{\partial x_2}\right)^2 + \frac{1}{2}\left(\frac{\partial v_1}{\partial x_2}\right)^2 + \frac{1}{2}\left(\frac{\partial v_2}{\partial x_1}\right)^2 + \left(\frac{\partial v_1}{\partial x_2}\right)\left(\frac{\partial v_2}{\partial x_1}\right).$$

Apart from the first two terms on the right hand sides, we see that both regularisers are actually very different. This is rather unexpected, since a rigid flow also fulfils the incompressibility condition (something that is rigid cannot compress or expand). In chapter 7, we will see various examples where both regularisers give a similar result, but also ones where they give very different results.

3.2.2 Anisotropic regulariser

A third regularisation functional can be thought of when taking our application of vascular structures in mind. For example, when we think of blood flow in blood vessels, we expect no flow outside the blood vessels, while there is a rather smooth flow within the blood vessels. This means that the gradients of v_1 and v_2 are expected to be close to zero if we move along the direction of the vessel, while the gradient can take a very high value if we move perpendicular to the direction of the vessel. Moreover, the smoothing should be the same for both v_1 and v_2, since both v_1 and v_2 should not change much within the blood vessel, unregarded the direction of the flow.

In order to get a general framework in which we can incorporate anisotropy, it might be wise to state a homogeneous isotropic regulariser, which we can adapt to make it as we would like it to be. Motivated by our wish to treat v_1 and v_2 in the same manner, we can consider the regulariser

$$R_I(v) = \frac{1}{2}\sum_{i=1}^{2} |\nabla v_i|^2 = \frac{1}{2}\sum_{i=1}^{2} (\nabla v_i)^T (\nabla v_i) = \frac{1}{2}\sum_{i=1}^{2} (\nabla v_i)^T \mathrm{Id}_2 (\nabla v_i), \tag{3.12}$$

which is the same regulariser Horn and Schunck [20] used in their famous method for determining optical flow. From writing it as in (3.12), it is immediately clear that we can make the expression anisotropic if we replace the identity matrix Id_2 by an anisotropy matrix D:

$$R_A(v) = \frac{1}{2}\sum_{i=1}^{2} (\nabla v_i)^T D (\nabla v_i). \tag{3.13}$$

The matrix D is often referred to as the diffusion tensor [44]. There are many ways to construct the diffusion tensor. We can separate two types of diffusion, by choosing on which variables it should depend:

1. **flow-driven diffusion.** The most intuitive is to let D depend on the flow itself: if we see a very different flow in one area compared to a flow in an adjacent one, we do not want to smooth across the boundary between the areas, but do want to smooth along this boundary. We call this type a flow-driven diffusion tensor. One potential problem with this type of diffusion-tensor is that v is not known beforehand. Therefore, we cannot see what the diffusion-tensor looks like, without first applying our model and calculating the optimal transport.

2. **intensity-driven diffusion.** Often, flow-edges are located at the same locations as the intensity-edges. When this is the case, we can make use of a intensity-driven diffusion tensor. This diffusion does not have the problem of the flow-driven case that the tensor is not known before we solve the optimal transport problem. We can first choose the desired diffusion, before applying the optimal transport model.

In our applications, we see that there is a rather big difference in intensities between areas where we expect flow and areas where we do not expect flow. For that reason, we will now focus ourselves on intensity driven diffusion tensors, but we should always keep in mind that it is possible to change this when applying our model to different images.

Edge detecting diffusion tensor

Nagel and Enkelmann [33] defined the following edge detecting tensor, which makes use of the first derivatives of the image intensity in every direction:

$$D_{ED}(\rho) = \frac{1}{|\nabla \rho|^2 + 2\varepsilon^2} \begin{pmatrix} \left(\frac{\partial \rho}{\partial x_2}\right)^2 + \varepsilon^2 & -\frac{\partial \rho}{\partial x_1}\frac{\partial \rho}{\partial x_2} \\ -\frac{\partial \rho}{\partial x_1}\frac{\partial \rho}{\partial x_2} & \left(\frac{\partial \rho}{\partial x_1}\right)^2 + \varepsilon^2 \end{pmatrix}, \tag{3.14}$$

where ε is a small parameter to ensure regularity of the matrix. To see the effect of the tensor, think of an image where $\frac{\partial \rho}{\partial x_2}$ is very large and $\frac{\partial \rho}{\partial x_1}$ is zero. We then obtain a tensor close to $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, which means that we only smooth in the x_1-direction and not in the x_2-direction. This is exactly what was desired, since the edge is parallel to the x_1-direction, because $\frac{\partial \rho}{\partial x_2}$ is large. This does not only work for edges parallel to the axes of the image, but for edges in any direction. In order to reduce the influence of noise, we could first convolve ρ with a Gaussian filter with variance σ_1^2 before applying the diffusion tensor.

Coherence enhancing diffusion tensor

Another option is the coherence enhancing diffusion tensor, proposed by Weickert [44]. This diffusion tensor is based on the structure tensor, which is constructed as follows: first the density ρ is smoothed by a Gaussian filter with variance σ_1^2 to get a smoothed image ρ_{σ_1}. Then we take the gradient of this density and compute the structure tensor

$$T_0(\nabla \rho_{\sigma_1}) = \nabla \rho_{\sigma_1} \nabla \rho_{\sigma_1}^T = \begin{pmatrix} \left(\frac{\partial \rho_{\sigma_1}}{\partial x_1}\right)^2 & \frac{\partial \rho_{\sigma_1}}{\partial x_1}\frac{\partial \rho_{\sigma_1}}{\partial x_2} \\ \frac{\partial \rho_{\sigma_1}}{\partial x_1}\frac{\partial \rho_{\sigma_1}}{\partial x_2} & \left(\frac{\partial \rho_{\sigma_1}}{\partial x_2}\right)^2 \end{pmatrix}, \tag{3.15}$$

When convolving the structure tensor component-wise with a different Gaussian filter with variance σ_2^2, we obtain the structure tensor

$$T_{\sigma_2}(\rho_{\sigma_1}) = (\nabla \rho_{\sigma_1} \nabla \rho_{\sigma_1}^T)_{\sigma_2}. \tag{3.16}$$

From the structure tensor, we can extract the (orthonormal) eigenvectors and corresponding eigenvalues. The eigenvector w_1 has a direction perpendicular to the edge direction and w_2 is the eigenvector parallel to the edge direction. The corresponding eigenvalues are denoted by μ_1 and μ_2. Our desired diffusion tensor should have the property that it diffuses strongly in the direction of w_2 and very weakly in the direction of w_1. Weickert [44] proposed to construct a diffusion tensor D_{CE} with the same eigenvectors as $T_{\sigma_2}(\rho_{\sigma_1})$ and with eigenvalues

$$\begin{aligned} \lambda_1 &= \alpha, \\ \lambda_2 &= \begin{cases} \alpha & \text{if } \mu_1 = \mu_2, \\ \alpha + (1-\alpha)\exp\left(\frac{-1}{(\mu_1-\mu_2)^2}\right) & \text{else.} \end{cases} \end{aligned} \tag{3.17}$$

Here $0 < \alpha \ll 1$ is a very small parameter that keeps D_{CE} uniformly positive definite. We see that smoothing in the direction of w_1 is very small, as is desired. The strength of the smoothing in the direction of w_2 depends on the difference between μ_1 and μ_2: if the structure tensor $T_{\sigma_2}(\rho_{\sigma_1})$ does not have a strong preference for one direction ($\mu_1 \approx \mu_2$), we get roughly the same smoothing in both directions. However, if a structure is clearly visible, $T_{\sigma_2}(\rho_{\sigma_1})$ will have a strong preference for one direction and we get a very strong smoothing in the direction of w_2. Note that $\alpha \leq \lambda_2 < 1$.

One of the big advantages of using C_{CE} is that we are free to choose our σ_1 and σ_2, depending on our intensity ρ. The choice of σ_1 should depend on the amount of noise in the image. The choice of σ_2 should scale with the size of the structures one would like to detect.

Quadratic form

In chapter 6 we will see that it is advantageous to write (3.13) in a quadratic form, such that we can write it as an L^2-norm of an operator working on v.

By calculating the leading principal minors of $D_{ED}(\rho)$, we know that it is positive definite. Moreover, $D_{CE}(\rho)$ was constructed with strictly positive eigenvalues, so this tensor is also positive definite. Therefore, we can find a unique Cholesky-factorisation $D = U^T U$, where U is upper triangular, and write the integral of (3.13) as

$$\mathcal{R}_A(v) = \frac{1}{2}\sum_{i=1}^{2} \|U\nabla v_i\|_2^2 = \frac{1}{2}\left\|\big(U\nabla v_1, U\nabla v_2\big)\right\|_2^2. \tag{3.18}$$

where in the last term, there is no summation outside the norm anymore.

3.3 Analysis

3.3.1 Convexity of model 2

It can be shown that our minimisation problem is separately convex in (ρ, m) and v, that is: if we fix v, it is convex in (ρ, m) and if we fix (ρ, m), it is convex in v. However, it is not convex in all the variables, without fixing at least one of them. This means that if we find a minimiser, we will not know if it is unique or if it is a global minimum. Because of this, we should be careful with using (3.9) if we do not have a decent initial guess for the velocity.

3.3.2 Existence of a minimiser of model 2

In this section we will prove the existence of a minimiser of the isotropic variant of our second model. In [21], a proof has been given with $R(v) = \|\nabla v\|^2$ as a regulariser. Here we will give a proof for the second model with a rigidity regulariser.

Before we begin the proof, let us define the function spaces to which our variables (ρ, m, v) belong to. Firstly, we will denote our space-time domain by $Q = \Omega \times [0,1]$. Furthermore, we assume that ρ is bounded in our complete domain by some bound M, therefore, we set $\rho \in L^\infty(Q)$ and assume $\|\rho\|_{L^\infty(Q)} \leq M$. Note that by this, we restrict ourself to the case that the density does not compress infinitely much. However, when we are in a discretised setup, we will always have this upper bound available, since the density in one pixel will not be higher than the complete density in Ω. Now, for m, we see that it appears in a quadratic form in $J(\rho, m)$; therefore it is natural to set it in the space $L^2(Q)^d$, where d is the dimension of our problem. Finally, because of our choice for $R(v)$, we need ∇v to be bounded in the L^2-norm. In principle, we do not need v to be bounded in the L^2-norm itself, since the regulariser only depends on the gradient of v. However, as will become clear in the proof, the boundedness of v in L^2 allows us to get convergence of the sequence $\rho_n v_n$ to ρv, which will be needed. Therefore, we set v in the Bochner space $L^2(0,1;H^1(\Omega))$. By writing our mass-conservation set in a weak form, we end up with the following set of constraints for the variables in our minimisation problem:

$$\begin{aligned} U = \big\{&(\rho, m, v) \in L^\infty(Q) \times L^2(Q)^d \times L^2(0,1;H^1(\Omega))^d : \ \|\rho\|_{L^\infty(Q)} \leq M, \\ &\forall \phi \in C_\Omega^\infty(Q), \ \int_0^1 \int_\Omega \rho \partial_t \phi + m \cdot \nabla \phi \, \mathrm{d}x \, \mathrm{d}t + \int_\Omega \rho_0(x)\phi(x,0) - \rho_1(x)\phi(x,1) \, \mathrm{d}x = 0\big\}, \end{aligned}$$

where $C_\Omega^\infty(Q) := \{\phi \in C^\infty(Q) : \; \phi|_{\partial\Omega\times[0,1]} = 0\}$. It can be seen that this set is non-empty, since $((1-t)\rho_0 + t\rho_1, m, 0)$ with $\operatorname{div}(m) = \rho_0 - \rho_1$ is an element of U. One can always construct such an m by simply taking $\partial_{x_i} m_i = (\rho_0 - \rho_1)/d$ and $\partial_{x_j} m_i = 0$ for $i \neq j$. Therefore we have at least one minimising sequence in U, which is needed for our proof.

Proposition 1. *The problem*

$$\min_{(\rho,m,v)\in U} \int_0^1 \int_\Omega J(\rho,m) + \frac{\lambda}{2}|m-\rho v|^2 + \frac{\mu}{2}\left\|(\nabla v + (\nabla v)^T)/2\right\|^2 \mathrm{d}x\,\mathrm{d}t, \tag{3.19}$$

$$\textit{with } J(\rho,m) = \begin{cases} \frac{|m|^2}{2\rho} & \textit{if } \rho > 0, \\ 0 & \textit{if } (\rho,m) = (0,0), \\ \infty & \textit{otherwise}, \end{cases}$$

and U defined as above, has at least a (local) minimiser.

The outline of the proof will be as follows:

1) We show that (3.19) is bounded from below and thus has a minimum.

2) Let (ρ_n, m_n, v_n) be a minimising sequence. We then find an upper bound for our (combination of) variables. Having an upper bound, we use a corollary [4, Thm. 2.1.1] of the Banach-Alaoglu Theorem [40, Thm. 3.15] to give us the existence of a limit $(\rho, m, v) \in U$ in the weak or weak-* convergence of (ρ_n, m_n, v_n).

3) To proof that the functional working on our weak limit does give us the minimum, we need to show that (3.19) is weakly lower semi-continuous. Moreover, we need to show that the limit of a product of sequences is equal to the product of the limits of sequences for combinations of variables in (3.19).

Proof.

1) We see that all the integrands in (3.19) are bounded from below by zero: the second and third term are both norms and therefore by definition non-negative. For $J(\rho,m)$, we only have to check if it is positive for $\rho > 0$, which is readily obtained by using $\|\rho\|_{L^\infty(Q)} \leq M$: $\frac{|m|^2}{2\rho} \geq \frac{|m|^2}{2M} \geq 0$. So (3.19) has a minimum.

2) Now let $(\rho_n, m_n, v_n) \in U$ be a minimising sequence. For $J(\rho,m)$, we can make the transformation

$$w := \begin{cases} \frac{m}{\sqrt{\rho}} & \text{if } \rho > 0 \\ 0 & \text{if } (\rho,m) = 0 \\ \infty & \text{else.} \end{cases}$$

The first term of (3.19) can now be rewritten

$$\int_0^1 \int_\Omega J(\rho,m)\,\mathrm{d}x\,\mathrm{d}t = \frac{1}{2}\|w\|^2_{L^2(Q)}.$$

Looking at (3.19) and our constraint set U, it is natural to impose the following a priori bounds: there exists a $M, C \geq 0$ such that for all $n \in \mathbb{N}$,

$$\|\rho_n\|_{L^\infty(Q)} \leq M, \qquad \|w_n\|_{L^2(Q)} \leq C, \qquad \|v_n\|_{L^2(0,1;H^1(\Omega))^d} \leq C, \qquad \|m_n - \rho_n v_n\|_{L^2(Q)} \leq C.$$

We will show that these a priori bounds are enough to obtain bounds for all variables in (3.19). Firstly, we have

$$\left\|(\nabla v_n + (\nabla v_n)^T)/2\right\|_{L^2(Q)^{d\times d}} \leq \|\nabla v_n\|_{L^2(Q)^{d\times d}} \leq \|v_n\|_{L^2(0,1;H^1(\Omega))^d}.$$

Furthermore, since w_n is bounded in the L^2-norm, we know that $w_n < \infty$ a.e. Therefore, we also know $m_n = w_n\sqrt{\rho_n}$ a.e. Consequently, we have the bounds

$$\|m_n\|_{L^2(Q)} \leq \|w_n\|_{L^2(Q)} \|\sqrt{\rho_n}\|_{L^\infty(Q)} \leq C\sqrt{M},$$

and

$$\|\rho_n v_n\|_{L^2(Q)} \leq \|m_n\|_{L^2(Q)} + \|m_n - \rho_n v_n\|_{L^2(Q)} \leq C\sqrt{M} + C.$$

Left to show is the boundedness of the variables in the mass-conservation set. Obviously, $L^2(0,1;H_0^1(\Omega)) \cap C_\Omega^\infty(Q) \subset C_\Omega^\infty(Q)$. By choosing a test function $\phi \in L^2(0,1;H_0^1(\Omega)) \cap C_\Omega^\infty(Q)$ for which $\phi(x,0) = \phi(x,1) = 0$, and using the definition of U, we see that the following holds:

$$\int_0^1 \int_\Omega \rho_n \partial_t \phi + m_n \cdot \nabla \phi \, \mathrm{d}x \, \mathrm{d}t = 0.$$

Since the dual of $L^2(0,1;H_0^1(\Omega))$ is $L^2(0,1;H^{-1}(\Omega))$ [11], we can derive a bound for both $\mathrm{div}(m)$ and $\partial_t \rho$. Using integration by parts and applying Hölder's inequality, we obtain

$$\begin{aligned}\left|\int_0^1 \int_\Omega \partial_t \rho_n \phi \, \mathrm{d}x \, \mathrm{d}t\right| &= \left|\int_0^1 \int_\Omega -\rho_n \partial_t \phi \, \mathrm{d}x \, \mathrm{d}t - \int_\Omega \rho_0(x)\phi(x,0) \, \mathrm{d}x + \int_\Omega \rho_1(x)\phi(x,1) \, \mathrm{d}x\right| \\ &= \left|\int_0^1 \int_\Omega m_n \cdot \nabla \phi \, \mathrm{d}x \, \mathrm{d}t\right| \leq \|m_n\|_{L^2(Q)} \|\nabla \phi\|_{L^2(Q)},\end{aligned}$$

$$\begin{aligned}\left|\int_0^1 \int_\Omega \mathrm{div}(m_n)\phi \, \mathrm{d}x \, \mathrm{d}t\right| &\leq \left|\int_0^1 \int_\Omega m_n \cdot \nabla \phi \, \mathrm{d}x \, \mathrm{d}t\right| + \left|\int_0^1 \int_{\partial\Omega} (m_n \cdot n)\phi \, \mathrm{d}s \, \mathrm{d}t\right| \\ &= \left|\int_0^1 \int_\Omega m_n \cdot \nabla \phi \, \mathrm{d}x \, \mathrm{d}t\right| \leq \|m_n\|_{L^2(Q)} \|\nabla \phi\|_{L^2(Q)},\end{aligned}$$

where we have used that $\phi|_{\partial\Omega \times [0,1]} = 0$. Using these inequalities and using the definition of the norm, we obtain our bounds

$$\begin{aligned}\|\partial_t \rho_n\|_{L^2(0,1;H^{-1}(\Omega))} &= \sup_{\substack{\phi \in L^2(0,1;H^1(\Omega)) \\ \|\phi\|=1}} \left|\int_0^1 \int_\Omega \partial_t \rho_n \phi \, \mathrm{d}x \, \mathrm{d}t\right| \\ &\leq \|m_n\|_{L^2(Q)} \|\nabla \phi\|_{L^2(Q)} \leq \|m_n\|_{L^2(Q)} \leq C,\end{aligned}$$

$$\begin{aligned}\|\mathrm{div}(m)\|_{L^2(0,1;H^{-1}(\Omega))} &= \sup_{\substack{\phi \in L^2(0,1;H^1(\Omega)) \\ \|\phi\|=1}} \left|\int_0^1 \int_\Omega \mathrm{div}(m_n)\phi \, \mathrm{d}x \, \mathrm{d}t\right| \\ &\leq \|m_n\|_{L^2(Q)} \|\nabla \phi\|_{L^2(Q)} \leq \|m_n\|_{L^2(Q)} \leq C.\end{aligned}$$

Hence all the variables in (3.19) are bounded. Therefore, we can conclude that, up to the extraction of a subsequence, there exists a $(\rho, m, v) \in U$ such that

$$\rho_n \rightharpoonup \rho \text{ in } L^\infty(Q), \qquad m_n \rightharpoonup m \text{ in } L^2(Q)^d, \qquad v_n \rightharpoonup v \text{ in } L^2(0,1;H^1(\Omega))^d.$$

3) We will now show that the functional working on our weak limit gives us the minimum. Therefore, we need to show that (3.19) is weakly lower semi-continuous. Moreover, we need to show that $w_n\sqrt{\rho_n} \rightharpoonup w\sqrt{\rho}$ in $L^2(Q)^d$ and $\rho_n v_n \rightharpoonup \rho v$ in $L^2(Q)^d$.

The energy functional $J(\rho_n, m_n)$ is weakly lower semi-continuous for the convergence in $L^2(Q) \times L^2(Q)^d$ [5]. The regularisation term is the L^2-norm of $(\nabla v_n + (\nabla v_n)^T)/2$, which is bounded. Since the L^2-norm is always lower semi-continuous, we have weak lower semi-continuity for the convergence of $(\nabla v_n + (\nabla v_n)^T)/2$ in $L^2(Q)^{d\times d}$. For the coupling term, we can use the same reasoning, therefore we also have weak lower semi-continuity for the coupling term in $L^2(Q)^d$.

It now remains to prove weak convergence of $w_n\sqrt{\rho_n}$ to $w\sqrt{\rho}$ and $\rho_n v_n$ to ρv in $L^2(Q)^d$. For this, we make use of the Aubin-Lions lemma [31], which gives us the following result:

Because $L^\infty(\Omega) \subset\subset L^2(\Omega)$ and $L^2(\Omega) \hookrightarrow H^{-1}(\Omega)$ [18] and we have $\partial_t \rho_n \in L^2(0,1;H^{-1}(\Omega))$, $\rho_n \in L^\infty(0,1;L^\infty(\Omega))$, by the Aubin-Lions lemma, we know $\rho_n \to \rho$ in $L^\infty(0,1;L^2(\Omega))$.

Next we calculate

$$\begin{aligned}
\lim_{n\to\infty} \|\rho_n v_n - \rho v\|_{L^1(Q)^d} &= \lim_{n\to\infty} \int_0^1 \int_\Omega |v_n\rho_n - v\rho| \,\mathrm{d}x\,\mathrm{d}t \\
&\le \lim_{n\to\infty} \int_0^1 \int_\Omega |v_n(\rho_n - \rho)| \,\mathrm{d}x\,\mathrm{d}t + \lim_{n\to\infty} \int_0^1 \int_\Omega |(v_n - v)\rho| \,\mathrm{d}x\,\mathrm{d}t \\
&\le \lim_{n\to\infty} \left(\|v_n\|_{L^2(Q)} \|\rho_n - \rho\|_{L^2(Q)} \right) + 0 \\
&\le C \cdot 0 + 0 = 0,
\end{aligned}$$

where we made use of the weak convergence of v_n, the boundedness of v_n and the strong convergence of ρ_n. Now since $\rho_n v_n \to \rho v$ in $L^1(Q)^d$ and $\rho_n v_n$ has some weak limit in $L^2(Q)^d$, it follows that the limits are the same and therefore $\rho_n v_n \rightharpoonup \rho v$ in $L^2(Q)^d$.

Noting that $\sqrt{\rho_n} \in L^2(Q)$, we can use the same reasoning for $w_n\sqrt{\rho_n}$ and find that $w_n\sqrt{\rho_n} \rightharpoonup w\sqrt{\rho}$ in $L^2(Q)^d$.

□

Remark 1. *For minimisation problems with other regularisers, we can make a similar proof. In fact, for some cases, we can even define a more general space in which v should belong:*

- *For $\mathcal{R}(v) = \|\nabla v\|_{L^2(Q)^d}$, it has been shown that the space $BL_1^2(\Omega) = \{u \in \mathcal{D}'(\Omega) \mid D^\alpha u \in L^2(\Omega) \quad \forall |\alpha| = 1\}$ as part of the Bochner space $L^2(0,1;BL_1^2(\Omega))$, is restrictive enough. Here $\mathcal{D}'(\Omega)$ denotes the space of distributions, which is the dual of the test-function space $\mathcal{D}(\Omega)$.*
- *For $\mathcal{R}_D(v) = \|div(v)\|_{L^2(Q)}$, the space $W^{div,2}(\Omega) := \{v \in L^2(\Omega)^2 \mid div(v) \in L^2(\Omega)\}$ as part of the Bochner space $L^2(0,1;W^{div,2}(\Omega))$, is restrictive enough.*
- *For $\mathcal{R}_A(v) = \frac{1}{2}\sum_{i=1}^2 \|U\nabla v_i\|_2^2$, we need the same space as in our proof, $L^2(0,1;H^1(\Omega))^d$.*

4 Segmentation

For our application of optimal transport to vascular structures, we do not expect flow everywhere. We have parts of the image where density can flow through, such as a channel or blood vessel and parts of the image which can be considered as background: areas where no or not much flow is expected. If we would know the segmentation between channels and background, or between different parts of the image, we could better give an estimate of the optimal transport.

A segmentation can be made based on image intensity (density), based on the flow or a combination of the two. In many data sets, the edges of the flow coincide with the edges in image intensity. See for instance the examples of haemoglobin microcirculation and the flow in a human brain in section 2.2.

First we will give a short overview on well known methods for segmentation. Next we will give some ideas on how we can incorporate a segmentation in our optimal transport models.

4.1 Overview of well known methods

Around the change of the century, some of the now best known segmentation methods have been developed. These methods were developed for static images. Therefore, they aim to give a segmentation based on image intensity and do not take any flow into account. It is important to see that different methods have different goals. Roughly, we can divide them into two groups [4]:

- Methods that try to find a new image that resembles the original image. The new image consists of several regions with (almost) constant intensity. Examples of these methods are the Mumford-Shah method [32] and the Chan-Vese method [14], which can be seen as a special case of the Mumford-Shah method.
- Methods that try to find a boundary that separates two or more regions or objects. Geodesic active contours and level set methods are of this kind.

The model introduced by Mumford and Shah [32] is the following: Let Ω be a bounded open set, let $0 \leq f(x) \leq 1$ be the image intensity for $x \in \Omega$ a.e. Now we want to minimize

$$\min_{u,\Gamma} \int_{\Omega\backslash\Gamma} (u(x) - f(x))^2 \,\mathrm{d}x + \alpha \int_{\Omega\backslash\Gamma} |\nabla u|^2 \,\mathrm{d}x + \beta \oint_{\Gamma} \mathrm{d}s, \tag{4.1}$$

where we search for a new image in the Sobolev space $u \in W^{1,2}(\Omega\backslash\Gamma)$. The (smooth) contour is represented by Γ and parametrised by s. α and β are nonnegative parameters which represent the importance of the function u to be piecewise constant and the boundary Γ to be small, respectively.

Ambrosio and Tortorelli [1] proposed to not treat the boundary Γ in (4.1) in an explicit way, but to define a function z which (lack of) strength represents the presence of a boundary. With their method, one minimises the following sequence of functionals for $\varepsilon \to 0$ where $0 < \varepsilon \ll 1$:

$$\min_{u,z} \int_{\Omega} (u(x) - f(x))^2 \,\mathrm{d}x + \int_{\Omega} z(x)^2 |\nabla u|^2 \,\mathrm{d}x + \int_{\Omega} \left(\varepsilon |\nabla z|^2 + \frac{1}{4\varepsilon}(z(x) - 1)^2 \right) \mathrm{d}x. \tag{4.2}$$

First of all, we see that we want to stay close to the original image $f(x)$. Since $\varepsilon \ll 1$, the most right part of (4.2) makes sure that $z(x) \approx 1$ in most part of the image. Therefore, the second term of (4.2) makes sure that $u(x)$ is almost constant in each part of the image, except around the boundary that we try to find. Finally, the third term makes sure that $z(x)$ is 'smooth enough'. In [1] it is shown that for $\varepsilon \to 0$, we obtain the Mumford-Shah functional.

The Chan-Vese method [14] is in fact a special case of the Mumford-Shah method, where we try to find two regions of constant intensity μ_1 and μ_2 as a new segmented image. We minimize

$$\min_{\mathcal{X},\mu1,\mu2} \int_\Omega \mathcal{X}(x)(\mu_1 - f(x))^2 \,\mathrm{d}x + \int_\Omega (1-\mathcal{X}(x))(\mu_2 - f(x))^2 \,\mathrm{d}x + \beta \mathrm{Per}(\mathcal{X}) + \gamma \int_\Omega \mathcal{X}(x)\,\mathrm{d}x, \quad (4.3)$$

where $\mathcal{X}(x) \in \{0,1\}$ for all x and $\mathrm{Per}(\mathcal{X})$ is the notion of the perimeter of $\mathcal{X}$ (or the length of the boundary). Again $\beta \geq 0$ is used to define the importance for the length of the boundary to be small. Now we also have an option to set a penalty on the area within the contour, by choosing $\gamma \geq 0$. How to choose β and γ obviously depends on the image. When we think of our vascularity examples, and we would like to segment vessels from the background, it makes sense to not penalize the length of the boundary too much, since it is expected to be very large.

In the second class of segmentation methods, one is looking for a boundary, without specifying what we see inside the different regions. By modelling the boundary by geodesic active contours, one begins with an initial guess for the boundary, which will propagate over time as there are forces that push and pull the boundary. Because of this flexible movement, geodesic active contours are often called snakes. Kass *et al.* [24] proposed to minimise an energy, which takes the derivative, as well as the curvature of the contour Γ into account. The main idea is to fit the contour to points in the image where the gradient of the intensity is largest.

$$\min_\Gamma \alpha \int_0^1 |\Gamma'(s)|^2 \,\mathrm{d}s + \int_0^1 |\Gamma''(s)|^2 \,\mathrm{d}s - \int_0^1 |\nabla f(\Gamma(s))| \,\mathrm{d}s, \quad (4.4)$$

where s is the parametrisation of Γ.

Caselles *et al.* [10] showed that the curvature term in (4.4) is not necessary, since curvature is already implicitly minimised by incorporating the derivative of the contour. The minimisation of energy functionals like (4.4) is often done by iteratively minimising the separate parts that have an influence on the total energy.

The idea behind the level-set method is the following: we regard the contour that we would like to get, as the zero level-set of a function $\phi : \Omega \mapsto \mathbb{R}$. If we let the function $\phi(x,t)$ propagate in time with a speed function that depends on the gradient of the image intensity $|\nabla f(x)|$, we see that the zero level-set of this function also propagates. By choosing the speed function wisely, we converge to a function where $\phi(x,t) = 0$ gives us the desired segmentation. A more detailed explanation can be found in [4].

4.2 Segmentation in optimal transport models

We have now shortly described a number of segmentation methods that can be used to make segmentations in images. In the following we will assume that we have a segmentation a priori available, and want to use this segmentation in order to get an improved good solution to our optimal transport problem.

4.2.1 Binary segmentation

First we will describe how we can make use of a binary segmentation. We assume that our domain Ω has a subset S, which is not necessarily connected. We only allow for transport in the subset S, so transport in $\Omega\backslash S$ is prohibited. If we think about it carefully, we see that this is actually exactly the same minimisation problem as (3.9), where we replace our domain Ω by our new subset S. Since we do not allow transport in $\Omega\backslash S$, we can impose the extra boundary condition $m \cdot n|_{\partial S \backslash \partial\Omega} = 0$ on the part that is not on the boundary of our original domain Ω.

This segmentation can also be used when we expect very different transports in different segments $S_1, S_2, \ldots, S_n$ of the image, but when it is not expected that density is transported from one segment to the other. As an example, think of several channels parallel to each other, in which density is transported. We simply calculate the optimal transport for each segment separately and combine our solutions.

4.2.2 Weak segmentation

When we do not have a precise description of the boundary of a segment, but do have a rough idea where the transport is located, we might want to use a weak segmentation. That is, we can replace our anisotropy matrix A in (3.9) by a scalar times the identity matrix, and we obtain for J:

$$J_\beta(\rho, m) = \begin{cases} \frac{\beta|m|^2}{2\rho} & \text{if } \rho > 0, \\ 0 & \text{if } (\rho, m) = (0,0), \\ \infty & \text{otherwise,} \end{cases} \tag{4.5}$$

where β is a function that depends on x. This means that the transport is more likely to take place on locations where $\beta(x)$ is small. Taking $\beta(x) \to \infty$, we obtain a binary segmentation for J. Note that if we would like to get the same result as with a binary segmentation for the complete functional, we also need to incorporate $\beta(x)$ in $K(\rho, m, v)$, $R(v)$ and ι_{MC}.

4.2.3 Anisotropy as segmentation

In section 4.2.2, we obtained a weak segmentation by replacing the matrix A in (3.9) by a scalar. In the same spirit, we could try to construct an anisotropy matrix A such that we obtain a segmentation that is implicitly contained in A.

Let us recall the coherence enhancing diffusion tensor D_{CE} from section 3.2.2. We can use the properties of this diffusion tensor to create a desired anisotropy matrix. We write

$$D_{CE} = \begin{pmatrix} d_{11} & d_{12} \\ d_{12} & d_{22} \end{pmatrix},$$

where D_{CE} is known to be symmetric [44]. Now we construct the anisotropy matrix

$$A_{CE} = \begin{pmatrix} 1 + \gamma(1 - d_{11}) & -\gamma d_{12} \\ -\gamma d_{12} & 1 + \gamma(1 - d_{22}) \end{pmatrix}, \tag{4.6}$$

where γ is a parameter which scales the 'strength' of the anisotropy. The matrix A_{CE} is defined like this because it has some nice properties: assume that we have a momentum m, which is aligned with one of the eigenvalues (w_1 or w_2) of D_{CE}. Without loss of generality, we assume m has length 1, so it is exactly the same as one of the eigenvectors. If we calculate the values for $J_{A_{CE}}(\rho, w_i)$ of (3.7) and compare it to $J(\rho, w_i)$ of (3.6), we get

$$\begin{aligned} J_{A_{CE}}(\rho, w_1) &= (1 + \gamma(1 - \alpha))\frac{|w_1|^2}{2\rho} = \big(1 + \gamma(1 - \alpha)\big)J(\rho, w_1), \\ J_{A_{CE}}(\rho, w_2) &= \big(1 + \gamma(1 - \lambda_2)\big)\frac{|w_2|^2}{2\rho} = \big(1 + \gamma(1 - \lambda_2)\big)J(\rho, w_2), \end{aligned}$$

where α was a very small parameter, as chosen in (3.17), and λ_2 was the eigenvalue corresponding to direction w_2 along the structure edge. This shows that for a momentum in the direction of the edge (w_1), we get a cost that is approximately $(1 + \gamma)$ as high as the isotropic cost. For a momentum in the direction along the edge (w_2), we get a cost that is $\big(1 + \gamma(1 - \lambda_2)\big)$ as high as the isotropic cost. When a structure is clearly visible, we have $\lambda_2 \approx 1$ and we thus get a cost that is approximately the same as an isotropic cost. However, when a structure is not clearly visible, the cost in this direction will also be higher.

Using this anisotropy matrix, we do not only get that the transport in one direction is preferred above the other direction, but also that moving through parts of the domain where there is a clear vascular structure gives us a much lower cost than moving through parts of the domain where no vascular structure is visible. This is very preferable in the first two examples given in chapter 2: when we do not see vascularity, we are probably in the background, and we do not expect transport, and thus no momentum. When we do see vascularity, we expect a transport, and it is therefore wise to reduce the cost in $J_{A_{ce}}$.

5 Joint segmentation and optimal transport

Since a segmentation is not always known beforehand, we would benefit from a model in which the segmentation is carried out at the same time as the optimal transport. In this chapter we propose a joint segmentation and optimal transport model.

As we have seen in section 2.2, many data sets contain images with only two segments: in the first segment the flow is very different from the flow in the second segment, which might not contain flow at all. Often, the movement within each segment is rather smooth, while it changes sharply on the boundary of the segments. We describe a joint model for these kind of images. As explained in chapter 4, a segmentation can be made on basis of a combination of properties, here we choose to make a flow-driven segmentation, without taking the density into account.

We combine model 2 (3.9) with a Mumford-Shah like functional (section 4.1) to obtain a joint model. The minimisation problem reads

$$\begin{aligned}\min_{(\rho,m,v_{in},v_{out},\mathcal{X})} \int_0^1 \int_\Omega & J_A(\rho,m) + \iota_{MC}(\rho,m) + \mu R(v_{in}) + \mu R(v_{out}) \\ & + \lambda \mathcal{X} K(\rho,m,v_{in}) + \lambda(1-\mathcal{X})K(\rho,m,v_{out})\,\mathrm{d}x + \beta \mathrm{Per}(\mathcal{X})\,\mathrm{d}t \qquad (5.1)\end{aligned}$$

$$\text{where } J_A(\rho,m) = \begin{cases} \frac{m^TAm}{2\rho} & \text{if } \rho>0, \\ 0 & \text{if } (\rho,m)=(0,0), \\ \infty & \text{otherwise.}\end{cases}$$

$$MC = \{(\rho,m) \mid \partial_t\rho + \mathrm{div}(m) = 0,\ \rho(\cdot,0)=\rho_0,\ \rho(\cdot,1)=\rho_1,\ m\cdot n|_{\partial\Omega}=0,\ \rho\geq 0\},$$

$$K(\rho,m,v) = \frac{1}{2}|m-\rho v|^2,$$

and $R(v)$ some convex regularisation functional.

where $\mathcal{X}(x) \in \{0,1\}$ and $\mathrm{Per}(\mathcal{X})$ is the perimeter of $\mathcal{X}$. Just as in model 2, we now model the optimal transport by minimising the energy functional $J_A(\rho,m)$ and projecting on the mass conservation set MC. The difference from model 2 is that we now do not try to find a single velocity v, but instead try to find two different velocities, v_{in} and v_{out}. Each velocity exists on the complete domain Ω and will be regularised on the full domain. However, the coupling between ρ, m and $v_{in/out}$ is only made whenever $\mathcal{X}$ is zero or one respectively. This means that we will obtain two smooth velocity fields that are both only close to m and ρ within each segment. The desired velocity field is a combination of v_{in} and v_{out}:

$$v = \mathcal{X}v_{in} + (1-\mathcal{X})v_{out}.$$

The minimisation problem (5.1) is highly non-convex: not only do we have the non-convexity that was already present in model 2, we also have to deal with the non-convexity of the binary segmentation function $\mathcal{X}$. A solution to the latter problem is the convexification of the functions $\mathcal{X}$ and $(1-\mathcal{X})$ to the convex functions z^2 and $(1-z)^2$ respectively [13, 7, 19], where $z \in [0,1]$. With this convexification, minimisation problem (5.1) changes to

$$\begin{aligned}\min_{(\rho,m,v_{in},v_{out},z)} \int_0^1 \int_\Omega & J_A(\rho,m) + \iota_{MC}(\rho,m) + \mu R(v_{in}) + \mu R(v_{out}) \\ & + \lambda z^2 K(\rho,m,v_{in}) + \lambda(1-z)^2K(\rho,m,v_{out}) + \beta TV(z)\,\mathrm{d}x\,\mathrm{d}t, \qquad (5.2)\end{aligned}$$

where we used the total variation (TV) of z to measure the length of the segmentation boundary, since we are now working with a continuous function instead of a binary one. The final segmentation is obtained by thresholding at $z=\frac{1}{2}$:

$$v(x) = \begin{cases} v_{in}(x) & \text{if } z(x) \geq \frac{1}{2}, \\ v_{out}(x) & \text{if } z(x) < \frac{1}{2}, \end{cases}$$

or equivalently

$$v = [z]v_{in} + (1 - [z])v_{out}.$$

Because of the non-convexity of this joint model, it is not clear if we can find a minimum and solve (5.2) properly.

6 Numerical Framework

In chapters 3 and 5, we have derived the following models:

- Model 1: (an)isotropic optimal transport without regulariser.
- Model 2: (an)isotropic optimal transport with regulariser;
  a. regularising the L^2-deviation from divergence-free constraint;
  b. regularising the L^2-deviation from rigidity constraint;
  c. L^2 anisotropic regulariser in v_1 and v_2.
- Joint segmentation-optimal transport model.

These models will be implemented in a Modified Primal-Dual Hybrid Gradient algorithm (PDHGM) [12], which is one algorithm among a class of primal-dual algorithms. In order to comprehend what happens in these algorithms, we need to understand that up to now, we have been looking at the primal minimisation problem. With the use of Fenchel duality [39], we can rewrite the problem to its dual maximisation problem or its primal-dual minmax problem [17]. Primal-dual algorithms are designed to solve the latter, which are very common in variational methods. For more information about the connection between different primal-dual algorithms which make use of the duality of minimisation problems, we refer to [17].

A key tool for the use of primal-dual algorithms is the so called proximal splitting method [37]. The idea of the proximal splitting method is very similar to the gradient descent method, which is often used for solving smooth convex minimisation problems. In the gradient descent method, small steps are taken iteratively. These steps are in the direction where the directional derivative of the functional to be minimised is largest; doing so leads to the global minimum, since we have convexity. When the minimisation problem is non-smooth, one could make use of the proximal splitting method. Similar to gradient descent, small steps are taken iteratively to find a minimum. However, instead of using the gradient to search for the direction in which the descent is largest, the algorithm is looking for a new point that gives a small value for the functional, but is also very close to the current point. The proximal operator for a scaled functional $\tau F(x)$ is defined as

$$\mathrm{Prox}_{\tau F}(x) = \underset{y}{\mathrm{argmin}} \left\{ \frac{1}{2} \|x - y\|_2^2 + \tau F(y) \right\}. \tag{6.1}$$

We see that (6.1) is searching for a small value of $\tau F(y)$, while keeping y close to x. The right hand side consists of a strongly convex quadratic part and a convex functional $\tau F(y)$. Therefore, the right hand side of (6.1) is always strongly convex and it thus has a unique solution. Unfortunately, there is no general way of expressing the proximal operator explicitly. Sometimes an explicit expression for the proximal operator does not even exist. This is because there is no general way of solving this minimisation problem, without knowing more details about the structure of $F(x)$. But, if we assume $F(x)$ is strictly convex, C^2 in the whole domain and all the second derivatives of $F(x)$ are bounded, we can make an approximation, which is the same as one step in the gradient descent method:

$$\mathrm{Prox}_{\tau F}(x) \approx x - \tau \nabla F(x).$$

In chapter 6.3, we will see that not only the proximal operators for our functionals defined in chapter 3 are needed, but also the proximal operators of their complex conjugates. Making use of Moreau's decomposition [37], it can be seen that these are calculated easily if the 'normal' proximal operators are known:

$$\mathrm{Prox}_{\tau F^*}(x) = x - \tau \mathrm{Prox}_{F/\tau} \left(\frac{x}{\tau} \right). \tag{6.2}$$

We will find explicit solutions for all our functionals in section 6.1, which enables us to implement an efficient algorithm for our problem.

In the remainder of this chapter, we will first derive the proximal operators for models 1 and 2. Next, the operators needed in the regularisation functional of model 2 and their adjoints will be defined. Then, we will give the implementation of our two models in PDHGM, along with its discretisation scheme. After that, we will show the numerical convergence of our algorithms. Finally, the numerical framework for our joint model will be given.

6.1 Proximal operators for models 1 & 2

In order to implement our models in PDHGM, proximal operators for our functionals have to be derived. We will begin with the derivation of the proximal operators needed for model 1. After this, we will derive the proximal operators of model 2 that differ from the already defined operators.

6.1.1 Proximal operators for model 1

We begin by looking at the proximal operator of the indicator function ι_{MC}. We obtain the projection operator on the set MC in closed form [36]:

$$\begin{aligned}\operatorname{Prox}_{\tau\iota_{MC}}(\tilde{\rho},\tilde{m}) &= \operatorname{Proj}_{MC}(\tilde{\rho},\tilde{m})\\ &= (\operatorname{Id} - B^*(BB^*)^{-1}B)(\tilde{\rho},\tilde{m}) + B^*(BB^*)^{-1}(0,b_0),\\ \text{where } B(\tilde{\rho},\tilde{m}) &= \big(\partial_t\tilde{\rho} + \operatorname{div}(\tilde{m}), b(\tilde{\rho},\tilde{m})\big).\end{aligned} \tag{6.3}$$

Here b is the operator which takes the boundary conditions from our space-time domain $\Omega \times [0,1]$. $b_0 = (\rho_0, \rho_1, m(\partial\Omega)) = (\rho_0, \rho_1, 0)$ in case inflow or outflow is prohibited. In case inflow or outflow is allowed, we cannot simply omit the term $m(\partial\Omega) = 0$ without extending or altering the rest of the projection step: only $\partial_t\rho + \operatorname{div}(m) = 0$ is not enough to guarantee that the total density $\int_\Omega \rho(x,t)\,\mathrm{d}x$ stays the same at every time. In practice, it is easiest to simply scale the result after our projection step (6.3). This scaling can also be done when $\rho_0 \neq \rho_1$; we then assume the inflow or outflow is linear in time between the initial and final density. Using this extra operation after the projection step, we obtain:

$$\begin{cases}(\hat{\rho},\hat{m}) &= \operatorname{Proj}_{MC}(\tilde{\rho},\tilde{m}),\\ (\rho(\cdot,t), m(\cdot,t)) &= \left(\frac{\int_\Omega (1-t)\rho_0 + t\rho_1\,\mathrm{d}x}{\int_\Omega (\hat{\rho}(x,t))\,\mathrm{d}x}\right)(\hat{\rho}(\cdot,t), \hat{m}(\cdot,t)).\end{cases} \tag{6.4}$$

Note that after this scaling operation, it is not guaranteed to have $\partial_t\rho + \operatorname{div}(m) = 0$, since the scaling is different for every t. In practice, when the time-discretisation is chosen fine enough and when the step size τ of the algorithm is chosen small enough, this does not give any problems.

The projection operator (6.3) consists of applying forward operators B and B^*, along with the inverse operator $(BB^*)^{-1}$. We will now elaborate on how to apply this inverse operator. We first write $u = (m,\rho)$ as one couple for the two variables. We obtain the shorter notation $\operatorname{div}(u) = \partial_t\rho + \operatorname{div}(m)$, where the first divergence is taken in the space $Q = \Omega \times [0,1]$. For the sake of clarity, we will not bother about the second part of the operator B now, and simply write $Bu = \operatorname{div}(u)$.

Projection when inflow/outflow is prohibited

In case inflow or outflow is prohibited, we have the following adjoint:

$$\begin{aligned}&\text{for } g \in G_1 := \{\hat{g} \in (W^{1,2}(Q))^2 |\ \hat{g}\cdot n = 0 \text{ on } \partial Q\} \quad \text{and} \quad h \in W^{1,2}(Q),\\ \langle Bg, h\rangle &= \int_Q \operatorname{div}(g)h\,\mathrm{d}Q = -\int_Q g\cdot(\nabla h)\,\mathrm{d}Q + \int_{\partial Q} h(g\cdot n)\,\mathrm{d}s\\ &= -\int_Q g\cdot(\nabla h)\,\mathrm{d}Q = \langle g, B^*h\rangle.\end{aligned}$$

From which it follows that $B = \mathrm{div}$ and $B^* = \nabla$. We rewrite (6.3) to

$$\begin{aligned} \mathrm{Proj}_{MC}(u) &= u - B^*y + B^*(BB^*)^{-1}(0, b_0), \\ \text{where } y &= (BB^*)^{-1}Bu. \end{aligned} \tag{6.5}$$

The last line of (6.5) rewrites to

$$BB^*y = Bu. \tag{6.6}$$

In order to apply the operator B on $B^*y = \nabla y$, we need $\nabla y \in G_1$, which means $\frac{\partial y}{\partial n} = 0$ on ∂Q. This means that for applying $(BB^*)^{-1}$, we need to solve the following Poisson equation for y:

$$\begin{cases} \Delta y &= \mathrm{div}(u) \text{ in } Q \\ \frac{\partial y}{\partial n} &= 0 \text{ on } \partial Q. \end{cases} \tag{6.7}$$

On a rectangular domain, this Poisson equation can be solved by using the discrete cosine transform [41]. It should be noted that since B works on functions in the space G_1, there is a restriction on what u can look like: on the right side of (6.6), B is applied on u, which means that $u \cdot n|_{\partial Q} = 0$, which means that inflow or outflow is prohibited. Because of this, different boundary conditions are needed for the case in which inflow or outflow is allowed.

Projection when inflow/outflow is allowed

In case inflow or outflow is allowed, we follow a similar reasoning to obtain the boundary conditions for the Poisson equation:

$$\begin{aligned} &\text{for } g \in G_2 := \{\hat{g} \in (W^{1,2}(Q))^2 |\ \hat{g} \cdot n = 0 \text{ on } \Omega \times 0 \cup \Omega \times 1\} \\ &\text{and } h \in H_2 := \{\hat{h} \in W^{1,2}(Q) |\ \hat{h} = 0 \text{ on } \partial\Omega \times [0,1]\}, \\ \langle Bg, h \rangle &= \int_Q \mathrm{div}(g) h \,\mathrm{d}Q = -\int_Q g \cdot (\nabla h) \,\mathrm{d}Q + \int_{\partial Q} h(g \cdot n) \,\mathrm{d}s \\ &= -\int_Q g \cdot (\nabla h) \,\mathrm{d}Q = \langle g, B^*h \rangle, \end{aligned}$$

so now also $B = \mathrm{div}$ and $B^* = \nabla$. Again we will solve

$$BB^*y = Bu. \tag{6.8}$$

First the operator B^* is applied on y, which means $y = 0$ on $\partial\Omega \times [0,1]$, since $y \in H_2$. Furthermore, in order to apply the operator B on $B^*y = \nabla y$, we need $\nabla y \in G_2$, which means $\frac{\partial y}{\partial n} = 0$ on $\Omega \times 0 \cup \Omega \times 1$. The Poisson equation to be solved for y now becomes

$$\begin{cases} \Delta y &= \mathrm{div}(u) \text{ in } Q \\ \frac{\partial y}{\partial n} &= 0 \text{ on } \Omega \times 0 \cup \Omega \times 1, \\ y &= 0 \text{ on } \partial\Omega \times [0,1]. \end{cases} \tag{6.9}$$

In a rectangular domain, this Poisson equation can be solved by using a combination of discrete sine transforms in the space dimensions and a discrete cosine transform in the time dimension [41].

In [36], the following closed form for the proximal operator of $J(\rho, m)$ has been derived:

$$\mathrm{Prox}_{\tau J}(\tilde{\rho}, \tilde{m}) = \begin{cases} (\rho, \frac{\rho \tilde{m}}{\rho + \tau}) & \text{if } \rho > 0, \\ (0,0) & \text{else,} \end{cases} \tag{6.10}$$

where ρ is the largest real root of the third order polynomial

$$P(\hat{\rho}) = \hat{\rho}^3 + (2\tau - \tilde{\rho})\hat{\rho}^2 + (\tau^2 - 2\tilde{\rho}\tau)\hat{\rho} - \tau\Big(\frac{|\tilde{m}|^2}{2} + \tau\tilde{\rho}\Big). \tag{6.11}$$

It is easily seen that $(\rho, m) = (0, 0)$ is always a solution to (6.10), so the constraint $\rho \geq 0$ is always satisfied.

For the anisotropic variant of model 1, we only have to derive the proximal operator for J_A, since the projection on the mass conservation set stays the same. Similarly to [36], a closed form for $\mathrm{Prox}_{\tau J_A}$ has been derived in [21]:

$$\begin{gathered}\mathrm{Prox}_{\tau J_A}(\tilde{\rho}, \tilde{m}) = \begin{cases} \big(\rho, f(\rho)\big) & \text{if } \rho > 0, \\ (0, 0) & \text{else,} \end{cases} \\ \text{where } f(\rho) = \big(\rho\,\mathrm{Id}_2 + \frac{\tau}{2}(A + A^T)\big)^{-1}\rho\tilde{m}, \end{gathered} \tag{6.12}$$

and where ρ is the largest real solution to the fifth order polynomial

$$P(\hat{\rho}) = (\hat{\rho} - \tilde{\rho}) - \frac{\tau}{2}\tilde{m}^T(\hat{\rho}\,\mathrm{Id}_2 + \frac{\tau}{2}(A + A^T))^{-1}A(\hat{\rho}\,\mathrm{Id}_2 + \frac{\tau}{2}(A + A^T))^{-1}\tilde{m}. \tag{6.13}$$

6.1.2 Operators for model 2: Optimal transport with regulariser

In section 3.2, we introduced a coupling term $K(\rho, m, v)$ when using regularisers. Because the complete minimisation problem is not convex in all variables, the minimisation problem is split up in order to obtain a solution. We can perform a block coordinate descent and minimize each convex problem alternatively [43]. Using this method, we will find a local minimum of the joint problem [21]. This gives us the alternating minimisation method, also known as the Gauss-Seidel method [2]:

$$\begin{cases} (\rho^{k+1}, m^{k+1}) & = \underset{(\rho,m)}{\operatorname{argmin}} \left\{\mathcal{J}(\rho, m) + \lambda\mathcal{K}(\rho, m, v^k)\right\}, \\ v^{k+1} & = \underset{v}{\operatorname{argmin}} \left\{\lambda\mathcal{K}(\rho^{k+1}, m^{k+1}, v) + \mu\mathcal{R}(v)\right\}. \end{cases} \tag{6.14}$$

For readability, we will from now on write $K(\rho, m)$ and $K(v)$ for the coupling term, since v^k, respectively (ρ^{k+1}, m^{k+1}) are fixed, within each minimisation problem. For the first minimisation, we will define a joint proximal operator for J or J_A, and K. Following the same reasoning as in [36] and [21], we can write the proximal operator for $(J + \lambda K)$ as an explicit pointwise expression, where we have to solve a third order polynomial.

Proof. Let us define

$$(\rho, m) = \mathrm{Prox}_{\tau(J+K)}(\tilde{\rho}, \tilde{m}) = \underset{(\hat{\rho},\hat{m})}{\operatorname{argmin}} \left\{\frac{1}{2}\left\|(\tilde{\rho}, \tilde{m}) - (\hat{\rho}, \hat{m})\right\|_2^2 + \tau\big(J(\hat{\rho}, \hat{m}) + \lambda K(\hat{\rho}, \hat{m})\big)\right\}.$$

If $\rho = 0$, we immediately see that $m = 0$, since $m \neq 0$ would lead to a value of infinity. If $\rho > 0$, we have that $(J + \lambda K)$ is C^1 and strictly convex. Therefore we know

$$\begin{aligned} 0 &= \nabla\left\{\frac{1}{2}\left\|(\tilde{\rho}, \tilde{m}) - (\rho, m)\right\|_2^2 + \tau\big(J(\rho, m) + \lambda K(\rho, m)\big)\right\} \\ &= (\rho, m) - (\tilde{\rho}, \tilde{m}) + \tau\nabla J(\rho, m) + \tau\lambda\nabla K(\rho, m). \end{aligned}$$

Working out these equations, leads to the following pointwise expression:

$$\mathrm{Prox}_{\tau(J+K)}(\tilde{\rho},\tilde{m}) = \begin{cases} \left(\rho, \frac{\rho\tilde{m}+2\lambda\tau\rho^2 v}{\rho+\tau+2\lambda\tau\rho}\right) & \text{if } \rho > 0, \\ (0,0) & \text{else,} \end{cases} \tag{6.15}$$

where ρ is the largest real solution to the third order polynomial

$$\begin{aligned} P(\hat{\rho}) &= a\hat{\rho}^3 + b\hat{\rho}^2 + c\hat{\rho} + d, \\ \text{where } a &= 1 + 4\tau\lambda\big(1 + \frac{1}{2}|v|^2\big) + 4\tau^2\lambda^2\big(1+|v|^2\big), \\ b &= 2\tau - \tilde{\rho} + 4\tau^2\lambda - 4\tau\lambda\tilde{\rho} - 4\tau^2\lambda^2\tilde{\rho} - 2\tau\lambda\tilde{m}^T v - 4\tau^2\lambda^2\tilde{m}^T v + 4\tau^2\lambda|v|^2 + 2\tau^3\lambda^2|v|^2, \\ c &= \tau\big(\tau - 2\tilde{\rho} - 4\tau\lambda(\tilde{\rho} + \tilde{m}^T v) + 2\tau^2\lambda|v|^2\big), \\ d &= -\frac{\tau}{2}\big(|\tilde{m}|^2 + 2\tau\tilde{\rho}\big). \end{aligned} \tag{6.16}$$

□

We see that for $\lambda = 0$, (6.15) and (6.16) reduce to (6.10) and (6.11) respectively.

The proximal operator for $J_A + \lambda K$ is derived analogously and writes

$$\begin{aligned} \mathrm{Prox}_{\tau(J_A+K)}(\tilde{\rho},\tilde{m}) &= \begin{cases} \big(\rho, f(\rho)\big) & \text{if } \rho > 0, \\ (0,0) & \text{else,} \end{cases} \\ \text{where } f(\rho) &= \big((\rho + 2\lambda\tau\rho)\,\mathrm{Id}_2 + \frac{\tau}{2}(A + A^T)\big)^{-1}(\rho\tilde{m} + 2\lambda\tau\rho^2 v) \end{aligned} \tag{6.17}$$

and where ρ is the largest real solution to the fifth order polynomial

$$\begin{aligned} P(\hat{\rho}) &= 2\hat{\rho}(\tilde{\rho} - \hat{\rho}) + \tau z^T B^{-1}(\hat{\rho}A)B^{-1}z + 2\lambda\tau v^T B^{-1} z - 2\lambda\tau|v|^2, \\ \text{where } B &= \big((\hat{\rho} + 2\lambda\tau\hat{\rho})\mathrm{Id}_2 + \frac{\tau}{2}(A + A^T)\big), \\ \text{and } z &= (\tilde{m} + 2\lambda\tau\hat{\rho}v). \end{aligned} \tag{6.18}$$

For the second minimisation of (6.14), we still need to define the proximal operators. We can write the proximal operator for $\lambda K(v) = \frac{\lambda}{2}|m - \rho v|^2$ explicitly: we define

$$v = \mathrm{Prox}_{\tau K}(\tilde{v}) = \underset{\hat{v}}{\mathrm{argmin}}\left\{\frac{1}{2}\left\|\tilde{v} - \hat{v}\right\|_2^2 + \tau\lambda K(\hat{v})\right\}.$$

Since $K(v)$ is strictly convex (for $\rho > 0$) and C^1, we have

$$\begin{aligned} 0 &= \nabla\left\{\frac{1}{2}\left\|\tilde{v} - v\right\|_2^2 + \tau\lambda K(v)\right\} \\ &= v - \tilde{v} + \tau\lambda\frac{\partial K}{\partial v} \\ &= v - \tilde{v} - \tau\lambda\rho(m - \rho v), \\ \Rightarrow v = \mathrm{Prox}_{\tau K}(\tilde{v}) &= \frac{\tilde{v} + \tau\lambda\rho m}{1 + \tau\lambda\rho^2}. \end{aligned} \tag{6.19}$$

In section 3.2.1 and 3.2.2, we have defined several regularisers that were all an L^2-penalisation of some kind. We will calculate the proximal operator for general $\mu R(w) = \frac{\mu}{2}\left\|w\right\|_2^2$, where $w = Hv$, H being the operator working on v. In section 6.2, we will recall the different operators H and calculate their inverses, in order to use it in the PDHGM algorithm. In a similar way to (6.19), we obtain

$$w = \mathrm{Prox}_{\tau R}(\tilde{w}) = \frac{\tilde{w}}{1 + \tau\mu}. \tag{6.20}$$

6.2 Adjoint operators in regulariser

In section 3.2.1 and 3.2.2 we have derived several regularisers, of which we can write the part inside the pointwise L^2-norm as an operator working on v. Here we will state the operators and their adjoints. For our derivation, we refer to Appendix B. Again, we will assume that $\Omega \subset \mathbb{R}^2$. In this section, we will make use of $W^{1,2}(\Omega)$, the Sobolev space which contains all functions such that all partial derivatives of order 0 and 1, belong to $L^2(\Omega)$. When the functions have compact support in Ω, we write $W_0^{1,2}(\Omega)$. Furthermore, we use $W^{\mathrm{div},2}(\Omega) := \{v \in L^2(\Omega)^2 \mid \mathrm{div}(v) \in L^2(\Omega)\}$, the space that contains all functions of which its divergence belongs to $L^2(\Omega)$.

divergence

$$\begin{aligned} &\text{For } v \in L^2(0,1; W^{\mathrm{div},2}(\Omega)) \quad \text{with } v \cdot n|_{\partial\Omega} = 0, \\ &\text{and } w \in L^2(0,1; W^{1,2}(\Omega)), \end{aligned}$$

we have the adjoint operators

$$H_{_D} v = \mathrm{div}(v), \qquad H_{_D}^* w = -\nabla w. \tag{6.21}$$

rigidity

$$\begin{aligned} &\text{For } v = (v_1, v_2) \in L^2(0,1; (W_0^{1,2}(\Omega))^2), \\ &\text{and } w = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}, \end{aligned}$$

we have the adjoint operators

$$\begin{aligned} H_{_R} v &= \frac{\nabla v + (\nabla v)^T}{2}, \\ H_{_R}^* w &= -\frac{1}{2}\left(\mathrm{div}\begin{pmatrix} 2w_1 \\ w_2 + w_3 \end{pmatrix}, \mathrm{div}\begin{pmatrix} w_2 + w_3 \\ 2w_4 \end{pmatrix}\right). \end{aligned} \tag{6.22}$$

anisotropic

$$\begin{aligned} &\text{For } v = (v_1, v_2) \in L^2(0,1; (W_0^{1,2}(\Omega))^2), \\ &\text{and } w = (w_1, w_2, w_3, w_4), \end{aligned}$$

we have the adjoint operators

$$\begin{aligned} H_{_A} v &= (U\nabla v_1, U\nabla v_2), \\ H_{_A}^* w &= -\left(\mathrm{div}\begin{pmatrix} U_1 w_1 + U_3 w_2 \\ U_2 w_1 + U_4 w_2 \end{pmatrix}, \mathrm{div}\begin{pmatrix} U_1 w_3 + U_3 w_4 \\ U_2 w_3 + U_4 w_4 \end{pmatrix}\right), \end{aligned} \tag{6.23}$$

where $U = \begin{pmatrix} U_1 & U_2 \\ U_3 & U_4 \end{pmatrix}$. If we make a Cholesky factorisation as proposed in section 3.2.2, U_3 will become zero.

Note that we defined v to belong to spaces in which it is also square integrable in Ω: $v \in L^2(\Omega)$, although this is not necessarily needed, since v only appears in the coupling term $K(\rho, m, v)$ or in combination with a gradient or divergence. However, this does give us some extra control on v, and therefore it was easier to prove the existence of a minimiser of our second model, as we have seen in section 3.3.

6.3 Implementation of models 1 & 2 via PDHGM

In this section, we will consider the implementation of our models in the Modified Primal-Dual Hybrid Gradient algorithm (PDHGM) as proposed by Chambolle and Pock [**?**, 12]. We choose to use this algorithm, because in many minimisation problems, it shows fast convergence. More specifically, Papadakis *et al.* showed in [36] that for problem (3.6), out of the four algorithms they tested, PDHGM showed the fastest convergence, both in the energy functional $\mathcal{J}(\rho, m)$ and in the variables ρ and m.

With the use of our proximal operators and the operators working on v, we can implement our model in PDHGM. For model 1 (3.6), no regularisation is required, so only one minimisation problem has to be solved. This gives us **algorithm 1** as shown below. However, for model 2 (3.9), a regularisation is desired and we alternately need to solve two 'inner' minimisation problems (6.14). In the same spirit of the proximal Gauss-Seidel method [3, 2], we propose to alternately do one iteration of each inner PDHGM algorithm, instead of iteratively finding the solution to each of the inner problems. This way, we hope to find a solution to the joint problem quicker, since we do not have to solve the inner problem to small precision every time. This implementation gives us **algorithm 2**, as shown below. For better readability, we have written the couple $u = (\rho, m)$ as a single variable in our algorithms. Moreover, we only state the algorithms for our isotropic optimal transport models with J as an energy functional; the implementation for anisotropic models can be obtained by simply replacing $\text{Prox}_{\tau J}$ and $\text{Prox}_{\tau(J+K)}$ by $\text{Prox}_{\tau J_A}$ and $\text{Prox}_{\tau(J_A+K)}$ respectively.

algorithm 1: we search for $u := (\rho, m)$ that minimises (3.6).

- Initialise by choosing τ, $\sigma > 0$ such that $\tau\sigma < 1$.
- Choose $\theta \in [0, 1]$.
- Choose initial values u^0,q^0 and set $\bar{u}^0 = u^0$.
- Update as follows:

$$\begin{cases} q^{k+1} &= \text{Prox}_{\sigma\iota^*_{MC}}(q^k + \sigma\bar{u}^k), \\ u^{k+1} &= \text{Prox}_{\tau J}(u^k - \tau q^{k+1}), \\ \bar{u}^{k+1} &= u^{k+1} + \theta(u^{k+1} - u^k), \end{cases} \tag{6.24}$$

algorithm 2: we search for $u := (\rho, m)$ and v that minimises (3.9).

- Initialise by choosing τ_1, τ_2, σ_1, $\sigma_2 > 0$ such that $\tau_1\sigma_1 < 1$ and $\tau_2\sigma_2 < \|H\|^2$.
- Choose θ_1, $\theta_2 \in [0, 1]$.
- Choose initial values u^0,q^0,v^0,w^0 and set $\bar{u}^0 = u^0$, $\bar{v}^0 = v^0$.
- Update as follows:

$$\left\{ \begin{array}{l} \begin{cases} q^{k+1} &= \text{Prox}_{\sigma_1\iota^*_{MC}}(q^k + \sigma_1\bar{u}^k), \\ u^{k+1} &= \text{Prox}_{\tau_1(J+K)}(u^k - \tau_1 q^{k+1}), \\ \bar{u}^{k+1} &= u^{k+1} + \theta_1(u^{k+1} - u^k), \end{cases} \\ \\ \begin{cases} w^{k+1} &= \text{Prox}_{\sigma_2 R^*}(w^k + \sigma_2 H\bar{v}^k), \\ v^{k+1} &= \text{Prox}_{\tau_2 K}(v^k - \tau_2 H^* w^{k+1}), \\ \bar{v}^{k+1} &= v^{k+1} + \theta_2(v^{k+1} - v^k). \end{cases} \end{array} \right. \tag{6.25}$$

Note that there is no operator working on u and q, since we did not need one in our first model.

6.3.1 Numerical discretisation

Before these algorithms can be used, we first need to define a discretisation. We will first define the discretisation for algorithm 1 and the first part of algorithm 2. To keep notation simple, we assume our space domain Ω to have a rectangular shape. For other shapes, consisting of multiple rectangular subdomains, the derivation is similar.

From section 6.1, we know that pointwise explicit expressions for $\text{Prox}_{\tau J}$ and $\text{Prox}_{\tau_1(J+K)}$ exist. In order to do (discrete) pointwise calculations, we need ρ and m to be defined on the same grid. Therefore, we define the following variables on a centred grid discretising our space-time box $\Omega \times [0,1]$:

$$\begin{aligned}
m_{1,c,ijk} &:= m_1(x_1 = \tfrac{i}{N}, x_2 = \tfrac{j}{P}, t = \tfrac{l}{Q}),\ 0 \leq i \leq N,\ 0 \leq j \leq P,\ 0 \leq l \leq Q,\\
m_{2,c,ijk} &:= m_2(x_1 = \tfrac{i}{N}, x_2 = \tfrac{j}{P}, t = \tfrac{l}{Q}),\ 0 \leq i \leq N,\ 0 \leq j \leq P,\ 0 \leq l \leq Q,\\
\rho_{c,ijk} &:= \rho(x_1 = \tfrac{i}{N}, x_2 = \tfrac{j}{P}, t = \tfrac{l}{Q}),\ 0 \leq i \leq N,\ 0 \leq j \leq P,\ 0 \leq l \leq Q,
\end{aligned}$$

where we use c to denote the centred grid and where N, P and Q are the amount of discrete points in each direction of the domain.

For the projection step $\text{Prox}_{\tau\iota_{MC}}$, we recall from (6.3), that one first has to solve a Poisson equation. In order to solve this equation, we need the variables $\partial_t \rho$ and $\text{div}(m)$ to be defined on the same centred grid. Therefore, we also define variables on a staggered grid:

$$\begin{aligned}
m_{1,s,ijk} &:= m_1(x_1 = \tfrac{i-1/2}{N}, x_2 = \tfrac{j}{P}, t = \tfrac{l}{Q}),\ 0 \leq i \leq N+1,\ 0 \leq j \leq P,\ 0 \leq l \leq Q,\\
m_{2,s,ijk} &:= m_2(x_1 = \tfrac{i}{N}, x_2 = \tfrac{j-1/2}{P}, t = \tfrac{l}{Q}),\ 0 \leq i \leq N,\ 0 \leq j \leq P+1,\ 0 \leq l \leq Q,\\
\rho_{s,ijk} &:= \rho(x_1 = \tfrac{i}{N}, x_2 = \tfrac{j}{P}, t = \tfrac{l-1/2}{Q}),\ 0 \leq i \leq N,\ 0 \leq j \leq P,\ 0 \leq l \leq Q+1,
\end{aligned}$$

where we use s to denote the staggered grid. The divergence and gradient operators in the projection operator (6.3) are applied with a centred difference scheme, in order to go from the staggered grid to a centred grid, after which the Poisson equation is solved and to go to a staggered grid again.

Since we use a staggered grid in the first step of algorithm 1 and a centred grid in the second step, we make use of interpolation operators I_c and I_s:

$$\begin{aligned}
I_c :&(m_{1,c}, m_{2,c}, \rho_c) \mapsto (m_{1,s}, m_{2,s}, \rho_s),\\
\text{where } m_{1,s,1jk} &= m_{1,c,1jk},\\
m_{1,s,ijk} &= \frac{m_{1,c,(i-1)jk} + m_{1,c,ijk}}{2} \quad \text{for } \ 2 \leq i \leq N\\
m_{1,s,(N+1)jk} &= m_{1,c,Njk}.
\end{aligned}$$

The interpolation for m_2 and ρ from a centred to a staggered grid is analogous.

$$\begin{aligned}
I_s :&(m_{1,s}, m_{2,s}, \rho_s) \mapsto (m_{1,c}, m_{2,c}, \rho_c),\\
\text{where } m_{1,c,ijk} &= \frac{m_{1,s,ijk} + m_{1,s,(i+1)jk}}{2} \quad \text{for } \ 1 \leq i \leq N.
\end{aligned}$$

The interpolation for m_2 and ρ from a staggered to a centred grid is analogous.

For the proximal operator $\text{Prox}_{\tau K}$, a pointwise explicit expression (section 6.1) is available. Since the expression is a combination of the variables ρ, m and v, we make use of a centred grid, so they are defined on the same points:

$$\begin{aligned}
v_{1,c,ijk} &:= v_1(x_1 = \tfrac{i}{N}, x_2 = \tfrac{j}{P}, t = \tfrac{l}{Q}),\ 0 \leq i \leq N,\ 0 \leq j \leq P,\ 0 \leq l \leq Q,\\
v_{2,c,ijk} &:= v_2(x_1 = \tfrac{i}{N}, x_2 = \tfrac{j}{P}, t = \tfrac{l}{Q}),\ 0 \leq i \leq N,\ 0 \leq j \leq P,\ 0 \leq l \leq Q.
\end{aligned}$$

Checkerboard instabilities are well known in computational fluid dynamics [42, page 314]. These instabilities occur when one takes a central discretisation when applying a difference operator. One of the solutions for checkerboard instabilities is to use staggered discretisation schemes, as we did for the projection operator. Another solution is to use a forward difference scheme for the difference operators and a backward difference scheme for the adjoint operators. Here we will do the latter. For the divergence operator, the discrete operator DIV reads

$$\begin{aligned}
\mathrm{DIV}(v)_{ijk} &= \frac{v_{1,c,(i+1)jk} - v_{1,c,ijk}}{h_1} + \frac{v_{2,c,i(j+1)k} - v_{2,c,ijk}}{h_2} \quad \text{for } i \neq N, j \neq P,\\
\mathrm{DIV}(v)_{Njk} &= \frac{v_{2,c,N(j+1)k} - v_{2,c,Njk}}{h_2} \quad \text{for } j \neq P,\\
\mathrm{DIV}(v)_{iPk} &= \frac{v_{1,c,(i+1)Pk} - v_{1,c,iPk}}{h_1} \quad \text{for } i \neq N,\\
\mathrm{DIV}(v)_{NPk} &= 0.
\end{aligned}$$

where h_1 and h_2 are the sizes between two consecutive points in x_1- and x_2-direction respectively. Here we used zero padding at the boundaries at $i = N$ and $j = P$. For the adjoint operator to be defined properly, we need backward differences with zero padding at $i = 1$ and $j = 1$. The other operators are implemented with similar forward differences and backward differences for their adjoints.

6.3.2 Numerical convergence

In this section, we will show the numerical convergence of both algorithms. In Figure 9 the convergence plots have been shown for algorithm 1 and 2. The data set that was used to create these convergence results is the synthetic data set that shows flow in a channel, as explained in section 7.1.

The top convergence plots correspond to the result in Figure 12b. The bottom convergence plots correspond to Figure 12e, in which a rigidity regulariser was applied.

In the top left we see that all the primal and dual variables are self-convergent. Moreover, in the top right we see that the energy functional $J(\rho, m)$ converges to a value around 240. In the bottom left, we see that all the variables in algorithm 2 are self-convergent, similar to the top left. In the bottom right however, we see a convergence plot that is very different from the top right: first the energy functional $J(\rho, m)$ and the regulariser $R(v)$ are getting lower in value, while the coupling $K(\rho, m, v)$ is increasing. After some iterations, the value of the coupling is decreasing, followed by an increase of the energy functional and later the regulariser. Finally, at about 1000 iterations, an equilibrium is found, and the result is stable. In the total energy we also first see a decrease, followed by an increase and finally a stable value. This result is probably due to the rather weak coupling between the first and the second part of algorithm 2. Both parts are only coupled by an the term $\frac{1}{2}|m - \rho v|^2$, which as an L^2-norm is much weaker than for instance the projection on the set $\{m = \rho v\}$. The L^2-norm first leaves some room for the energy functional and regulariser to deviate from a joint solution, after which a joint stable point is found.

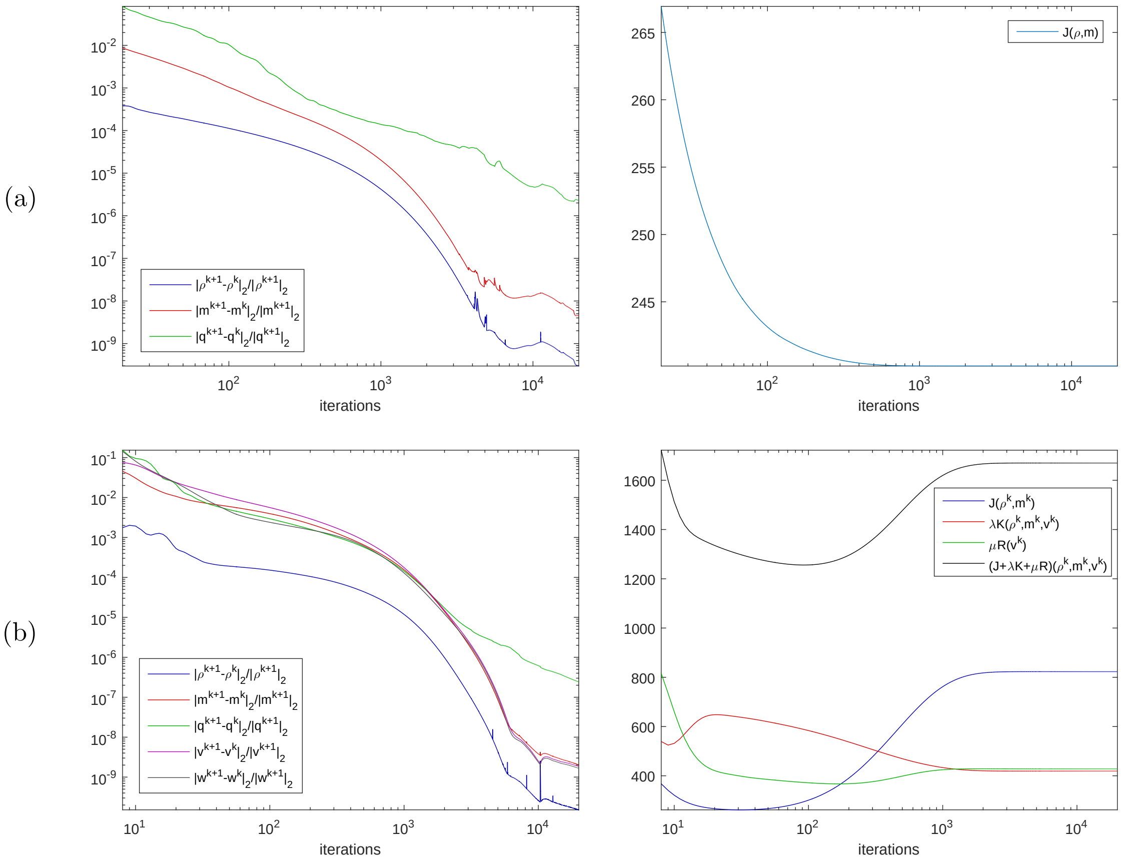


Figure 9: Convergence plot of the variables left and plot of the different parts of the minimisation functional right: (a) model 1 with $\sigma = 45$, $\tau = 0.02$; (b) model 2 with rigidity regulariser, $\lambda = 10$, $\mu = 10^{-4}$, $\sigma_1 = 45$, $\tau_1 = 0.02$, $\sigma_2 = 4.5 \cdot 10^{-4}$, $\tau_2 = 0.01$.

6.4 Implementation of joint model

In order to implement (5.2), the alternating minimisation scheme (6.14) is changed to calculate two velocities, and extended with an extra step in which the value for z is calculated:

$$\begin{cases} (\rho^{k+1}, m^{k+1}) & = \operatorname*{argmin}_{(\rho,m)}\big\{ \int_0^1 \int_\Omega J_A(\rho, m) + \iota_{MC}(\rho, m) + \lambda K(\rho, m, v^k)\big\}, \\ v_{in}^{k+1} & = \operatorname*{argmin}_{v_{in}}\big\{ \int_0^1 \int_\Omega \lambda(z^k)^2 K(\rho^{k+1}, m^{k+1}, v_{in}) + \mu R(v_{in})\,\mathrm{d}x\,\mathrm{d}t\big\}, \\ v_{out}^{k+1} & = \operatorname*{argmin}_{v_{out}}\big\{ \int_0^1 \int_\Omega \lambda(1-z^k)^2 K(\rho^{k+1}, m^{k+1}, v_{out}) + \mu R(v_{out})\,\mathrm{d}x\,\mathrm{d}t\big\}, \\ z^{k+1} & = \operatorname*{argmin}_{z}\big\{ \int_0^1 \int_\Omega \lambda z^2 K(\rho^{k+1}, m^{k+1}, v_{in}^{k+1}) \\ & \qquad +\lambda(1-z)^2 K(\rho^{k+1}, m^{k+1}, v_{out}^{k+1}) + \beta TV(z)\,\mathrm{d}x\,\mathrm{d}t\big\}, \\ v^{k+1} & = [z^{k+1}]v_{in}^{k+1} + (1-[z^{k+1}])v_{out}^{k+1}. \end{cases} \tag{6.26}$$

It should be noted that it is not clear if we can expect to get a decent solution by using four alternating minimisation steps that use each others output variables as input variables. Before we can state the algorithm that we will use to try to solve the joint model, we need to define the proximal operators for the fourth step of (6.26). Therefore, we look at the definition of the total variation:

$$TV(z) = \sup_{\substack{g\in C_0^1(\Omega)\\ \|g\|_\infty \le 1}} \langle z, \operatorname{div}(g)\rangle. \tag{6.27}$$

We write $H_D g = \operatorname{div}(g)$ as an operator working on $g \in C_0^1(\Omega)$ and define

$$G(z) = z^2(\lambda K(v_{in})) + (1-z)^2(\lambda K(v_{out})), \tag{6.28}$$

where $K(v_{in/out})$ is a shorter notation for $K(\rho, m, v_{in/out})$. Rewriting the fourth equation of (6.26) yields the following saddle-point problem:

$$\begin{aligned} &\min_z \big\{ \int_0^1 \int_\Omega \lambda z^2 K(v_{in}^{k+1}) + \lambda(1-z)^2 K(v_{out}^{k+1}) + \beta TV(z)\,\mathrm{d}x\,\mathrm{d}t\big\}2 \\ =&\min_z \big\{ \int_0^1 \int_\Omega G(z) + \beta \sup_{\|g\|_\infty\le 1} \langle z, H_D g\rangle\,\mathrm{d}x\,\mathrm{d}t\big\} \\ =&\min_z \max_g \big\{\beta\langle z, H_D g\rangle + G(z) - F^*(g)\big\}, \end{aligned} \tag{6.29}$$

$$\text{where } F^*(g) = \begin{cases} 0 & \text{if } g \in P, \\ \infty & \text{else,} \end{cases}$$

and $P = \{g : \ \|g\|_\infty \le 1\}$.

Once the proximal operators for $G(z)$ and $F^*(g)$ are known, saddle-point problem (6.29) can be directly implemented in PDHGM [12]. We calculate

$$\begin{aligned}
z = \mathrm{Prox}_{\tau G}(\tilde{z}) &= \operatorname*{argmin}_{\hat{z}} \left\{ \frac{1}{2} \|\tilde{z} - \hat{z}\|^2 + \tau G(\hat{z}) \right\} \\
&= \operatorname*{argmin}_{\hat{z}} \left\{ \frac{1}{2} \|\tilde{z} - \hat{z}\|^2 + \tau \left(\hat{z}^2 \lambda K(v_{in}) + (1-\hat{z})^2 \lambda K(v_{out}) \right) \right\} \\
\Rightarrow 0 &= (z - \tilde{z}) + \tau \left(2z\lambda K(v_{in}) + 2(z-1)\lambda K(v_{out}) \right) \\
\Rightarrow z &= \frac{\tilde{z} + 2\tau\lambda K(v_{out})}{1 + 2\tau\lambda(K(v_{in}) + K(v_{out}))},
\end{aligned}$$

$$g = \mathrm{Prox}_{\tau F^*}(\tilde{g}) = \mathrm{Proj}_P(\tilde{g}) = \frac{g}{\max\{1, \|g\|_\infty\}}. \tag{6.30}$$

In **algorithm 3** the algorithm for the joint model is shown.

algorithm 3: we search for $u := (\rho, m)$, v and z that minimises (5.2).

- Initialise by choosing τ_1, τ_2, τ_2, σ_1, σ_2, $\sigma_3 > 0$ such that $\tau_1\sigma_1 < 1$, $\tau_2\sigma_2 < \|H\|^2$ and $\tau_3\sigma_3 < \|H_D\|^2$.
- Choose θ_1, θ_2, $\theta_3 \in [0, 1]$.
- Choose initial values for all variables.
- Update as follows:

$$\left\{ \begin{array}{l}
\begin{cases}
q^{k+1} &= \mathrm{Prox}_{\sigma_1 \iota^*_{MC}}(q^k + \sigma_1 \bar{u}^k), \\
u^{k+1} &= \mathrm{Prox}_{\tau_1(J+K)}(u^k - \tau_1 q^{k+1}), \\
\bar{u}^{k+1} &= u^{k+1} + \theta_1(u^{k+1} - u^k),
\end{cases} \\[2em]
\begin{cases}
w^{k+1} &= \mathrm{Prox}_{\sigma_2 R^*}(w^k + \sigma_2 H \bar{v}^k_{in/out}), \\
v^{k+1}_{in/out} &= \mathrm{Prox}_{\tau_2 K}(v^k_{in/out} - \tau_2 H^* w^{k+1}), \\
\bar{v}^{k+1}_{in/out} &= v^{k+1}_{in/out} + \theta_2(v^{k+1}_{in/out} - v^k_{in/out}).
\end{cases} \\[2em]
\begin{cases}
g^{k+1} &= \mathrm{Prox}_{\sigma_3 F^*}(g^k + \sigma_3 H_D \bar{z}^k), \\
z^{k+1} &= \mathrm{Prox}_{\tau_3 G}(z^k - \tau_3 H_D^* g^{k+1}), \\
\bar{z}^{k+1} &= z^{k+1} + \theta_3(z^{k+1} - z^k).
\end{cases} \\[2em]
v^{k+1} = [z^{k+1}] v^{k+1}_{in} + (1 - [z^{k+1}]) v^{k+1}_{out}.
\end{array} \right. \tag{6.31}$$

In this algorithm, $\mathrm{Prox}_{\tau_2 K}(v_{in/out})$ has been altered to include the terms z^2 and $(1-z)^2$ respectively.

7 Results and discussion

In this chapter, we will show and discuss the results of both optimal transport models applied on several data sets. We will first consider two synthetic data sets, with the advantage that we do not have any noise on the images and that a ground truth transport is available. Next we will consider the three realistic data sets as shown and explained in section 2.2. All data sets contain two-dimensional rectangular images. Finally, we show and discuss the results of applying our joint model on one of the synthetic data sets. Throughout this chapter, parameters are chosen such that we visually obtained the best result. All the variables, except ρ, have an initial value of 0. For the initial density ρ an interpolation in time between ρ_0 and ρ_1 was taken: $\rho(x,t) = (1-t)\rho_0(x) + t\rho_1(x)$.

7.1 Synthetic data

We have created two synthetic data sets, which are a simplification of the realistic data that we expect to handle with our two models. In the first data set, we wish to model the transport of two distinct objects within the image, where transport through the boundary of the image is prohibited. In the second data set, we wish to model the transport of density in a channel-like structure, where transport through the boundary of the image is allowed.

7.1.1 Moving Gaussians

The initial image of the first data set consists of two Gaussians; one in the top and one in the bottom of the image domain. The final image contains these exact same Gaussians, but now rotated by almost 90 degrees. In Figure 10, we see the resulting interpolation of the densities at several points in time. In the first row, model 1 is used, since no regulariser was applied. In the second and third row, model 2 with an incompressibility respectively rigidity regulariser is used. The goal for this data set is to see the effect of the regulariser on the transport. To obtain good results, 31 time steps were used, but only six are shown.

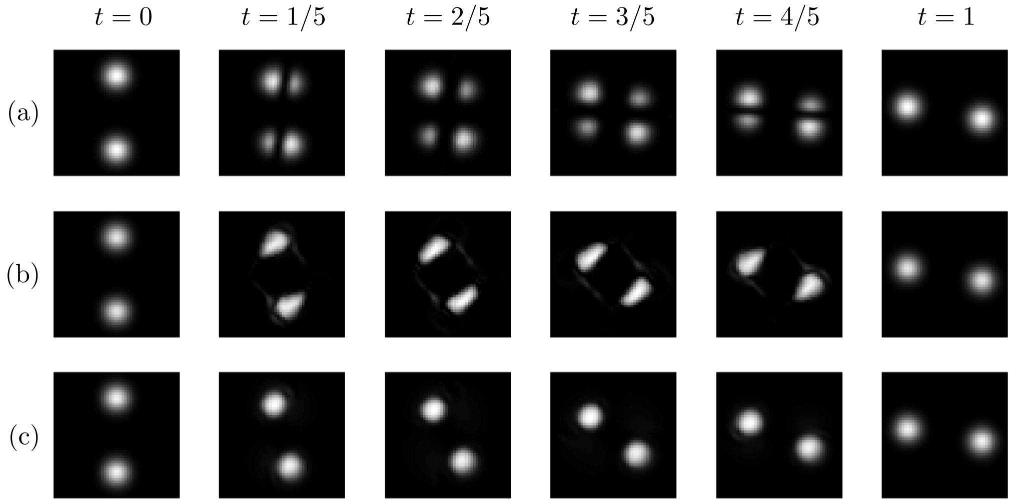


Figure 10: Interpolation of two Gaussians using different regularisers: (a) without regulariser; (b) incompressibility regulariser; (c) rigidity regulariser.

It can be seen that if no regulariser is applied, we obtain an interpolation in which the Gaussians first split and then re-emerge. The intuitive explanation for this is very straightforward: every initial density 'particle' wants to travel as little as possible, therefore travelling in the direction in which the final density is closest. We see that we get a slight loss of brightness during the transport, although not very extreme. When an incompressibility regulariser is applied, we obtain an interpolation in which the Gaussians mostly stay in one piece during the transport. A very small piece is still moving in a different direction, therefore giving a small penalisation of the divergence. We see that the Gaussians get deformed, but we do not have a loss or increase of brightness along the transport, so incompressibility is ensured. When a rigidity regulariser is applied, we obtain an interpolation that consists of two piecewise translations. There is no loss of brightness and the Gaussians stay in two pieces.

These results show that an incompressibility regulariser is somewhat 'less restrictive' than the rigidity regulariser: if a flow is incompressible, it might still deform the density, whereas this is not possible in the rigid case.

7.1.2 Flow in channel

The second data set consists of two images (Figure 11) with a background that stays at the same place and a channel-like structure in the foreground that is shifted three pixels to the right from the first to the second image. The amount of density that disappears from the first to the second image on the right side, re-appears on the left side of the second image, such that the total density in both images stays the same. Our goal for this data set is to see what the transport looks like when different models are applied. Since we know exactly which part of the image was shifted to the right, a ground truth to compare our results with is available. Moreover, we have an a-priori segmentation, which can be used in our models, to see if it benefits the results.

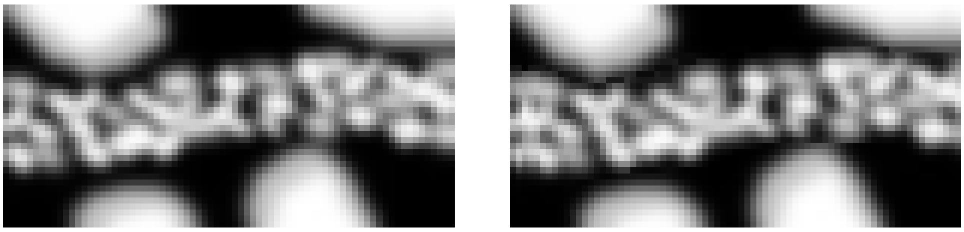


Figure 11: Synthetic data of flow in a channel: the middle part of the image has been translated three pixels to the right from the first image to the second.

For this data set, a wide variety of models have been applied, in order to see their capability of capturing a flow that is only apparent in part of the image. In Figure 12 results are shown. On the left hand side, the velocity is colour-coded. The direction of the velocity coincides with the location of the colours on the boundary of the image. Furthermore, if the absolute value of the velocity is lower, the colour will be more white. Since the 'whiteness' of the velocity is not always very visible, the absolute value of the velocity is shown on the right hand side, where the scale is the same for all results. It should be noted that we obtained a unique velocity field for every discrete time-step, but we only show the velocity field that belongs to the first time step. This is justified by the fact that the velocity looks very similar for all time steps. That is, the velocity field is almost constant in time.

In the first row, the ground truth velocity field is shown, which is a constant velocity directed to the right, with an absolute value of $|v| = 0.6$. It is important to keep in mind that many other ground truths would have given the exact same result. For instance, it does not matter what the velocity field is on locations where there is no density, since there is nothing to be transported.

In the second row, we see the result for an isotropic flow where no regularisers have been applied. As can be seen, the direction of the flow at the boundary of the image coincides with the ground truth, but inbetween it changes direction rather often. Moreover, we see that the absolute value of

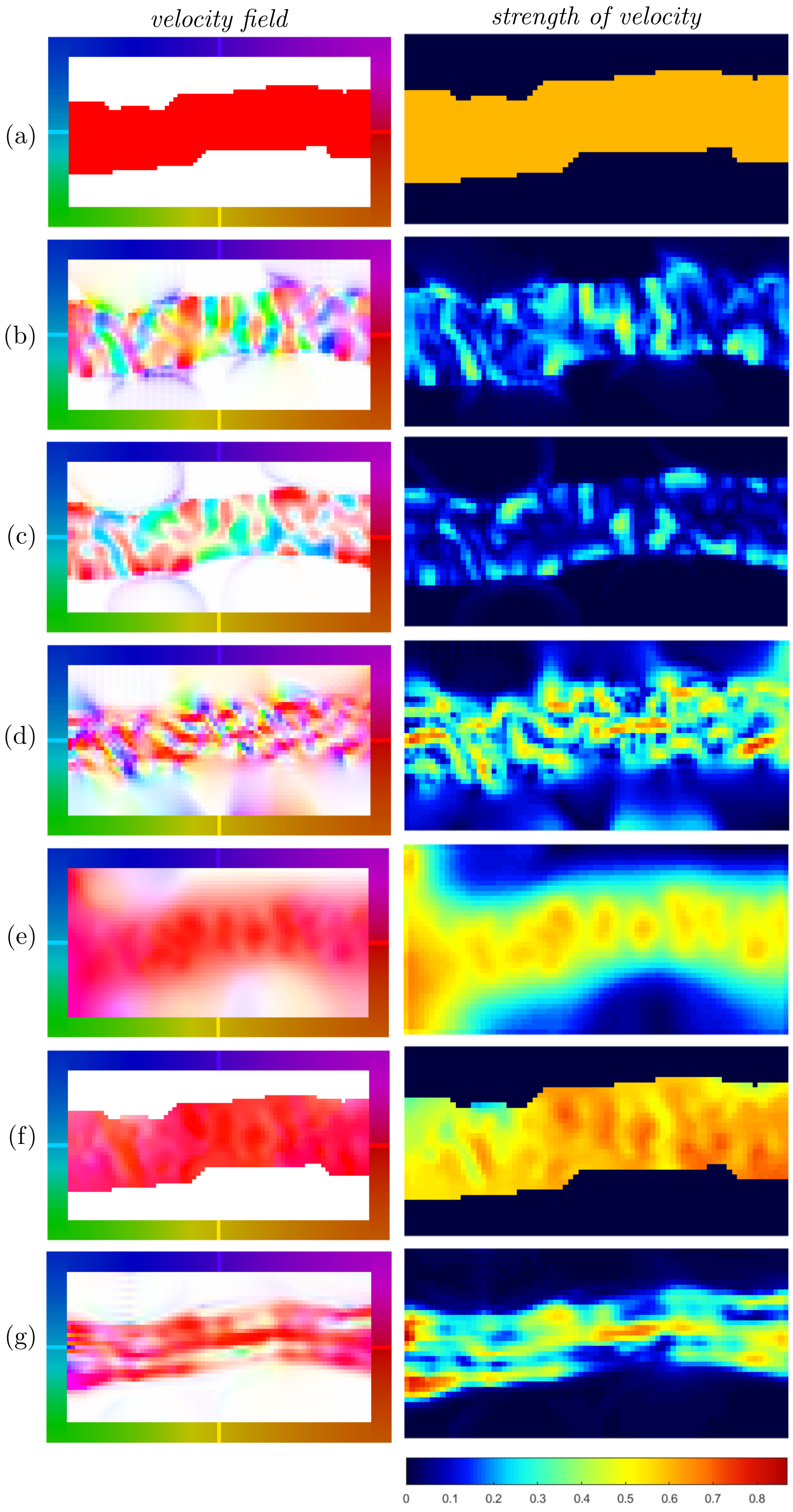


Figure 12: Velocity field and absolute value of the velocity for the flow in a pipe with different models: (a) ground truth; (b)-(c) no regulariser; (b) isotropic transport cost; (c) anisotropic transport cost; (d)-(g) isotropic transport cost with regulariser; (d) incompressibility regulariser; (e) rigidity regulariser; (f) rigidity regulariser with a-priori binary segmentation; (g) anisotropic regulariser.

the velocity is much lower than the ground truth, which explains why this flow is preferred above the ground truth.

In the third row, we have used an anisotropic model, where the anisotropy was obtained by the coherence enhancing diffusion tensor (section 3.2.2). We see that more of the flow is directed to the right, but the flow direction is still far from uniform. Here the absolute value of the velocity is also much lower than the ground truth.

In the fourth row, an incompressibility regulariser in combination with an isotropic model has been applied. We see many small regions where the direction of the flow is the same as the ground truth, but inbetween, the flow takes all kind of directions. As was already mentioned in the previous section, a incompressibility regulariser is not as restrictive as a rigid one and thus still gives quite some freedom in the resulting flow. This result does not give us an overall coherent flow. Moreover, the absolute value of the flow is not close to the ground truth, but at some places higher, and at some places much lower.

In the fifth row, a rigidity regulariser has been applied, and it is immediately clear that it is very similar to the ground truth. A difference with the ground truth is that flow is visible over the whole left side of the image. However, since there is no density here, the flow in this part of the image is not expected to completely coincide with the ground truth. Furthermore, we see that there is a gradual decrease of velocity strength from the middle to the top and bottom. This is due to the L^2-penalisation in the regulariser, which prefers a smooth transition between a high and low velocity. Finally, notice that if we would threshold the absolute value at $|v| = 0.3$, we would roughly get the segmentation back that is known from the ground truth. This will be used later on in our joint model (section 7.3).

In the sixth row, an a-priori binary segmentation has been applied, as well as a rigidity regulariser. As expected, we almost get the exact ground truth flow back, both in direction as in strength. The drawback of this model is that an a-priori segmentation is usually not known and can be hard to obtain by segmentation methods. In section 7.2 we will make use of segmentation methods to obtain an a-priori segmentation. In section 7.3, we will use the joint model to obtain a segmentation whilst calculating the optimal transport.

In the last row, we have used an isotropic model in combination with an anisotropic regulariser. We see that the velocity is very smooth along the direction of the channel, but at some places not very smooth perpendicular to it. The direction of the flow is overall quite similar to the ground truth, but it is clear that the absolute value of the velocity differs a lot within the channel.

In many results of this section, we see that the strength of the velocity is lower than the ground truth velocity. This can be explained as follows: when we compare this data set to the previous one, we see that there is a ‘ground density’ in the whole channel, whereas in the previous set, there was no density in the paths that were followed by the Gaussians. Instead of moving with the ground truth velocity, we obtain a lower transport cost if the ‘top density’ takes the place of the ground density further on, and the ground density takes the place of the top density even farther away. A more extensive explanation, including a very simple example, can be read in appendix C. A solution for this lower velocity could be obtained if one would first do some kind of clever background subtraction, for example by using Principal Component Analysis [9].

7.2 Real biomedical data

In section 2.2, three realistic data sets have been introduced, which will be used as inputs for our optimal transport models. For the first data set, we wish to model the non-straight transport of two cells, where transport through the boundary of the image is prohibited. In the second data set, we wish to model the transport of haemoglobin through blood vessels, where flow through the boundary of the image is expected. In the third data set, we are dealing with an image in which there is only an inflow of density in the bottom part of the image, which spreads through the image in a specific structure.

7.2.1 Moving white blood cell

For this set of results, only the first and last image of Figure 3 are used as input. As is almost always the case with real data, the raw images contain some noise and are therefore not suitable to use directly.

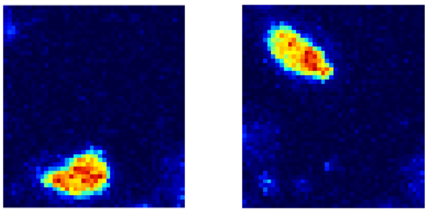
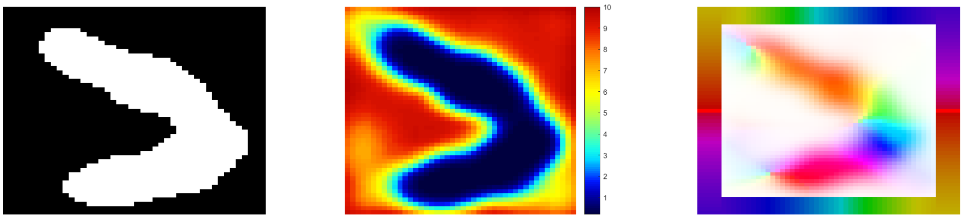


Figure 13: Colourful plot of the initial and final density of a moving white blood cell. Noise can be seen in the background.

In Figure 13, colourful plots of the initial and final density are shown. It is clear that there is some density in the background which is not desired, since we are only interested in the movement of the cell. Since our model was not designed to cope with noise, some preprocessing needs to be done. A Gaussian filter is applied to smoothen the initial and final density, after which the image is thresholded in order to remove the background density. Finally, the image is normalised to make sure that the total initial density is the same as the total final density. The result of this procedure gives us the far left and far right image of Figure 15.



(a) Binary segmentation of the channel

(b) Values of β for the functional (4.5)

(c) Colourful anisotropy plot.

Figure 14: Masks and anisotropy for the cell movement.

We want to obtain the curved c-shaped motion as is seen in Figure 3. Therefore, both a model with a weak segmentation (section 4.2.2) and an anisotropic model (section 4.2.3) are applied. If a segmentation was available, we would be able to use this directly in our model. In our case, a segmentation is not available, so we make use of the ground truth information. All six images are added and thresholded at some value, to obtain a binary segmentation (mask) that coincides with the path the cell has travelled (Figure 14a). Next, the mask is smoothed with a Gaussian filter and the value of β in (4.5) wass set to 1 when inside the channel and set to a high value when outside the channel, with a smooth transition inbetween (Figure 14b). To obtain the anisotropy matrix, the coherence enhancing diffusion filter is applied to the binary segmentation and equation (4.6) is used to obtain the matrix A_{CE} (Figure 14c). To obtain good results, 31 time steps were used, but only the six that coincide with the ones from the ground truth are shown.

In Figure 15 it can be seen that if no anisotropy or segmentation is used, we get a translational transport with some deformation of the cell. The result of the anisotropic model looks very much

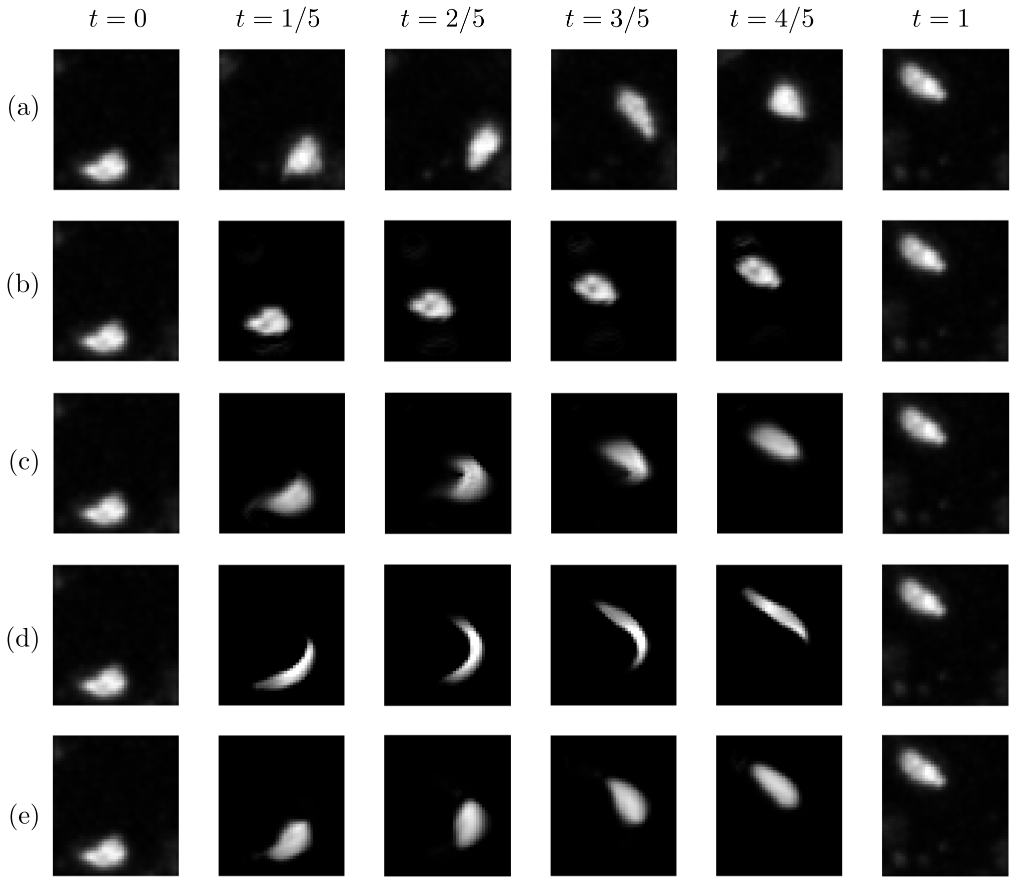
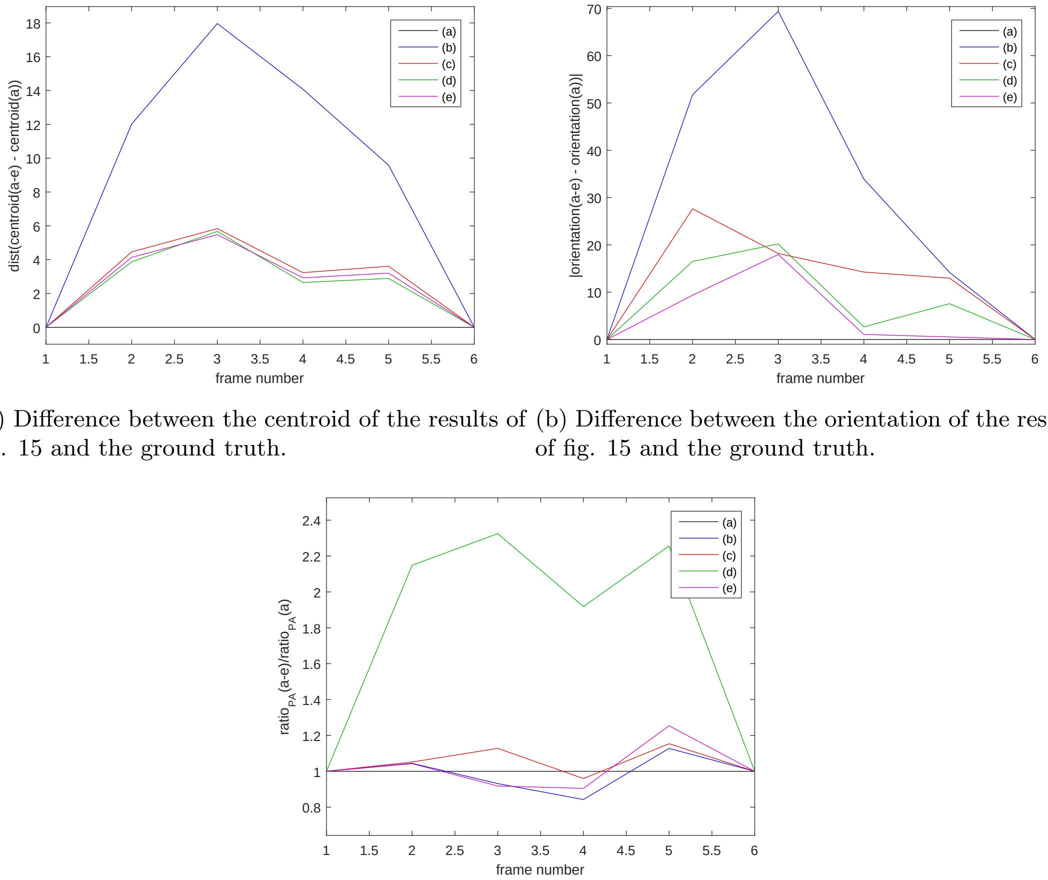


Figure 15: Interpolation of a moving cell with different models: (a) ground truth; (b) isotropic, no segmentation, no rigidity; (c) anisotropic, no segmentation, no rigidity; (d) isotropic, with segmentation, no rigidity; (e) isotropic, with segmentation, with rigidity.

like the ground truth, including the desired c-shape. However, in the third and fourth image, we see a sharp corner in the density, due to the anisotropy created from the binary mask. When a weak segmentation is used, we obtain the desired c-shape. It is clear that we also need a rigidity regulariser if we do not want the cell to become very elongated during the transport.



(a) Difference between the centroid of the results of fig. 15 and the ground truth.

(b) Difference between the orientation of the results of fig. 15 and the ground truth.

(c) Ratio of the results of fig. 15 divided by the ratio of the ground truth.

Figure 16: Comparison between the results obtained by the different models and the ground truth.

As can be seen in Figure 15, the anisotropic model and the isotropic model with weak segmentation and rigidity both give a promising result. However, it is hard to say which of the models gives a better result. Therefore, a set of properties has been chosen to compare the results of Figure 15. First, the results is thresholded to get a connected object for all of the six frames. Next, the MATLAB-function `regionprops` is used to extract some properties that will be compared. The first property is the 'centroid', or centre of mass. A solution which is visually close to the ground truth should at least be located around the same centroid. The centroid gives us a coordinate in an xy-coordinate system. The second property is the orientation of the cell. The orientation is defined as the angle between the horizontal axis and the major axis of the ellipse that has the same second-moments as the cell. The value is given between $-90°$ and $90°$. The last property that is used to compare the results is the ratio of the perimeter squared to the area: $\text{ratio}_{PA} = \frac{\text{Per}(S)^2}{\text{Area}(S)}$, where S is the region of the thresholded cell. A perfectly round object would give us a ratio of 4π, where the ratio of a very elongated object becomes very large.

In Figure 16, we present plots of the Euclidean distance between the centroids, the absolute difference between the orientation and the ratio of the results divided by the ratio of the ground truth. For result (b), it is clear that it does not score well with respect to the location of the centroid and the orientation of the cell, therefore not being close to the ground truth. Result (c) scores well for all properties, although it is slightly worse than (d) and (e) in orientation. Result (d) does not score well with respect to the ratio, because it is much more elongated than the ground truth, but it does score well in other properties. Result (e) scores very well with respect to all properties, therefore making it the result that is closest to the ground truth, when taking our chosen properties to compare (b) to (e).

Although the result with a weak segmentation and rigidity is slightly better than the anisotropic one, it is worth noticing that we obtain two very similar outcomes, while the underlying models are very different; the anisotropic model only requires a single algorithm and not a coupled one.

7.2.2 Haemoglobin microcirculation

For this set of results, a small part of two consecutive frames of the haemoglobin video image (section 2.2.2) was taken as input. The contrast in the image was inverted in order to have a high density (light) inside the blood vessels and a low density (dark) outside. In Figure 17, the initial and final density are shown.

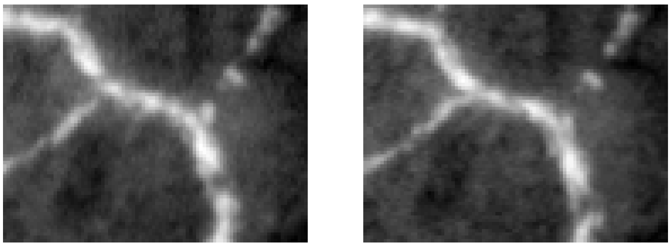
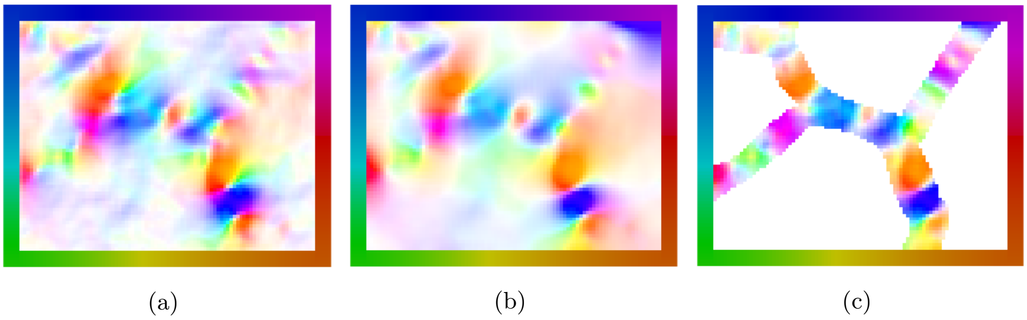


Figure 17: Part of two consecutive frames representing haemoglobin microcirculation: initial and final density.

The density and flow of the haemoglobin microcirculation is in some aspects similar to the synthetic data set of flow in a channel (Figure 11): both show a density with some structure moving through the boundary of the image. Moreover, the background is (almost) nonmoving. However, in the current data set, the vascular structures are slightly thinner, the density has a little bit less structure and there are bifurcations in the vessel network. Also, the flow does not have one specific direction, but is only roughly pointed in one direction throughout the image.



(a)

(b)

(c)

Figure 18: Velocity fields for haemoglobin microcirculation: (a) isotropic, no regulariser; (b) isotropic, rigidity regulariser; (c) isotropic, rigidity regulariser with a-priori binary segmentation.

Inspired by the promising results from Figure 12, three models have been applied to the current data set. The results are shown in Figure 18. On the left we see the result of applying the isotropic variant of model 1, without using any regulariser. The velocity field shows many directions, just as in Figure 12a. We see that the flow is stronger inside the blood vessels than outside the vessels, as expected. In the middle we see the result of applying the isotropic model with rigidity regulariser. In contrast to the result seen in 12e, the velocity field does not show one principal direction, but seems to change direction a couple of times along the vessel. However, it is mostly directed along the vessel and not perpendicular to it. On the right, we see the result of applying the isotropic model with rigidity regulariser and an a-priori binary segmentation. The velocity field has a more piecewise constant behaviour, but it is still does not have one main direction.

An explanation for these results can be the following: the transport cost $J(\rho, m)$ for the results in Figures 18b and 18c is lower than the transport cost that would belong to a more smooth flow in one main direction. This causes the preference for a piecewise smooth flow instead of a completely smooth one. In general, the rigidity regulariser helps to make the flow smoother, but in this case it does not help if we increase its parameter μ. The reason for this could be that in any of the cases (either the obtained results or the expected flow), the transport is not completely rigid, therefore causing a value for the regulariser $R(v)$ anyway.

An extra difficulty in this data set compared to the synthetic one is that we now do not have a completely static background. Moreover, the background density is now much higher than in the synthetic case. It seems that the model has difficulties with these extra obstacles.

7.2.3 Flow in human brain

In Figure 19, two preprocessed frames from the video sequence described in section 2.2.3 are shown. There is a clearly visible increase of density between the first and the second image, due to an inflow in the bottom of the image. Because flow is only expected within the channel-like structure, the isotropic model will be compared with the anisotropic model. For the anisotropic model, again the coherence enhancing diffusion tensor was used to make the anisotropy matrix (section 4.2.3). Since the total density increases over time, we enforce a linear increase, as explained in section 3.1.1. That is, we enforce $\int_\Omega \rho(x,t)\,\mathrm{d}x = t\int_\Omega \rho_0\,\mathrm{d}x + (1-t)\int_\Omega \rho_1\,\mathrm{d}x$. We note that no segmentation is used for this data set and that flow is able to flow through any of the boundaries.

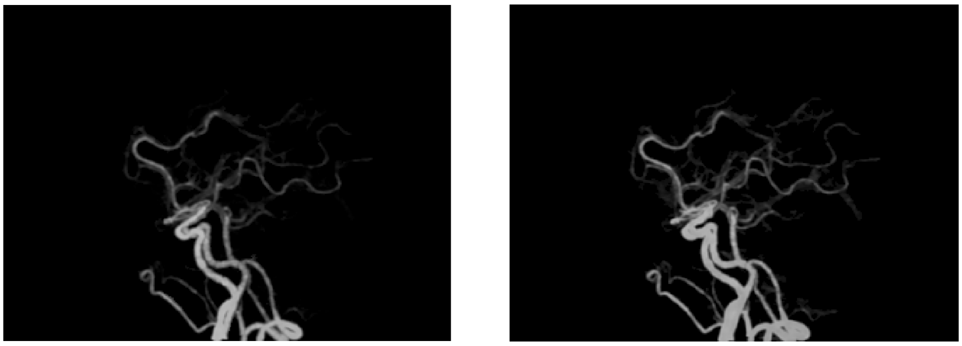


Figure 19: Preprocessed data of flow in a human brain: there is an increase of density from the first image to the second, which is caused by inflow from the bottom.

In Figure 20, both the colour-coded velocity field (of the first time step) and the absolute value of the velocity are shown for the isotropic and anisotropic model. For both models, the result shows an overall movement from bottom to top, which means that the inflow appears at the bottom of the image, as expected. Furthermore, we see that the anisotropic result gives a great improvement compared to the isotropic result: in the isotropic case density is flowing inbetween vessels, whereas in the anisotropic case the flow follows the vessel network. This can be seen by the absence of a purple colour between the vessels in the anisotropic result. Moreover, in the branches that are laid out more horizontally, we see more colours than only purple. This means that the flow is more oriented to the side, instead of only to the top. In the plots on the right side of Figure 20, a rather homogeneous strength of the velocity can be seen in the isotropic case. In the anisotropic case, the velocity is more concentrated in the middle of the vessels. Finally, in the anisotropic case, two distinct vessels are visible in the root of the vessel network. In the isotropic case, these vessels appeared to be combined. It is clear that for this data set, it helps to take anisotropy into account.

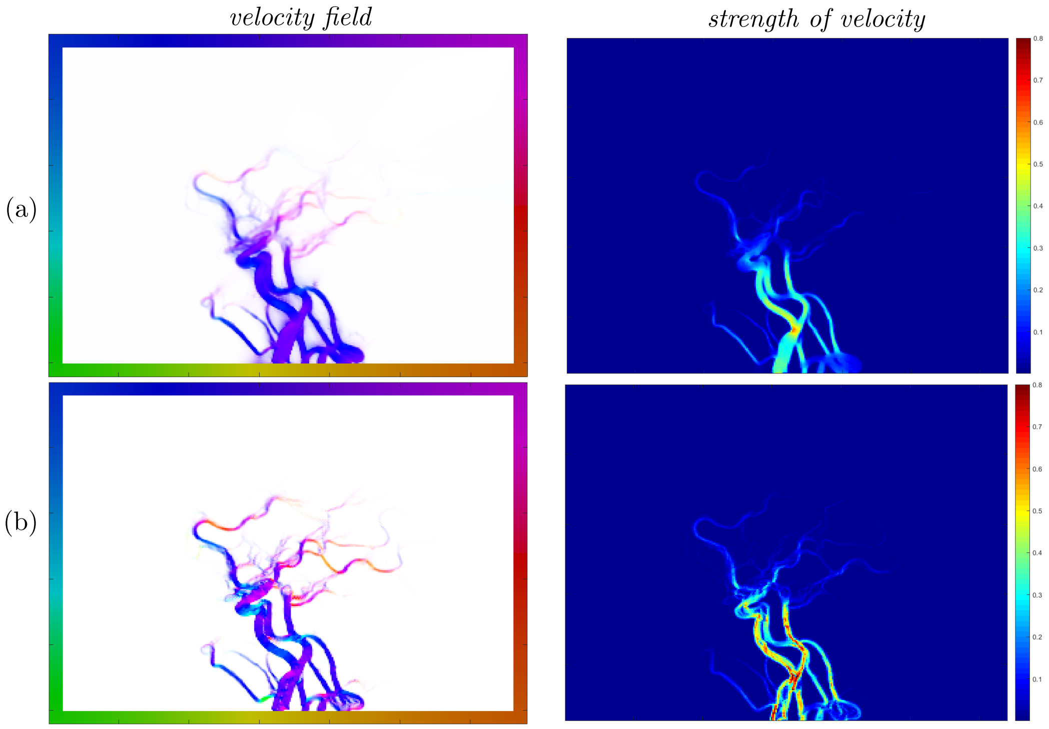


Figure 20: Velocity field and absolute value of the velocity for the flow in a human brain: (a) isotropic flow; (b) anisotropic flow.

7.3 Joint model

Algorithm 3 will be applied to the synthetic data set of flow in a pipe (Figure 11). The best result of model 2, without using an a-priori known segmentation, was obtained by applying a rigidity regulariser, which gave the result as in Figure 11e. As noted in section 7.1.2, the a-priori segmentation could be approximated if we would threshold the result at $|v| = 0.3$. For the initialisation of z in our joint model, we use the results of thresholding at $|v| = 0.2, 0.3, 0.4$. This way, we can see the effect of different threshold values on the results. We initialise z to 0 or 1 if the value of $|v|$ at that point is lower or higher respectively than the threshold value. For the other variables, the same initialisation as in chapter 7 has been taken. This means that we only use the former result for the initialisation of z, not for the other variables.

In Figure 21, the results of applying algorithm 3 have been shown for different initialisations of z. In the first column, the initial values of z are shown, where white means a value of $z = 1$ and black a value of $z = 0$. In the second column, the resulting values of z are shown after applying the algorithm. In the third column, we see the velocity field v and in the last column, the absolute value $|v|$ is shown.

It can be seen that all initialisations roughly give the same resulting segmentation and velocity field. However, in the top row, we see that the resulting segmentation is slightly bigger than in

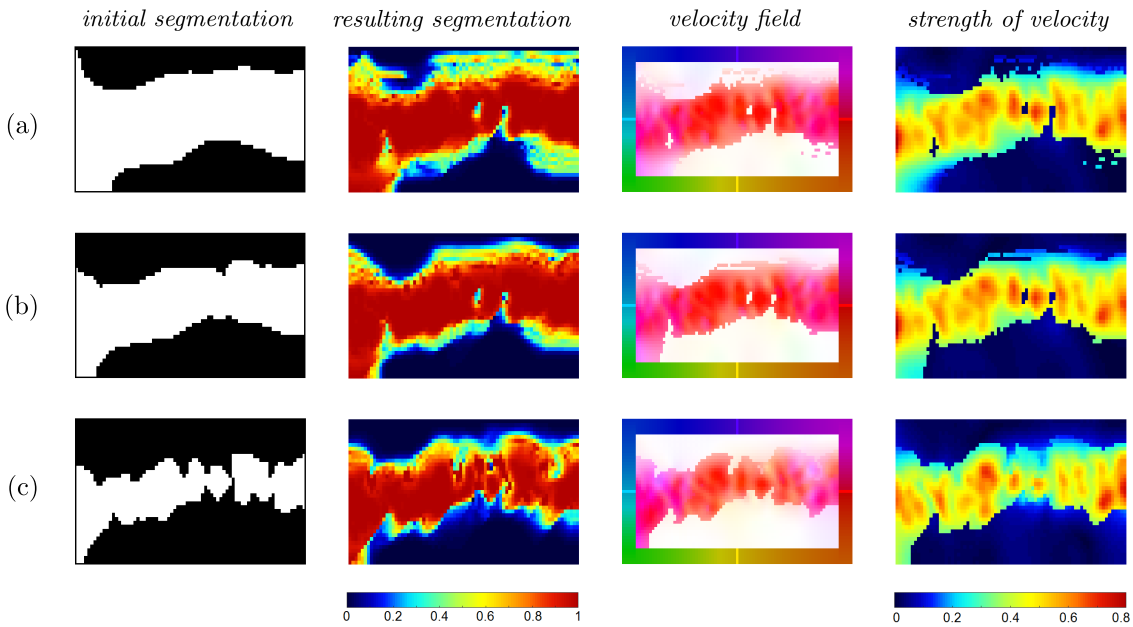


Figure 21: From left to right: the initial segmentation, the resulting segmentation z, the velocity field v and the absolute value of the velocity $|v|$ for three initialisations. Initialisations by thresholding the result of Figure 11e at (a) $|v| = 0.2$, (b) $|v| = 0.3$, (c) $|v| = 0.4$.

the middle row, which is on its turn bigger than the segmentation in the bottom row. From this we know that different initial values lead to different solutions, which is not unexpected, since the joint model is non-convex. One explanation for these different results is the following: algorithm 3 finds two velocity fields, one (v_{in}) will be strongly directed to the right, while the other (v_{out}) has a value around zero. As the algorithm is iteratively finding its solution, $v_{in/out}$ is getting closer to $\frac{m}{\rho}$ (for $\rho > 0$) and ρ and m are also getting closer to $v_{in/out}$. This means that ρ and m are adapting such that after some iterations we have $K(\rho, m, v_{in}) \ll K(\rho, m, v_{out})$ for points inside the initial segmentation and vice versa. Hence the value of z will not change any more in these points and the resulting segmentation is then obtained. Therefore, it is important that the initial segmentation is already quite good, since it will not easily change completely during the execution of the algorithm.

This is a big difference from regular Mumford-Shah models: there the underlying image intensity is fixed, while one searches for a smooth image close to the original image. In our case, the underlying variables (ρ and m) are varying themselves, causing the non-convexity of the problem.

8 Summary and Outlook

In this thesis, we combined isotropic and anisotropic optimal transport models with segmentation models for the interpolation and velocity estimation of flow in vascular structures. For the problem, a convex variational method was used and primal-dual proximal splitting algorithms were implemented.

We started by giving the motivation for this research. A description of what vascular structures look like was given and some explicit examples were shown. Next, we derived an isotropic optimal transport model with the L^2-Wasserstein distance as a cost functional. Both a variant in which inflow and outflow are prohibited and a variant in which inflow and outflow are allowed were developed. The isotropic model was extended to an anisotropic model by changing the transport cost. A second model was introduced in which the first model was coupled with a regularisation of the velocity. Two of the proposed regularisers are fluid dynamics constraints, namely an incompressibility and a rigidity regulariser. A third regulariser is an anisotropic one, that smooths the velocity field in a specific direction. In order to get this direction, we explored both an edge detecting and a coherence enhancing diffusion tensor. For the second (coupled) model, we showed theoretical existence of a minimiser, where any of the regularisers may be used. Then we gave an overview of existing segmentation methods, followed by an explanation on how to use segmentation within our framework. This includes the use of a binary segmentation, a weak segmentation and the practical use of anisotropy, which can be seen as some sort of segmentation. After that, a first possibility for a joint model was explored. The derivation of a joint model revealed the difficulties of modelling the optimal transport and segmentation jointly: the problem is non-convex and therefore difficult to solve. It is not clear if a minimiser for our proposed joint model exists.

In the next part, we explained proximal splitting and defined the proximal operators needed for the implementation. The two optimal transport models and the joint segmentation-optimal transport model were implemented in the PDHGM-algorithm in MATLAB. Because many algorithms make use of proximal splitting methods, the implementation in a different algorithm can be done without much extra work. We chose to use PDHGM, because it showed promising results in [36]. For our second (coupled) model and our joint model, we used an alternating minimisation procedure. For the first model, we used a staggered scheme to prevent the appearance of checkerboard instabilities. Convergence plots of both optimal transport models showed that stable results can be obtained. Moreover, the convergence plot of the second model revealed clearly how the alternating algorithm works and what its disadvantages are. The minimum is obtained after some iterations in which the energy functional and the regulariser first find their own minimum. After this, the coupling makes sure that a new minimum is found, while simultaneously the variables are close to each other.

Finally, all the algorithms were applied on realistic and synthetic data sets. In the data set of two moving Gaussians, as well as in the data set of a moving white blood cell, inflow or outflow was prohibited. We wanted to see if using anisotropy, segmentation or regularisers would help us to obtain a result that is expected or corresponds to the ground truth. We come to the conclusion that using an incompressibility or rigidity regulariser helps, where the latter gives the most promising results. This is especially the case in combination with an a-priori known segmentation. In the case of the moving white blood cell, using anisotropy showed a similar result as using rigidity in combination with a segmentation. A clear advantage to the sole use of anisotropy is that the first model can be applied. This means that a convex minimisation problem has to be solved and thus the minimum is a global one.

In the other data sets inflow and outflow was allowed. It was shown that our models could cope with the inflow that was expected in the real data set of flow in the human brain. Furthermore, we saw that anisotropy helped when dealing with very elongated directed structures: we could model the right flow in the example of the human brain by applying the anisotropic model. In the other examples with in-/outflow, anisotropy did not have the effect we hoped for: although the flow is directed, it can still go in two exactly opposite directions. It was shown that the use of a rigidity regulariser helps in cases where the direction of the flow is uniform within a large region and when there is not much background density in the image. If a segmentation is not available, the rigidity penalisation of the flow at the boundary of the vascular structure is rather high. Because of this, the extra penalisation of a change of flow direction inside the vessel does not have a strong effect on the energy to be minimised. However, if a segmentation is available, using a rigidity regulariser gives a better result, with a velocity estimation that is somewhat smoother.

From the joint segmentation-optimal transport model, we can not draw a strong conclusion. It seems that our proposed joint model depends much on the initialisation of all variables. Moreover, it is not clear how the alternating algorithms are reacting on each other and on a different choice of parameters. In the joint model, the segmentation is done based on the velocity estimation, which is a variable that was obtained in an indirect way via the coupling of the second model. It would be better if the segmentation could be done by using the density and momentum as input variables, which are the only variables needed for the first optimal transport model. Despite the shortcomings of the joint model, we have obtained some good results by first applying the second model and using these results as input for the joint model.

The final conclusion of this work is that it is beneficial to include segmentation, anisotropy and fluid dynamics constraints in the optimal transport model. Without any of them, one can not expect to get a reasonable interpolation and velocity estimation for image sequences of vascular structures. When these components are included however, we can find solutions to the problems that were the motivation for this research.

One of the tasks for further research is to have a closer look at the question which model has to be chosen for which problem. Combinations of anisotropy, segmentation and fluid dynamics constraints should certainly be taken into account to make use of the strengths of multiple models. Moreover, an analysis of the choice of model parameters could be considered, since it is not clear how the parameters depend on e.g. the type of image or image size. Also, one could experiment with the use of adaptive paramters for PDHGM [12]. Throughout this work, we have not considered the handling of noise in the model. Many images contain noise, which can cause a violation of mass conservation. Moreover, it can cause a velocity estimation with very strong but sparse velocities, because the model tries to deal with noisy parts that occur at different places in the initial and final density. In [28], Maas *et al.* have proposed a model in which the mass conservation does not strictly have to be attained. One could try to combine their model with the ones described in this thesis. Another interesting direction is to investigate the use of other cost functionals than the L^2-Wasserstein distance. Some thoughts and explanations on the use of L^p-Wasserstein distances have been given in appendix A. A disadvantage of using model 2 instead of model 1 was that the minimisation problem was not convex anymore, resulting in a problem that is more difficult to solve. One could overcome this by not using the velocity variable, but design a regulariser based on only the density and momentum variable instead. This way, the coupling functional is not needed and hopefully, the minimisation problem stays convex. Finally, we think that the development of a joint segmentation-optimal transport model is very promising. The current joint model is a first step in this direction, but further research definitely has to be conducted. Using a convex version of model 2 without the coupling functional would be an important step in this direction, since then this model could be used as a component in the joint model. From an application-point of view, we think that preprocessing the images by using some kind of background subtraction would help to get a proper velocity estimation, as is explained in more detail in appendix C.

References

[1] L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[2] H. Attouch, J. Boltehttp, P. Redont, A. Soubeyran, H. Attouch, J. Boltehttp, P. Redont, A. Soubeyran, and P. Redont. Mathematics of Operations Research Proximal Alternating Minimization and Projection Methods for Nonconvex Problems : An Approach Based on the Kurdyka-Lojasiewicz Inequality Proximal Alternating Minimization and Projection Methods for Nonconvex Problems : An Approach Based on the Kurdyka-Łojasiewicz Inequality. (August 2016), 2010.

[3] H. Attouch, P. Redont, and A. Soubeyran. A new class of alternating proximal minimization algorithms with costs-to-move. *SIAM Journal on Optimization*, 18(3):1061–1081, 2007.

[4] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing Partial Differential Equations and the Calculus of Variations*. 2000.

[5] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[6] J. D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[7] B. Berkels. An unconstrained multiphase thresholding approach for image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5567 LNCS:26–37, 2009.

[8] Y. E. Boink. Segmentation of blood vessel network from video images using Fast Marching methods. internship report, University of Twente, 2015.

[9] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

[10] V. Caselles and R. O. N. Kimmel. Geodesic Active Contours. 22(1):61–79, 1997.

[11] B. Cengiz. The Dual of the Bochner Space Lp(mu,E) for mu. 22:343–348, 1998.

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

[13] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. Technical report, SIAM journal on applied mathematics, 2006.

[14] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb 2001.

[15] J. Dolbeault, B. Nazaret, and G. Savaré. A new class of transport distances between measures. *Calculus of Variations and Partial Differential Equations*, 34(2):193–231, 2009.

[16] M. Droske and M. Rumpf. Multi Scale Joint Segmentation and Registration of Image Morphology. 1917:1–14, 1917.

[17] E. Esser, X. Zhang, and T. F. Chan. A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.

[18] E. Feireisl and A. Novotnỳ. *Singular Limits in Thermodynamics of Viscous Fluids*. Advances in Mathematical Fluid Mechanics. Birkhäuser Basel, 2009.

[19] S. Heldmann, T. Polzin, A. Derksen, and B. Berkels. *An image registration framework for sliding motion with piecewise smooth deformations*, pages 335–347. Springer International Publishing, Cham, 2015.

[20] B. K. Horn and B. G. Schunck. Determining optical flow. 1980.

[21] R. Hug, E. Maitre, and N. Papadakis. Multi-physics Optimal Transportation and Image Interpolation . 2014.

[22] L. V. Kantorovich. On the translocation of masses. *Dokl. Akad. Nauk SSSR 37*, pages 194–201, 1942.

[23] L. V. Kantorovich. On a problem of Monge. *Uspekhi Mat. Nauk 3*, (2):225–226, 1948.

[24] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[25] P. Kundu, I. Cohen, and D. Dowling. *Fluid Mechanics*. Academic Press, 2012.

[26] C. Le Guyader and L. A. Vese. A combined segmentation and registration framework with a nonlinear elasticity smoother. *Computer Vision and Image Understanding*, 115(12):1689–1709, 2011.

[27] J. Maas, M. Rumpf, C. Schönlieb, and S. Simon. A generalized model for optimal transport of images including dissipation and density modulation. *arXiv preprint arXiv:1504.01988*, (3):1–24, 2015.

[28] J. Maas, M. Rumpf, and S. Simon. Generalized optimal transport with singular sources. 2016.

[29] E. Mémin and P. Pérez. Dense Estimation and Object-Based Segmentation of the Optical Flow with Robust Techniques. *IEEE Transactions on Image Processing*, 7(5), 1998.

[30] G. Monge. Mémoire sur la théorie des déblais et des remblais. *De lImprimerie Royale*, 1781.

[31] A. Moussa. Some variants of the classical Aubin–Lions lemma. *Journal of Evolution Equations*, 16(1):65–93, 2016.

[32] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[33] H. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5):565–593, Sept 1986.

[34] S. Ozeré, C. Gout, and C. Le Guyader. Joint Segmentation/Registration Model by Shape Alignment via Weighted Total Variation Minimization and Nonlinear Elasticity. *SIAM Journal on Imaging Sciences*, 8(3):1981–2020, 2015.

[35] S. Ozeré and C. Le Guyader. *Nonlocal Joint Segmentation Registration Model*, pages 348–359. Springer International Publishing, Cham, 2015.

[36] N. Papadakis, G. Peyré, and E. Oudet. Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1):1–22, 2014.

[37] N. Parikh and S. Boyd. Proximal Algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[38] T. Preusser, M. Droske, C. S. Garbe, A. Telea, and M. Rumpf. A Phase Field Method for Joint Denoising, Edge Detection, and Motion Estimation in Image Sequence Processing. *SIAM Journal on Applied Mathematics*, 68(3):599–618, 2008.

[39] R. Rockafellar. *Convex Analysis.* Princeton landmarks in mathematics and physics. Princeton University Press, 1997.

[40] W. Rudin. *Functional analysis.* International Series in Pure and Applied Mathematics. McGraw-Hill Inc., New York, second edition, 1991.

[41] G. Sköllermo. A Fourier method for the numerical solution of Poisson's equation. *Mathematics of Computation*, 29(131):697–711, 1975.

[42] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, 2001.

[43] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

[44] J. Weickert. *Anisotropic diffusion in image processing*, volume 256. 1998.

Appendices

A Alternative transport costs for model 1

Instead of using the L^2-Wasserstein distance as a cost functional for the optimal transport, one could consider other functions. Papadakis *et al.* [36] considered the H^{-1}-Sobolev norm between ρ_0 and ρ_1 as a cost functional, which is explained in detail in [15]. Using only the H^{-1}-distance gives an L^2-interpolation in time between the initial and final density:

$$\rho(x,t) = (1-t)\rho_0(x) + t\rho_1(x).$$

In [36] numerical simulations have been performed for different interpolations between the L^2-Wasserstein and the H^{-1}-distance.

Here, we will consider the L^p-Wasserstein distance for general $p \geq 1$. Benamou and Brenier [6] showed that in the case of a L^2-Wasserstein distance, problem (3.4) can be rewritten into a fluid dynamics framework. In fact, this can be done for any L^p-Wasserstein distance with $p \geq 1$. The derivation is analogous from the one in [6] and gives the following result:

$$\begin{aligned} d_p(\rho_0,\rho_1)^p &= \inf_{T\in\mathcal{T}} \int_\Omega \|T(x)-x\|^p \, \rho_0(x)\,\mathrm{d}x, \\ &= \min_{(\rho,v)\in MC_{\rho v}} \frac{1}{2}\int_0^1 \int_\Omega \rho(x,t)|v(x,t)|^p \,\mathrm{d}x\,\mathrm{d}t, \end{aligned} \tag{8.1}$$
$$\text{where } MC_{\rho v} = \{(\rho,v) \mid \partial_t\rho + \operatorname{div}(\rho v) = 0,\ \rho(\cdot,0)=\rho_0,\ \rho(\cdot,1)=\rho_1,\ \rho\geq 0,\ v\cdot n|_{\partial\Omega}=0\}.$$

As in (3.5), $\mathcal{T}$ is the set of all possible mappings that map ρ_0 to ρ_1. Similar to the L^2-case, we can apply the change of variables $(\rho,m) := (\rho,\rho v)$ and obtain the following convex minimisation problem:

$$\min_{(\rho,m)} \mathcal{J}_p(\rho,m) = \min_{(\rho,m)} \int_0^1 \int_\Omega J_p(\rho,m) + \iota_{MC}(\rho,m)\,\mathrm{d}x\,\mathrm{d}t, \tag{8.2}$$
$$\text{where } J_p(\rho,m) = \begin{cases} \frac{|m|^p}{2\rho^{p-1}} & \text{if } \rho > 0, \\ 0 & \text{if } (\rho,m) = (0,0), \\ \infty & \text{otherwise.} \end{cases}$$
$$\text{and } MC = \{(\rho,m) \mid \partial_t\rho + \operatorname{div}(m) = 0,\ \rho(\cdot,0)=\rho_0,\ \rho(\cdot,1)=\rho_1,\ m\cdot n|_{\partial\Omega}=0,\ \rho\geq 0\}.$$

Note that for $p > 1$, we have a strict convexity, while this is not the case for $p = 1$.

In table 2, an overview of the effects of different cost functionals on the functional $J(\rho,m)$ have been shown.

We will now illustrate the effects of the different cost functionals by looking at two simple examples. In the first example, we consider the spatial domain $[0,1] \subset \mathbb{R}$ with the following initial and final density:

$$\begin{aligned} \rho_0(x) &= \begin{cases} 1 \text{ for } \frac{1}{5} \leq x \leq \frac{2}{5}, \\ 0 \text{ else.} \end{cases} \\ \rho_1(x) &= \begin{cases} 1 \text{ for } \frac{3}{5} \leq x \leq \frac{4}{5}, \\ 0 \text{ else.} \end{cases} \end{aligned} \tag{8.3}$$

In figure 22, the optimal transports for the L^2-, L^1- and H^{-1}-cost functionals are shown. In the L^2-case, we obtain a solution with a constant velocity and corresponding density. Because of the

L^2-Wasserstein	$\rho\lvert v\rvert^2$	$\frac{\lvert m\rvert^2}{\rho}$
L^1-Wasserstein	$\rho\lvert v\rvert$	$\lvert m\rvert$
L^p-Wasserstein for $p \geq 1$	$\rho\lvert v\rvert^p$	$\frac{\lvert m\rvert^p}{\rho^{p-1}}$
H^{-1}-Sobolev	$\rho^2\lvert v\rvert^2$	$\lvert m\rvert^2$
Interpolated L^2/H^{-1}	$\rho^{2-p}\lvert v\rvert^2$	$\frac{\lvert m\rvert^2}{\rho^p}$

Table 2: Overview of the effect of different cost functionals on the functional $J(\rho, m)$.

quadratic cost in v in (8.1) when $p = 2$, it is favourable to have a constant low velocity instead of a very high velocity for a small amount of time. In the L^1-case, minimisers are not necessarily unique, because the convexity in (8.2) is not a strict one. In figure 22b, one of the minimisers of the L^1-problem is shown. This solution shows a very rapid transport, which is sparse in time. However, one has to realise that this this solution is not unique. In fact, the L^2-solution is also a solution of the L^1-problem: from (8.1), we see that the cost of v is linear. This means that both a constant low velocity and a very high velocity for a small amount of time give the same (minimal) cost. In the H^{-1}-case, we see an L^2-interpolation in time between the initial and final density. This interpolation is shown in figure 22c.

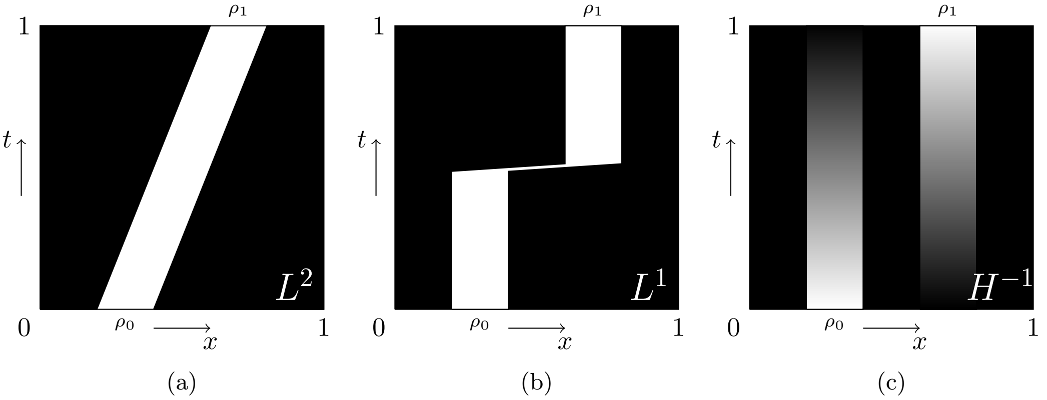


Figure 22: Solutions to the minimisation problem with ρ_0 and ρ_1 as defined in (8.3) for different cost functionals. (a) For the L^2-cost, we obtain a gradual translation in space, (b) for the L^1-cost, we obtain a fast translation in a short period of time, (c) for the H^{-1}-cost, we obtain an interpolation in time between ρ_0 and ρ_1.

In the second example, we consider the movement of two circles in the domain $[0, 3] \times [0, 7] \subset \mathbb{R}^2$. The initial density has value one in a circular region around the points $(\frac{1}{2}, \frac{1}{2})$ and $(2\frac{1}{2}, 3\frac{1}{2})$, which is shown in figure 23a. The final density is shown in figure 23b. It has value one in a circular region around the points $(2\frac{1}{2}, 3\frac{1}{2})$ and $(\frac{1}{2}, 6\frac{1}{2})$.

As shown in Figures 23c and 23d, The L^2-transport is very different from the L^1-transport. In the L^2-case, the circle in the bottom left is replacing the circle on the right and the circle on the right is transported to the top left. This can be explained by the fact that the velocity has a square in the cost functional and it thus is cheaper to move two densities with a moderate velocity instead

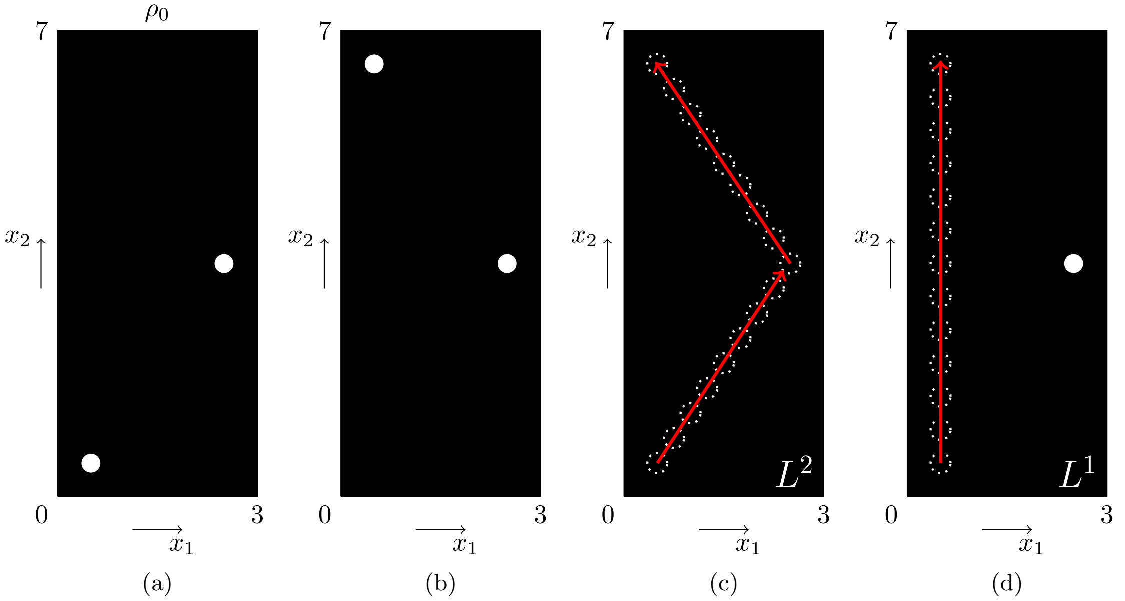


Figure 23: (a) initial density; (b) final density; (c) interpolation for the L^2-Wasserstein distance where both circles are translated over a small distance; (d) interpolation for the L^1-Wasserstein distance where only one circle is translated over a large distance.

of moving one with a high velocity. The difference in cost functional is calculated as

$$\int_0^1 \int_\Omega \rho(x,t)|v_c(x,t)|^2 \, \mathrm{d}x \, \mathrm{d}t = 13 \int_\Omega \rho_0 \, \mathrm{d}x < 18 \int_\Omega \rho_0 \, \mathrm{d}x = \int_0^1 \int_\Omega \rho(x,t)|v_d(x,t)|^2 \, \mathrm{d}x \, \mathrm{d}t,$$

where $|v_c| = \sqrt{13}$ is the velocity belonging to the transport in Figure 23c and $|v_d| = 6$ the one belonging to Figure 23d.

In the L^1-case, the circle in the bottom left is transported to the top left, whereas the circle on the right stays in place. The cost for one fast transport is in this case lower than two transports with moderate velocity:

$$\int_0^1 \int_\Omega \rho(x,t)|v_c(x,t)| \, \mathrm{d}x \, \mathrm{d}t = \sqrt{13} \int_\Omega \rho_0 \, \mathrm{d}x > 3 \int_\Omega \rho_0 \, \mathrm{d}x = \int_0^1 \int_\Omega \rho(x,t)|v_d(x,t)|^2 \, \mathrm{d}x \, \mathrm{d}t.$$

These examples show that changing the L^2-Wasserstein distance to an L^1-Wasserstein has the effect that sparsity in the transport is allowed. In the first example, we have seen the sparsity in time: it is allowed to have a high velocity in a short amount of time. In the second example, we have seen sparsity in space: one transport along a long distance may give a lower cost than multiple transports along a short distance.

Although it is interesting to see the effects of an L^p-transport cost (for $p \to 1$), it is not easy to implement via proximal splitting methods. In the L^2-case, we had a proximal step for $J(\rho, m)$ which could be calculated explicitly (see (6.10) and (6.11)). However, for $p > 1$, we obtain the following equations:

$$(\rho, m) = \mathrm{Prox}_{\gamma J_p}(\tilde{\rho}, \tilde{m}) \Rightarrow \begin{cases} \frac{\tau p}{2\rho} m|m|^{p-2} + m - \tilde{m} = 0 \\ \rho^{p+1} - \tilde{\rho}\rho^p - \frac{\tau}{2}|m|^p. \end{cases} \tag{8.4}$$

We are interested in the case where $1 < p < 2$. It is clear that the equation for m can not be solved explicitly for $1 < p < 2$. Moreover, we see that the equation for ρ is no polynomial anymore. If we take $p \in \mathbb{Q}$, we can rewrite the equation in a higher order polynomial. This polynomial can not be solved explicitly, but solving it with Newton's method would give a result. It is clear that solving both equations with high precision is computationally very expensive, since both will have to be solved in a numerical way, probably in an iterative way. Therefore, the implementation of other transport costs remains an open question.

B Derivation of adjoint operators in regulariser

In section 6.2, we stated the operators and their adjoints that are used in the regulariser of model 2. Here, we present the derivation of the adjoints, given the operators and the function spaces in which the variables are defined.

divergence

$$\begin{aligned} &v \in L^2(0,1;W^{\mathrm{div},2}(\Omega)) \quad \text{with } v\cdot n|_{\partial\Omega}=0,\\ &w \in L^2(0,1;W^{1,2}(\Omega)). \end{aligned}$$

For $H_D v = \operatorname{div}(v)$, we derive the following adjoint operator:

$$\begin{aligned} \langle H_D v, w\rangle = \langle \operatorname{div}(v), w\rangle &= \int_0^1 \int_\Omega (\operatorname{div}(v)) w \,\mathrm{d}x\,\mathrm{d}t \\ &= -\int_0^1 \int_\Omega v\cdot(\nabla w)\,\mathrm{d}x\,\mathrm{d}t + \int_0^1 \oint_{\partial\Omega} (v\cdot n) w\,\mathrm{d}s\,\mathrm{d}t \\ &= -\int_0^1 \int_\Omega v\cdot(\nabla w)\,\mathrm{d}x\,\mathrm{d}t = \langle v, -\nabla w\rangle = \langle v, H_D^* w\rangle, \end{aligned}$$

where we used integration by parts at the third equality and the fact that $v\cdot n|_{\partial\Omega}=0$ at the fourth equality. From this derivation, it is clear that

$$H_D^* w = -\nabla w.$$

rigidity

$$\begin{aligned} v &= (v_1, v_2) \in L^2(0,1;(W_0^{1,2}(\Omega))^2),\\ w &= \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}. \end{aligned}$$

For $H_R v = \dfrac{\nabla v + (\nabla v)^T}{2} = \begin{pmatrix} 2\frac{\partial v_1}{\partial x_1} & \frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1} \\ \frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1} & 2\frac{\partial v_2}{\partial x_2} \end{pmatrix}$, we derive the following adjoint operator:

$$\begin{aligned} \langle H_R v, w\rangle = \left\langle \frac{\nabla v + (\nabla v)^T}{2}, w \right\rangle &= \frac{1}{2}\int_0^1 \int_\Omega \begin{pmatrix} 2\frac{\partial v_1}{\partial x_1} & \frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1} \\ \frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1} & 2\frac{\partial v_2}{\partial x_2} \end{pmatrix} \cdot \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix} \mathrm{d}x\,\mathrm{d}t \\ &= \frac{1}{2}\int_0^1 \int_\Omega 2\frac{\partial v_1}{\partial x_1} w_1 + \frac{\partial v_1}{\partial x_2} w_2 + \frac{\partial v_2}{\partial x_1} w_2 + \frac{\partial v_1}{\partial x_2} w_3 + \frac{\partial v_2}{\partial x_1} w_3 + 2\frac{\partial v_2}{\partial x_2} w_4 \,\mathrm{d}x\,\mathrm{d}t \\ &= \frac{1}{2}\int_0^1 \int_\Omega (\nabla v_1) \begin{pmatrix} 2w_1 \\ w_2 + w_3 \end{pmatrix} + (\nabla v_2) \begin{pmatrix} w_2 + w_3 \\ 2w_4 \end{pmatrix} \mathrm{d}x\,\mathrm{d}t \\ &= -\frac{1}{2}\int_0^1 \int_\Omega v_1 \operatorname{div} \begin{pmatrix} 2w_1 \\ w_2 + w_3 \end{pmatrix} + v_2 \operatorname{div} \begin{pmatrix} w_2 + w_3 \\ 2w_4 \end{pmatrix} \mathrm{d}x\,\mathrm{d}t \\ &\qquad + \int_0^1 \oint_{\partial\Omega} v_1 \begin{pmatrix} 2w_1 \\ w_2 + w_3 \end{pmatrix} \cdot n + v_2 \begin{pmatrix} w_2 + w_3 \\ 2w_4 \end{pmatrix} \cdot n \,\mathrm{d}s\,\mathrm{d}t \\ &= -\frac{1}{2}\int_0^1 \int_\Omega v_1 \operatorname{div} \begin{pmatrix} 2w_1 \\ w_2 + w_3 \end{pmatrix} + v_2 \operatorname{div} \begin{pmatrix} w_2 + w_3 \\ 2w_4 \end{pmatrix} \mathrm{d}x\,\mathrm{d}t \\ &= \langle v, H_R^* w\rangle, \end{aligned}$$

where we used integration by parts in the fifth equality and the fact that $v_1|_{\partial\Omega} = v_2|_{\partial\Omega} = 0$ in the sixth equality. From this derivation, we know that

$$H_{_R}^* w = -\frac{1}{2}\left(\operatorname{div}\begin{pmatrix} 2w_1 \\ w_2 + w_3 \end{pmatrix}, \operatorname{div}\begin{pmatrix} w_2 + w_3 \\ 2w_4 \end{pmatrix}\right). \tag{8.5}$$

anisotropic

$$\begin{aligned} v &= (v_1, v_2) \in L^2(0,1;(W_0^{1,2}(\Omega))^2), \\ w &= (w_1, w_2, w_3, w_4). \end{aligned}$$

For $H_{_A} v = (U\nabla v_1, U\nabla v_2)$ with $U = \begin{pmatrix} U_1 & U_2 \\ U_3 & U_4 \end{pmatrix}$, we derive the following adjoint operator:

$$\begin{aligned}
\langle H_{_A} v, w\rangle &= \langle (U\nabla v_1, U\nabla v_2), w\rangle \\
&= \left\langle \left(U_1\frac{\partial v_1}{\partial x_1} + U_2\frac{\partial v_1}{\partial x_2},\ U_3\frac{\partial v_1}{\partial x_1} + U_4\frac{\partial v_1}{\partial x_2},\ U_1\frac{\partial v_2}{\partial x_1} + U_2\frac{\partial v_2}{\partial x_2},\ U_3\frac{\partial v_2}{\partial x_1} + U_4\frac{\partial v_2}{\partial x_2}\right), (w_1, w_2, w_3, w_4)\right\rangle \\
&= \int_0^1 \int_\Omega \frac{\partial v_1}{\partial x_1}(U_1 w_1 + U_3 w_2) + \frac{\partial v_1}{\partial x_2}(U_2 w_1 + U_4 w_2) \\
&\qquad + \frac{\partial v_2}{\partial x_1}(U_1 w_3 + U_3 w_4) + \frac{\partial v_2}{\partial x_2}(U_2 w_3 + U_4 w_4)\,\mathrm{d}x\,\mathrm{d}t \\
&= \int_0^1 \int_\Omega \nabla v_1 \cdot \begin{pmatrix} U_1 w_1 + U_3 w_2 \\ U_2 w_1 + U_4 w_2 \end{pmatrix} + \nabla v_2 \cdot \begin{pmatrix} U_1 w_3 + U_3 w_4 \\ U_2 w_3 + U_4 w_4 \end{pmatrix} \mathrm{d}x\,\mathrm{d}t \\
&= -\int_0^1 \int_\Omega v_1 \operatorname{div}\begin{pmatrix} U_1 w_1 + U_3 w_2 \\ U_2 w_1 + U_4 w_2 \end{pmatrix} + v_2 \operatorname{div}\begin{pmatrix} U_1 w_3 + U_3 w_4 \\ U_2 w_3 + U_4 w_4 \end{pmatrix} \mathrm{d}x\,\mathrm{d}t \\
&\qquad + \int_0^1 \oint_{\partial\Omega} v_1 \begin{pmatrix} U_1 w_1 + U_3 w_2 \\ U_2 w_1 + U_4 w_2 \end{pmatrix} \cdot n + v_2 \begin{pmatrix} U_1 w_3 + U_3 w_4 \\ U_2 w_3 + U_4 w_4 \end{pmatrix} \cdot n\,\mathrm{d}s\,\mathrm{d}t \\
&= -\int_0^1 \int_\Omega v_1 \operatorname{div}\begin{pmatrix} U_1 w_1 + U_3 w_2 \\ U_2 w_1 + U_4 w_2 \end{pmatrix} + v_2 \operatorname{div}\begin{pmatrix} U_1 w_3 + U_3 w_4 \\ U_2 w_3 + U_4 w_4 \end{pmatrix} \mathrm{d}x\,\mathrm{d}t \\
&= \langle v, H_{_A}^* w\rangle,
\end{aligned}$$

where we used integration by parts in the fifth equality and the fact that $v_1|_{\partial\Omega} = v_2|_{\partial\Omega} = 0$ in the sixth equality. From this derivation, we know that

$$H_{_A}^* w = -\left(\operatorname{div}\begin{pmatrix} U_1 w_1 + U_3 w_2 \\ U_2 w_1 + U_4 w_2 \end{pmatrix}, \operatorname{div}\begin{pmatrix} U_1 w_3 + U_3 w_4 \\ U_2 w_3 + U_4 w_4 \end{pmatrix}\right).$$

C Why a ground density causes a lower velocity estimation

As explained in section 7.1.2, having a ‘ground density’ along the path that is followed in the transport can cause a lower velocity estimation than expected. In this appendix, we give an easy example where the effect is immediately visible. Consider again the one-dimensional example of a moving block: the initial density ρ_1 is a rectangular function with value 1 between $x = 0$ and $x = 1$. The final density ρ_1 has the same shape, but is moved to the right, such that it has value 1 between $x = 2$ and $x = 3$.

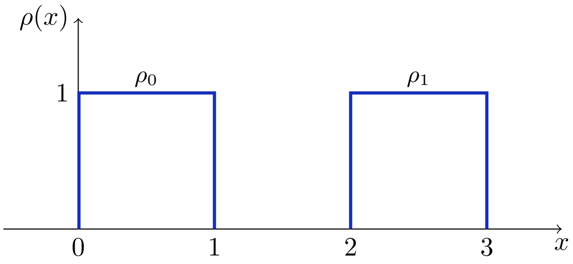


Figure 24: Rectangular initial and final density without a ground density.

The minimal transport cost is obtained by the constant velocity field $v(x,t) = 2$.

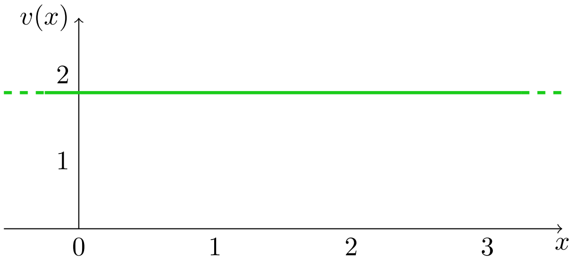


Figure 25: Constant velocity transporting ρ_0 to ρ_1.

Note that this is not the only velocity field yielding a minimal transport cost. In fact, the velocity may take any finite value at points where there is no density. From (3.5), we calculate the transport cost:

$$\frac{1}{2}\int_0^1 \int_\Omega \rho(x,t)|v(x,t)|^2 \,\mathrm{d}x\,\mathrm{d}t = 2. \tag{8.6}$$

Now we change this density to the one shown in Figure 26. Here we have a ‘ground density’ of $\frac{1}{2}$ and a ‘top density’ of 1.

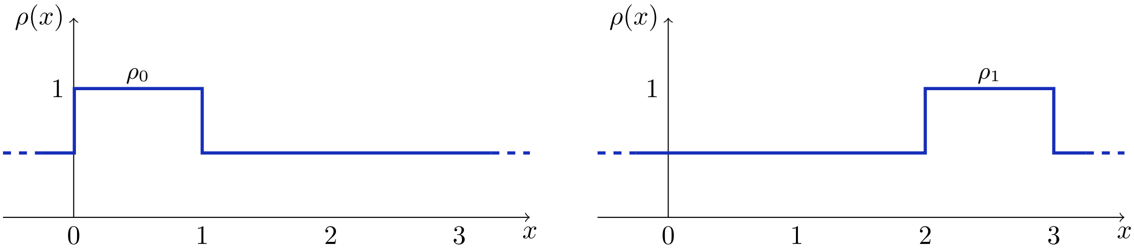


Figure 26: Rectangular initial and final density with a ground density.

Obviously, our velocity $v(x,t) = 2$ will still be one that transports ρ_0 to ρ_1. We calculate the transport cost, where we only take the domain between $x = 0$ and $x = 3$ into account:

$$\frac{1}{2}\int_0^1\int_0^3 \rho(x,t)|v(x,t)|^2\,\mathrm{d}x\,\mathrm{d}t = 4. \tag{8.7}$$

By applying our first model to this example, we obtain the interpolation and velocity field as in Figure 27:

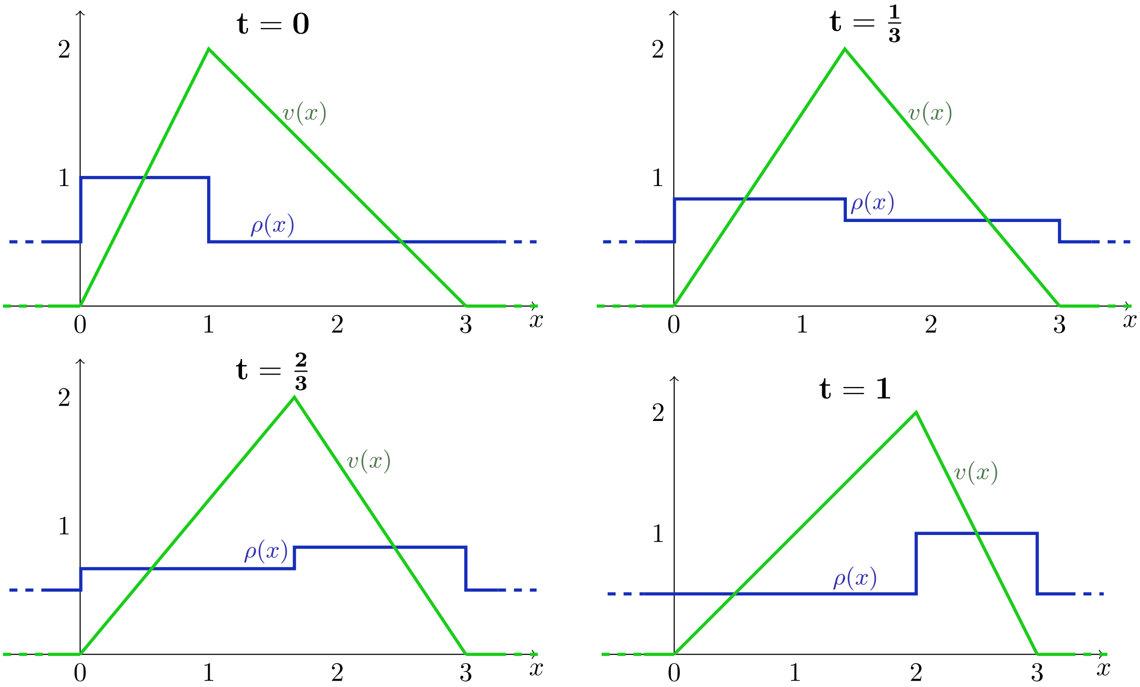


Figure 27: Optimal interpolation and velocity field at several points in time.

The functions of the density and velocity are the following:

$$\rho(x,t) = \begin{cases} 1 - \frac{t}{2} & \text{for } 0 \le x \le t+1, \\ \frac{1}{2} + \frac{t}{2} & \text{for } t+1 \le x \le 3, . \\ \frac{1}{2} & \text{else.} \end{cases} \tag{8.8}$$

$$v(x,t) = \begin{cases} \frac{2x}{t+1} & \text{for } 0 \leq x \leq t+1, \\ \frac{6-2x}{2-t} & \text{for } t+1 \leq x \leq 3, . \\ 0 & \text{else.} \end{cases} \tag{8.9}$$

By using (8.8) and (8.9), we calculate the transport cost

$$\begin{aligned} &\frac{1}{2}\int_0^1 \int_0^3 \rho(x,t)|v(x,t)|^2 \,\mathrm{d}x\,\mathrm{d}t \\ =&\frac{1}{2}\int_0^1 \int_0^{t+1} \left(1-\frac{t}{2}\right)\left(\frac{2x}{t+1}\right)^2 \mathrm{d}x\,\mathrm{d}t + \frac{1}{2}\int_0^1 \int_{t+1}^3 \left(\frac{1}{2}+\frac{t}{2}\right)\left(\frac{6-2x}{2-r}\right)^2 \mathrm{d}x\,\mathrm{d}t = \frac{13}{9}, \end{aligned}$$

which is much lower than the transport cost calculated in (8.6) and even lower than the transport cost calculated in (8.7).

As is clear from Figure 27, there is only transport between $x = 0$ and $x = 3$, while the rest of the density is not transported. The part of ρ_0 that lies between $x = 0$ and $x = 1$ is decompressed and takes the place of ρ_1 between $x = 0$ and $x = 2$. The part of ρ_0 that lies between $x = 1$ and $x = 3$ is compressed and takes the place of ρ_1 between $x = 2$ and $x = 3$. This transport costs less energy than a complete translational transport and is only possible if a ‘ground density’ is available. In some applications, this behaviour is not expected: for example in the application of blood flowing in a blood vessel. To overcome this problem, one could try to first do a clever kind of background subtraction, before the model will be applied.