Penalized Splitting in Modelling Structures of Live Tissues

by
Caroline Myklebust

Master of Science Thesis in
Applied and Computational Mathematics

Department of Mathematics
University of Bergen

June 2020

Abstract

This thesis presents an addition to current mathematical models of vascular tissue. The hypothesis is that by penalizing splitting points in a structure, the optimal solution will include fewer splittings. As this becomes a matter of optimal resource allocation, the theory of Optimal Transport is presented. This is followed by describing a recent model of the structures of interest. To gain a deeper understanding of how to classify splitting points, their properties are studied both dynamically and using image processing. The results are two proposed PDEs that incorporate the splitting penalty properties.

Experiments show that both of the derived models can penalize splitting points. However, they both focus on removing the smallest splitting points of the structure. A closer examination yields that both models have successfully targeted some of the desired points.

Acknowledgements

Firstly, I would like to thank my friends and family for all the love and support through this time. I have greatly appreciated all the encouraging messages and well wishes. These past years have been a journey both personally and academically, and I am thankful for the opportunity and the possibilities ahead.

Secondly, I would like to give gratitude to my advisors Jan Martin Nordbotten and Erik Hanson. On this journey you have shown enthusiasm in the field of work, you have advised me with patience even when asked the same questions twice and shown trust in me as a student and mathematician to be able to execute the task at hand. I would not have been able to complete this thesis without your advice and guidance.

I would also like to thank the University of Bergen, and more specifically the Department of Mathematics, for teaching me everything I know over the past five years. Many great professors have motivated me to stay curious through the years, and I am very grateful for that.

Contents

Notation

- **Groups**

$\mathbb{R}$: The set of all real, additive numbers

$\mathbb{C}$: The set of complex numbers

$\mathcal{C}_b(X)$: The set of all continuous and bounded functions in the set X.

$\mathcal{T}(f^+, f^-)$: The set of all transport maps from f^+ to f^-.

$\Pi(f^+, f^-)$: The set of all transport plans from f^+ to f^-.

- **Operators**

∇: The gradient operator

$\nabla\cdot$: The divergence operator

Δ: The Laplacian

D^2u: The Hessian of a scalar function u.

I: The identity matrix

det: The determinant of a matrix.

tr: The trace operator of a matrix.

Id: The identity map

- **Properties**

max: The number, x, in a set, Z, s.t. all other numbers $z_i \leq x$ for $z_i \in Z$.

min: The number, y, in a set, Z, s.t. all other numbers, $z_i \geq y$ for $z_i \in Z$.

sup: The smallest upper bound of a set.

inf: The largest lower bound of a set.

- **Relations**

$\approx$: Approximately equal to

$>$: Greater than

$\geq$: Greater than or equal to

$:=$: Defined as

Abbreviations

OT - Optimal Transport

OTP - Optimal Transport Problem

MK - Monge Kantorovich

DMK - Dynamic Monge Kantorovich

BTP - Branched Transport Problem

OR - Operational Research

FD - Finite Differences

FV - Finite Volumes

FE - Finite Elements

Chapter 1

Introduction

1.1 Motivation

Many things act as optimal as possible. Examples are how water droplets form together, how root systems optimally gather the most nutrients for a plant, road networks, and how the blood is transported to support all vital organs as efficiently as possible. The thing all of these have in common is a pre-determined pattern. We are in this thesis interested in describing these patterns by mathematics.

First, look at how biology often can be represented by fractals, which are repetitive patterns. See the example in fig. 1.1. By knowing more about how these structures can be described mathematically, we will know more about their patterns.

That is why we will begin by introducing Optimal Transport, which is the field of studying optimal resource allocation. By building a model for these structures using the concepts of Optimal Transport, our goal is to mimic the real optimal structures by a pre-determined pattern from the model.

The main motive is to model vascular tissue. However, it can be noted that the mathematics closely resemble models of other structures shown to exhibit tendencies based on cost and energy efficiency. This is why we can relate the vascular models, to models of road networks, etc.

Existing models describing such patterns end up with a lot of small branches, see fig. 1.2. It is hard to explain that these small branches make it an optimal structure. By comparison, imagine a driveway placed directly by a highway. The large difference in speed will make it hard to match both scenarios. The more common structure is to have a driveway at the end of a smaller road from the highway. In this scenario, the speed slowly slows down along the way. It can also be thought that veins follow

(a) 'Fluitenkruidbloemen' by Rasbak under the GFDR license.

(b) Generated in MATLAB.

Figure 1.1: A figure of how biology can be mimicked by repetitive patterns. Figure a) is an image of a Wild Chervil plant and figure b) is generated in MATLAB by programming a decreasing figure with random angles and random branches.

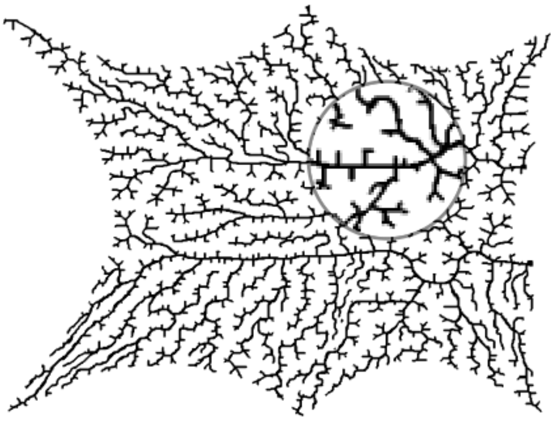


Figure 1.2: A model of the veins on a frog tongue. Notice how the structure has a lot of small branches directly from the main artery.

this structure, which is the reason why we have the main arteries transporting blood to major body parts, and smaller arteries distributing to the whole body.

In this thesis, we wish to convey these properties by investigating how to penalize branches who break with this idea. We will regard the transportation system as a splitting structure, and it will be referred to as a splitting structure throughout. In this thesis, we will work on creating a continuous splitting penalizing function whose goal is to add an extra cost to the ”driveways” placed by the ”highways” such that an optimal allocation model would not want to include these, as they would be expensive to build. The idea is that including these properties in the existing models will yield structures without the additional, and unnatural branches.

1.2 Chapter overview

This section gives a brief overview of each chapter.

Chapter 1 – Introduction This chapter.

Chapter 2 – Optimal Transport Presents the theory of Optimal Transport.

Chapter 3 – Building the Mathematical Model Builds the most recent mathematical model, the eDMK, for structures of interest.

Chapter 4 – Describing Splitting Structure Derives an original continuous functional to describe the added cost of splitting.

Chapter 5 – Expanded Models and Solution Strategies Introduces a second original splitting penalty strategy based on graph structure dynamics, then sets up the full proposed PDEs and discusses solution strategies.

Chapter 6 – Experiments Tests the mathematical models through different numerical experiments.

Chapter 7 – Discussion Discusses the outcomes of the experiments and the limitations of the models.

Appendices There is only one appendix, namely Appendix A.

The appendix includes topics that are used and referred to throughout the thesis. It also links to great resources that provide more information about each of the topics included there.

Chapter 2

Optimal Transport

The theory of Optimal Transport was first introduced by Gaspard Monge in 1781 [Monge, 1781]. He introduced the notion of an optimal way to allocate military resources from one location to another in the most efficient way. Today this resource allocation theory has applications within many fields such as Machine Learning, Operational Research, and Economics. Throughout this chapter, we will see some examples, as well as direct our attention towards a biologically inspired formulation of Optimal Transport. This chapter is structured as follows:

First, we will start by introducing some definitions of sets and transport maps in section 2.1, as it is important to have a certain understanding of this before venturing into the Optimal Transport theory.

Section 2.2 will introduce the Monge Problem, which is the fundamental formulation for the study of Optimal Transport. Some physical examples of the Monge Problem will also be provided here, giving motivation for Kantorovich formulation presented in section 2.3.

The Dual Kantorovich formulation is introduced in sec. 2.4. It will also present the direct link between the Dual Kantorovich Problem and Linear Programming, which might be more familiar to the reader.

Section 2.6 discusses the general properties necessary in order to establish existence and uniqueness for the Optimal Transport problems.

Finally, Branched Optimal Transport is mentioned in sec. 2.7. This becomes an essential topic later in this work, and creates a prelude to the next chapter.

This chapter closely follows the presentation of Optimal Transport done in [Hamfeldt, 2019] based on the works of [Villani, 2009] and [Santambrogio, 2015]. However, the notation follows [Facca, 2017] as this will lead to fewer complications in succeeding chapters.

2.1 Definitions

Here are some important definitions when discussing the theory of Optimal Transport. Also, see [Villani, 2009] for these and more definitions used when studying Optimal Transport.

Definition 2.1 (Measures). *In Mathematical Analysis,* ***a measure*** *of a set is a way of assigning a number to each subset of the set. We can think of the measure as assigning a sense of magnitude to each subset within the set.* ⌟

A **measurable set** then directly follows from the definition above as a set to which we can apply a notion of magnitude.

Definition 2.2 (Density). *A* ***density*** *usually refers to the magnitude or quantity of a certain item within a fixed amount of space (ie. a volume or a set).* ⌟

Definition 2.3 (Transformation). *Transformation is a function that maps values from one set to another, ie.* $T : \mathbf{R}^d \to \mathbf{R}^n$ *where possibly* $d = n$*. Note: the different sets can also be more specifically defined.* ⌟

Definition 2.4 (Map). *A* ***map*** *or a* ***mapping*** *is a function describing the relationship between structures or objects.* ⌟

Definition 2.5 (Transport Map). *A* ***transport plan*** *is a map* T $T : M \to F$ *That maps a set* M *to a set* F *where* M *and* F *are one-to-one.* ⌟

2.2 The Monge Formulation

The Original Monge Problem is based on finding the optimal map T moving soil from an excavation, f^+, to an embankment, f^-, of equal volume.

Pointwise transporting all the points from f^+, onto their final state, f^-, this problem can be expressed as follows:

$$\min \int |x - T(x)| df^+(x) \tag{2.1}$$

In other words, we seek to minimize the distance weighed by the change of mass. This formulation is set up using Monges original cost function $c(x, T(x)) = |x - T(x)|$,

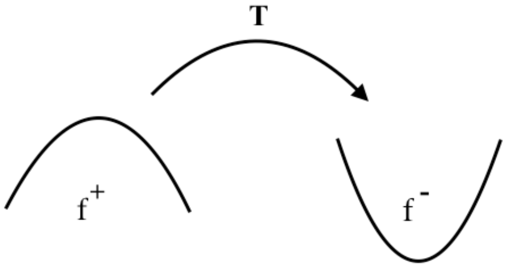


Figure 2.1: A visualizaition of the transformation made from the initial measure, f^+, to the target measure f^-.

but it can also be stated for an arbitrary cost function, $c(x, T(x))$:

$$\min \int c(x, T(x)) df^+(x) \tag{2.2}$$

Later this will be regarded using a few different costs.

Given a source measure f^+ and a target measure, f^-, then $f^+(E)$ tells us how much mass is in the set E. The **mass balance** needs to be enforced, ie. $f^+(\mathbb{R}^n) = f^-(\mathbb{R}^n)$. This ensures that the two sets are of equal volume.

We then seek a transport map $T(x)$ st. $T : X \to Y$ where X and Y are separable spaces supporting the source and target measures. Usually, we assume $X = Y = \Omega$, where Ω is an open, bounded, convex and connected domain in $\mathbb{R}^n$.

The measure $T_{\#}f^+$ is called the **pushforward** or **image measure** of f^+ through T and is defined as $T_{\#}f^+(A) = f^+(T^{-1}(A))$ for a measurable set A.

In order to ensure **mass conservation**, $T_{\#}f^+(A) = f^-(A)$ is required for all A, $A \subset Y$. This implies that $T_{\#}f^+ = f^-$, ie. all of the mass can be mapped from one measure to the other.

The set of all possible transport maps can be denoted as $\mathcal{T}(f^+, f^-)$ where:

$$\mathcal{T}(f^+, f^-) = \begin{Bmatrix} \text{Measurable map } T : X \to Y \\ \hfill \text{s.t. } T_{\#}f^+ = f^- \end{Bmatrix} \tag{2.3}$$

The properties stated above can be summarized into the following problem:

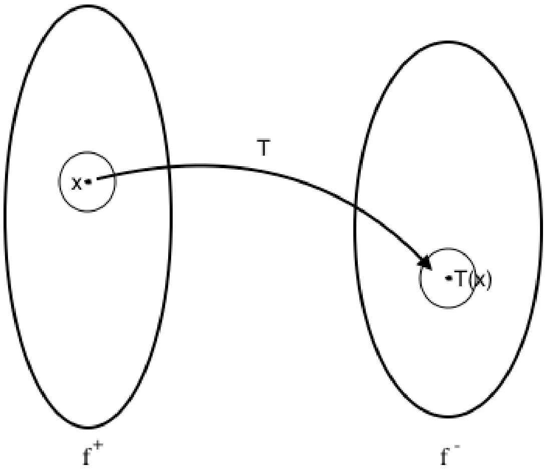


Figure 2.2: The two different measures f^+ and f^- with a transported point mass, x.

The Monge Problem *Given two nonnegative measures, f^+ and f^- on X and Y satisfying $f^+(X) = f^-(Y)$ and a cost functional $c : X \times Y \to \mathbb{R}$, find $T^* \in \mathcal{T}(f^+, f^-)$ solving*

$$\inf_{T \in \mathcal{T}} \{ \int_{\mathbf{R}^n} c(x, T(x)) df^+(x) \} \tag{2.4}$$

When discussing the wellposedness of this problem, the critera are:

- Existence
- Uniqueness
- Stability

All of these will be discussed in more detail later, as well as solution strategies to these types of problems. First recall that the problem described by Monge uses the case $c(x, y) = |x - y|$. We will now study what happens with the uniqueness and existence of an optimal solution by changing the cost function in the following example.

Example 2.6 (Book Shifting Problem). Given two books and a desired final location 1 unit length away from the initial location. There are two different ways of moving the books, one is to collectively shift them, and the second is to move one of the books two units over (see fig. 2.3). Looking at different costs, is there an optimal transport map?

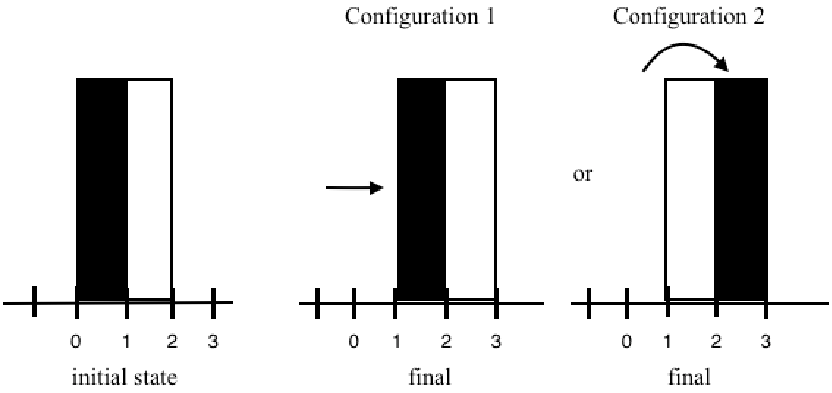


Figure 2.3: The two different configurations for shifting the books over to their desired final state.

<u>**Cost 1:**</u> $c(x, y) = |x - T(x)|$

Configuration 1: $c_1 = |5 - 4| + |6 - 5| = 2$
Configuration 2: $c_2 = |5 - 5| + |4 - 6| = 2$
For this cost, the optimal map is non-unique, as both configurations require the same cost.

<u>**Cost 1:**</u> $c(x, y) = |x - T(x)|^2$

Configuration 1: $c_1 = \frac{1}{2}|4 - 5|^2 + \frac{1}{2}|6 - 5|^2 = 1$
Configuration 2: $c_2 = \frac{1}{2}|5 - 5|^2 + \frac{1}{2}|4 - 6|^2 = 2$
Hence, using this cost function yields an optimal solution.

We see that the original Monge cost $c(x, y) = |x - y|$ does not yield a unique solution as both of the scenarios obtain the same cost, whereas it becomes more efficient to move both of the books instead of one by using the quadratic cost.

┘

Example 2.7 (Mines to Factories). Say we have a mine producing a mass 1 unit of coal to be distributed, and two factories each with capacity 0.5 (see fig. 2.4). The optimal solution would be to split the mass and distribute 0.5 units to each factory. Since the Monge formulation does not allow for this, the current problem does not have a solution. ┘

To be able to discuss a solution of problems such as the one in example 2.7, we need

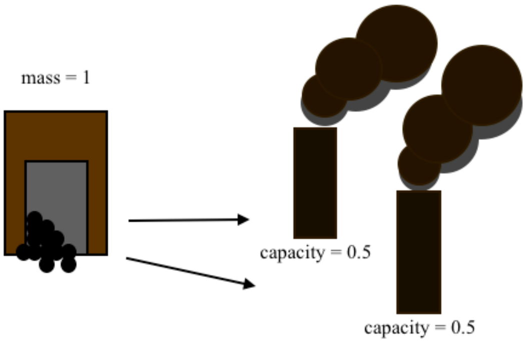


Figure 2.4: A figure representing the distribution of coal from mines to factories.

to introduce the Kantorovich relaxation.

2.3 Kantorovich Formulation

This is a generalization of the Monge formulation that allows for mass splitting. The original Kantorovich formulation is stated in [Kantorovich, 2006].

Instead of trying to find a map as in the previous section, we now seek a **Transport plan** that allows for mass splitting. This means that mass from one point can go to multiple points, or even to the continuum.

This formulation works with measures instead of densities, where the source measure is denoted f^+ and the target measure is denoted f^-. They have support on the sets X and Y respectively.

We want to know how much mass gets moved from a point x to y. This information will be stored in another measure π which needs to exist on the product space of the two variables, $X \times Y$.

If $A \subset X$, $B \subset Y$, then $\pi(A, B)$ yields how much mass is transported from A to B. The constraints that impose mass conservation will then be:

Choose some point $x \in X$. Then we require:

$$\pi(x, Y) = f^+(x) \tag{2.5}$$

This means that the mass transported from a point x to all of Y, needs to be equal to the mass of x.

More generally for a set $A \subset X$

$$\pi(A, Y) = f^+(A). \tag{2.6}$$

We say that f^+ is the **marginal** of π on X. The same can be said the other way around. Now $c(x, y)$ needs to be weighted by the amount of mass that is transported between x and y.

The Kantorovich Primal Problem *Given two non-negative finite measures, f^+ and f^- on X and Y respectively satisfying $f^+(X) = f^-(Y)$, and a given cost function $c : X \times Y \to \mathbb{R}^+$, find the optimal transport plan $\pi \in \Pi(f^+, f^-)$*

$$\inf_{\pi \in \Pi} \{ \int_{XxY} c(x, y) d\pi(x, y) \} \tag{2.7}$$

Where $\Pi(f^+, f^-)$ consists of measures whose marginals on X and Y are f^+ and f^- respectively. Minimizing the cost of going from x to y weighted by how much is moved from x to y. Notice that this formulation ends up having a linear constraint in π.

This problem is feasible (ie. well-posed w/ existing infimum) if:

- If we have a mass balance between the two measures.
- If c is bounded below and X, Y are bounded, we will have a finite infimum.

The following theorem summarizes the existence of a solution for the Kantrovich Problem:

Theorem 2.8. *Suppose $X, Y \subseteq \mathbb{R}^n$ are compact and that $c(x, y)$ is continuous, then the Kantorovich problem (eq. 2.7) has a minimum.*

The proof of this theorem is stated in full in [Santambrogio, 2015].

Most common cases where we find the Kantorovich problem:

- Discrete OT (f.ex. mines to factories, dirac masses to dirac masses etc.)
- Continuous OT (f^+ ,f^- are nice continuous measures with densities $df^+(x)$ and $df^-(x)$.)
- Semi discrete OT

2.4 The Kantorovich Dual Formulation

We will start this section by looking at an example of the discrete version of eq. 2.7. Then we will use Linear Algebra to introduce the duality and finally state the formal Dual Kantorovich Problem.

Example 2.9. (The Discrete Kantorovich Problem) [Hamfeldt, 2019] Given a finite source and target measure:

$$f^+(x) = \sum_{i=1}^{n} u_i \delta_{x_i}(x), \;\; f^-(y) = \sum_{j=1}^{m} v_j \delta_{y_j}(y), \tag{2.8}$$

π is finite dimensional, and $\pi_{i,j}$ tells us how much mass is transported from x_i to y_j.

$$\begin{aligned} \text{Constraints:} \\ \pi(x,Y) = f^+(x) \\ \pi(X,y) = f^-(y) \\ \sum_{j=1}^{m} \pi_{i,j} = f_i^+ \;, \sum i = 1^n \pi_{i,j} = f_j^- \\ \pi_{i,j} \geq 0 \\ \text{Objective function:} \\ \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} \pi_{i,j} \end{aligned} \tag{2.9}$$

The discrete optimization problem is to minimize the objective function subject to the constraints. This is a linear problem very familiar to those familiar with linear programming.

The Discrete Optimization Problem

$$\begin{aligned} \min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} \pi_{i,j} & \\ s.t. & \\ \sum_{j=1}^{m} \pi_{i,j} = f_i^+ & \qquad i = 1, \dots, n \\ \sum_{i=1}^{n} \pi_{i,j} = f_j^- & \qquad j = 1, \dots, m \\ \pi_{i,j} \geq 0 & \quad i = 1, \dots, n, j = 1, \dots, m \end{aligned} \tag{2.10}$$

From a linear problem like this, we are able to construct a dual formulation. More explicitly, it can be stated as follows.

The Primal Problem ⌟

$$\begin{aligned} \min\ b^T y \\ s.t.\ A^T y = c \\ y \geq 0 \end{aligned} \tag{2.11}$$

The Dual Problem

$$\begin{aligned} \max\ c^T x \\ s.t\ Ax \leq b \end{aligned} \tag{2.12}$$

The proof of these being equivalents can be found in [Villani, 2009]. Back to the Discrete Kantorovich formulation, its dual formulation can be written as follows: (Note that the vector x from the previous example is now represented by $u_1, \dots, u_n$,$v_1, \dots, v_m$)

The Dual Discrete Kantrovich Problem

$$\begin{aligned} \max \sum_{i=1}^{n} u_i f_i^+ + \sum_{j=1}^{m} v_j f_j^- \\ st.\ u_i + v_j \leq c_{i,j} \quad i = 1, \dots, n\ j = 1, \dots, m \end{aligned} \tag{2.13}$$

Moreover: The max of The Dual equals the min of The Primal. We will formally state this in the following theorem.

The Dual Kantorovich Problem *Suppose that we have two nonnegative measures f^+ and f^- on X, Y satisfying $f^+(X) = f^-(Y)$, and a given cost function $c : X \times Y \to \mathbb{R}$. Let $\mathcal{L}_c$ be the set*

$$\mathcal{L}_c := \left\{ \begin{array}{r} (u, v) \in \mathcal{C}_b(X) \times \mathcal{C}_b(Y) \ \text{s.t.:} \\ u(x) + v(y) \leq c(x, y)\ \forall\ (x, y) \in X \times Y \end{array} \right\} \tag{2.14}$$

Find (u^*, v^*) solving the minimization problem

$$\sup \int_X u(x) df^+(x) + \int_Y v(y) df^+(y) \tag{2.15}$$

Where $\mathcal{C}_b(X)$ denotes the set of all continuous and bounded functions on X. We can state the following theorem about the existence of the solution for this problem:

Theorem 2.10 (Kantorovich Duality). *Given two nonnegative measures f^+ and f^- on X, Y satisfying $f^+(X) = f^-(Y)$, and a given, semi-continuous, cost function $c : X \times Y \to \mathbb{R}$, the following equality holds:*

$$\min_{\pi \in \Pi(f^+,f^-)} \int_{XxY} c(x,y) d\pi(x,y) = \max_{(u,v)\in\mathcal{L}_c} \int_X u(x) df^+(x) + \int_Y v(y) df^+(y) \qquad (2.16)$$

The full proof can be found in [Villani, 2009]. We can then apply Thm. 2.8 for existence of the Dual Problem satisfying the conditions of Thm. 2.10.

2.4.1 Cyclicity and Monotonicity

The Dual Kantorovich Problem is also cyclically monotone [Villani, 2009]. That property leads to the following:

Theorem 2.11 (Rockafellar). *A cyclically monotone map can be expressed as the gradient of a convex function.*

This means that we can write $T(x) = \nabla u(x)$ where u is a convex function.
From mass-preservation, we had that $\det(\nabla T(x)) = \frac{f(x)}{g(T(x))}$

$$\implies \boxed{\det(D^2 u(x)) = \frac{f(x)}{g(\nabla u(x))}} \qquad (2.17)$$

Which is the **Monge-Ampere equation**.

Notice that this becomes an elliptic PDE. This exemplifies the connection between the MK equations and PDEs. We will use these properties later.

2.5 The Wasserstein Metric

The idea behind using a Wasserstein metric is to get some kind of indication of the required cost of transporting something from one location to another. A larger distance or a larger volume of the quantity being transported would naturally lead to an increase in total transportation costs. It can identify an optimal transportation cost between two probability measures (f^+, f^-)

$$C(f^+, f^-) = \inf_{\pi \in \prod(f^+,f^-)} \int c(x,y) d\pi(x,y), \qquad (2.18)$$

where $c(x,y)$ denotes the cost of transporting one unit from x to y.

Definition 2.12 (The Wasserstein Distance). *For* $\forall\ p \in [1, \infty)$*, the Wasserstein distance is defined as:*

$$W_p(f^+, f^-) = (\inf_{\pi \in \Pi(f^+, f^-)} \int_{X \times Y} c(x, y) d\pi(x, y))^{\frac{1}{p}} \tag{2.19}$$

⌟

Theorem 2.13. W_p *defines a metric on the space of probability measures.*

Conditions

1. Triangle inequality $W_p(f^+, f^-) \leq W_p(f^+, \sigma) + W_p(\sigma, f^-)$ where $\sigma \in [f^+, f^-]$
2. Symmetry : $W_p(f^+, f^-) = W_p(f^-, f^+)$
3. Positivity: $W_p(f^+, f^-) \geq 0$, $W_p(f^+, f^-) = 0 \implies f^+ = f^-$.

Proof. (1): Triangle Inequality. Fully derived in [Villani, 2009] and [Hamfeldt, 2019].
(2): Immediate.
(3): $W_p(f^+, f^-) \geq 0$. Suppose $W_p(f^+, f^-) = 0$, Then $\exists\ \pi \in \pi(f^+, f^-)$ s.t.
$\int_{X \times Y} |x - y|^p d\pi(x, y) = 0$.
π is supported on the set: $(x, y) \in X \times Y | x = y$.
Choose any $A \subset X$
$f^+(A) = \int_A df^+(x) = \int_{A \times Y} d\pi(x, y) = \int_{A \times A} d\pi(x, y) = \int_{X \times A} d\pi(x, y) = f^-(A) \implies f^+ = f^-$

□

2.6 Existence and Uniqueness

There are a lot of studies done on proving the existence and uniqueness of an optimal transport plan in Optimal Transport Theory. Here we will only re-give the main conjecture done, but interested readers are advised to also read Villani's full work. The existence and uniqueness theorem is based on the importance of having a convex function, so we will start this section by defining convexity:

Definition 2.14 (Convex function). *A function* g *is* ***convex*** *if* $\forall\ x, y \in \Omega(g)$ *and* $\lambda \in [0, 1]$*, then* $g(\lambda x + (1 - \lambda)y) \leq \lambda g(x) + (1 - \lambda)g(y)$*. (Ie. all secant lines lie* ***above*** *the function.)* ⌟

Properties of Convex Functions

- Convex functions are locally Lipschitz continuous, see Def. A.11.
- Convex functions are differentiable on their domain.

More commonly, if $f \in C^2$, then f is said to be **convex** if $D^2 f \geq 0 \, \forall \, x \in \Omega$.

Convex Optimization When a function is known to be convex, it is common to be interested in finding its minimum. Generally, if a function is continuous and differentiable, we can obtain its minimum by setting its derivative to zero.

This leads to the following theorem:

Theorem 2.15. *Consider a compact domain $\Omega \in \mathbb{R}^d$, two probability measures, f^+, f^- such that . Assume that the transport cost has the form $c(x, y) = g(|x - y|)$, where g is a strictly convex function, then there exists a unique transport plan, $\pi^* \in \sum(f^+, f^-)$ of the form $\pi^* = (Id, T^*)_{\#}, f^-$, with $T^* \in T(f^+, f^-)$. Moreover, there exists a Kantorovich potential u and T^* who satisfy the following:*

$$T^*(x) = x - (\nabla g)^{-1}(\nabla(u^*(x))) \tag{2.20}$$

The proof is stated in [Villani, 2009].

This is reliant on using a strictly convex cost function, which excludes the original Monge problem which uses the cost: $c(x, y) = |x - y|$. We refer the reader to [Facca, 2017] for analysis done using the Monge cost.

2.7 Examples of Optimal Transport

2.7.1 Branched Optimal Transport

Branched Optimal Transport is a type of Optimal Transport that encourages moving objects together and then splitting into their final destinations. This might make transportation costs cheaper. This type of Optimal Transport can be found naturally in problems such as road construction, distribution, and resource allocation (rivers, plants, etc.). The *Gilbert-Steiner Problem* [Gilbert, 1967] creates the basis of what encourages a ”Y” type transport instead of a ”V” type transport (see fig. 2.5). We will use these properties when building the dynamic model on a graph structure in the subsequent chapter.

(a) "V" transport.

(b) "Y" transport.

Figure 2.5: Two different ways of transporting coal from mines to factories. Depending on transportation cost, it might be beneficial to transport them together along the common direction.

2.7.2 Congested Optimal Transport

Although these are important topics when discussing Optimal Transport theory, this will not be of major concern here. The reader is referred to [Facca, 2017] for more on this.

Chapter 3

Building The Mathematical Model

The goal of this chapter is to describe the existing mathematical model of a vascular structure. To do this, we use both the dynamics on a discrete graph structure and we couple this with the Optimal Transport theory formulated in the previous chapter. This chapter is important, as the model derived here will be a fundamental building block for subsequent chapters. This chapter is structured as follows:

In section 3.1 we build the governing equations based on the structure of the discrete flow system. Starting with the Fick-Poiseville flux between nodes, Kirchoff's law for mass conservation, and the conductivity dynamics. Finally, we present the full model for the discrete system.

Section 3.2 will up-scale resulting equations from section 3.1. This entails to generalize the properties of the graph structure to the whole continuum. Then we will compare this result and the Monge-Kantorovich equations from Chapter 3.

In section 3.3 we will look at the existence and uniqueness of the PDE in section 3.2. We will also discuss the potential problems with the existence of a solution to the model.

For an update to the most recent models in the literature, section 3.5 will look at the improved measures of adding an extra branching inclination. This will be the base of which we are directly building upon in the following chapters.

The main resources followed in this chapter are [Tero et al., 2006] and [Facca, 2017].

3.1 Discrete Model - Modelling The Dynamics of Biological Tissues

All vascular tissue consist of veins and arteries. ”Arteries deliver blood from the ventricles to vascular beds, while veins return it to the atria” ([Li, 2004] p.18). What we will study is the path taken by arteries transporting blood from the whole domain of interest to some sink that will further transport it to the heart. The exact opposite could be studied by changing the initial conditions and therefore changing the nature of the system. A very interesting problem would be to couple the two processes to simulate a full system.

In this section, we will closely follow the work of [Tero et al., 2006] to describe any graph structure. To construct the discrete model based on the physical principles of our medium, we start by identifying that the branched structure of the arteries can be regarded as a network transportation system with splitting in the nodes, see fig. 3.1. Already by looking at the figure, we can note that in order to describe a transport plan describing the graph, we need to use the Monge Kantorovich formulation, as we need to allow for splitting.

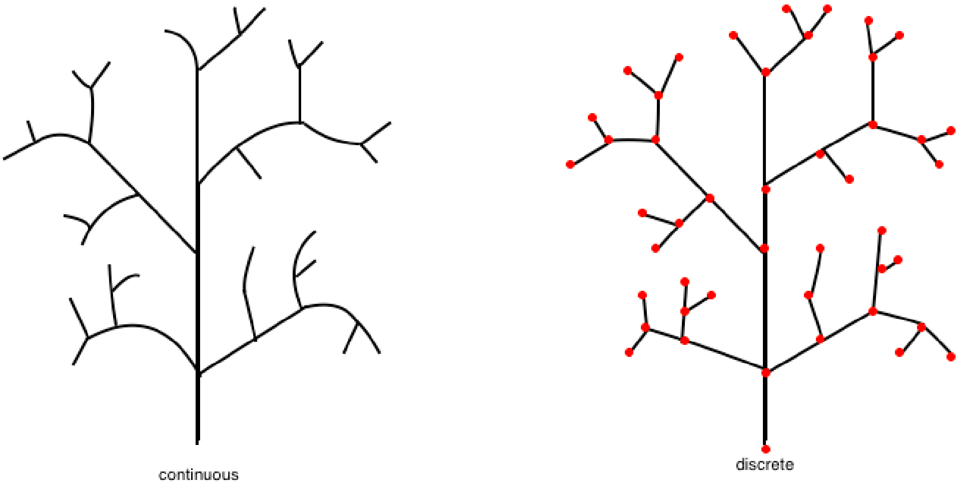


Figure 3.1: Image a) is the continuous shape of the vein being discretized in image b). This is how any vascular structure can be regarded as the graph structure. We will use this structure to build our model starting with the dynamic properties of the discrete case.

The model by [Tero et al., 2006] describes a connected planar graph where we have an equation describing the density transported. The density equation is typically a conservation law coupled with a nonlinear dynamic equation for the flow conductivity along the edges of the graph.

Consider a graph, $G = (V, E)$ with n vertices (nodes) $1, \dots, n \in V$ and m edges, $1, \dots, m \in E$. We can define a length of each edge, L_e, as the length between the nodes of e and the conductivity, D_e, as the connectivity between nodes. Say we have a variational function, p, connected to the potential, or pressure, on G. The flux, Q_e, along each edge of G can then be expressed as:

$$Q_e = \frac{D_e}{L_e}(p_u - p_v) \tag{3.1}$$

We impose mass conservation in all nodes except for the source and the sink nodes. This means that the total flow into a node equals the total flow out of a node, and so their flux is zero. We also impose that the source and sink nodes are equal of equal magnitude, but opposite signs. This is summarized by [Tero et al., 2006] as follows:

$$\sum_{e \in \sigma(v)} Q_e = f(v) := \begin{cases} 0 & v \neq v_0, v_f, \\ 1 & v = v_0, \\ -1 & v = v_f \end{cases} \tag{3.2}$$

Where $\sigma(v)$ signifies the set of all edges containing the node v.

To express the change of conductivity in time, we want to impose the property that larger fluxes will reinforce the density transported along an edge, while smaller fluxes will result in smaller conductivity [Tero et al., 2006]. Because the flux can be either positive or negative, we need to include an absolute sign to reflect that we are only interested in the magnitude of the flux, not its sign. The resulting equation becomes an ODE for the time-development of the conductivity along each edge:

$$\frac{d}{dt} D_e(t) = g(|Q_e(t)|) - D_e(t) \tag{3.3}$$

Where g is an increasing function from $\mathbb{R}^+$ to itself with $g(0) = 0$, and the decay term $-D_e(t)$ is introduced to prevent non-boundedness.

This means that we can use the following set of equations summarized by [Facca, 2017] to describe our dynamic model pair (D_e, p_v) yielding the optimal distribution on the graph structure:

$$\sum_{e\in\sigma(v)} Q_e = f(v) := \begin{cases} 0 & v \neq v_0, v_f, \\ 1 & v = v_0, \\ -1 & v = v_f \end{cases} \quad \forall v \in V, \text{ Kirchoff-law} \tag{3.4a}$$

$$Q_e = \frac{D_e}{L_e}(p_u - p_v) \;\; \forall e \in E \;\text{ Fick-Poiseville} \tag{3.4b}$$

$$\frac{d}{dt}D_e(t) = g(|Q_e(t)|) - D_e(t) \;\; \forall e \in E \;\text{ Conductivity Dynamics} \tag{3.4c}$$

$$D_e(0) = \hat{D}_e(0) > 0 \;\; \forall e \in E, \quad \text{initial data} \tag{3.4d}$$

In [Tero et al., 2006], multiple variations of g were tested. It was shown that using $g(x) = x$, the conductivity converges to zero in every edge except the edges forming the shortest path connecting the start and end node. This could be an indication that it was successful in finding the shortest path. The mass densities also accumulated along the edges with nonzero conductivity.

Moreover, [Bonifaci et al., 2012] showed that for using $g(x) = x$, we have equivalence between the graph problem, and an Optimal Transport problem on the graph G:

$$\begin{aligned} \min_{Q\in\{Q_e\}_{e\in E}} \quad & \sum_{e\in E} Q_e L_e \\ s.t. \; \sum_{e\in\sigma(v)} & Q_e = f_v \; \forall v \in V \end{aligned} \tag{3.5}$$

Which, under certain assumptions on the graph structure, the solution of our discrete equations (3.4) converges to a stationary solution, Q, which is also the solution of the above Optimal Transport problem.

3.2 Continuous Model - DMK

Now, we rewrite the equation (3.4), extending it to the continuum. We start by discussing assumptions that need to be made to upscale the model, and end by presenting the full, continuous PDE and link this back to the work done in Ch. 2. This section will follow the notation of [Facca, 2017].

3.2.1 Extension to the Continuum

Starting from a 1D graph structure as defined in sec. 3.1, certain assumptions need to be made to upscale this to the continuum. It is important to note that since the dimensionless model does not have a spatial direction, the literature only points towards indications that the difference between nodes can be approximated by the gradient. This leads to the following approximation:

$$\frac{1}{L_{i,j}}(p_i - p_j) \approx \nabla p. \tag{3.6}$$

For more on this, see Xia's work, as referred to in [Facca, 2017]. Moreover, this is how the gradient is approximated discretely later when solving the equations, which verifies that these directly correspond to each other.

3.2.2 Building the Continuous Model

Say we remove the graph structure, and instead consider an open, bounded domain, $\Omega \subset \mathbb{R}^d$. From here on, we will only study the case $g(x) = x$, ie. a linear relation.

Setting the flux, Q, as $Q = \mu\nabla u$, the potential function p on the graph structure as u. Set $u, v \in V$ to be two nodes in the graph, D_e the connectivity between them, and L_e the length between them. Recall that μ denotes the transport map. The flux is the transport multiplied by the change in pressure/potential. From this, and point made in sec. 3.2.1, we get that eqn. 3.1 becomes:

$$Q_e = \frac{D_e}{L_e}(p_u - p_v) = \frac{\mu}{h}(u_i - u_j) = \mu\nabla u \tag{3.7}$$

Setting this into the equation 3.2 yields:

$$\sum_{e\in\sigma(v)} Q_e = Q_{e1} + Q_{e2} = \nabla \cdot Q_e = \nabla \cdot (\mu\nabla u) = f \tag{3.8}$$

For all nodes v that are not edge nodes.

Equation 3.3 becomes the following when extended:

$$\begin{aligned} \frac{d}{dt}D_e(t) &= g(|Q_e(t)|) - D_e(t) \\ \frac{d\mu}{dt} &= g(|\mu\nabla u|) - \mu \\ \mu' &= \mu(g|\nabla u| - 1) \end{aligned} \tag{3.9}$$

Provided that μ is strictly positive.

The generalizations into the continuum $\Omega \subset \mathbb{R}^n$ as made above can be summarized as the PDE system referred to as the *Dynamic Monge Kantorovich* (DMK) equations:

$$
\begin{aligned}
-\nabla \cdot (\mu(t,x)\nabla u(t,x)) &= f(x) = f^+(x) - f^-(x) && \text{(3.10a)}\\
\mu'(t,x) &= \mu(t,x)(|\nabla u(t,x)| - 1) && \text{(3.10b)}\\
\mu(0,x) &= \mu_0(x) && \text{(3.10c)}
\end{aligned}
$$

with $(\mu, u) : ([0,+\infty) \times \Omega) \to (\mathbb{R}^+, \mathbb{R})$, zero Neumann boundary conditions, as well as f satisfying the conditions specified and μ_0 being the initial condition on μ.

Note: Using the Neumann boundary conditions on all boundaries does naturally lead to non-uniqueness, as all the derivatives are determined at the boundaries, but the actual values are not. This leads to an unknown integration constant, C.

3.2.3 Connection with Monge-Kantorovich

Solving eq. 3.10 to equilibrium would entail that the system has converged, and no longer has a time evolution, ie. $\frac{d\mu^*}{dt}(x,t) = 0 \Rightarrow 0 = \mu^*(x)\nabla^*(x) = f(x)$. The system can be rewritten as:

$$
\begin{cases}
-\nabla \cdot (\mu^*(x)\nabla u^*(x)) = f(x), \\
|\nabla u^*(x)| = 1 \text{ where } \mu^*(x) > 0, \\
\quad \text{no assumption on } (|\nabla u^*(x)| - 1) \text{ where } \mu^*(x) = 0
\end{cases}
\tag{3.11}
$$

Which are the *Monge-Kantorovich Equations* [Facca, 2017].

3.3 Existence and Uniqueness

In this section we will follow Facca's notation derived in [Facca, 2017] and [Facca et al., 2018] closely to state the existence and uniqueness of the DMK. Later on, no proofs of the further extensions will be conducted, and we will only refer to this section as proof of existence for the base model. The following theorem states the existence of eq. 3.10:

Theorem 3.1. *Given Ω an open, bounded, convex and connected domain in $\mathbb{R}^d$ with smooth boundary, $f \in L^\infty(\Omega)$ with zero mean and $\mu_0 \in C^\delta(\Omega)$ with $\mu_0 > 0$ and*

$0 < \delta < 1$ there exists $\tau_0 > 0$ depending on f and μ_0, such that the system

$$\int_\Omega \mu(t,x)\nabla u(t,x)\nabla\phi(x) = \int_\Omega f(x)\phi(x)dx \ \ \forall\ \phi \in H^1(\Omega) \tag{3.12a}$$

$$\partial_t\mu(t,x) = \mu(t,x)|\nabla u(t,x)| - \mu(t,x) \tag{3.12b}$$

$$\mu(0,x) = \mu_0(x) > 0 \tag{3.12c}$$

$$\int_\Omega u(t,x)dx = 0 \tag{3.12d}$$

admits a solution pair

$$(\mu, u) \in \mathcal{C}^1([0,\tau_0[\mathcal{C}^\delta(\Omega)) \times \mathcal{C}^1([0,\tau_0[,\mathcal{C}^{1,\delta})(\Omega)) \tag{3.13}$$

The full proof is derived in [Facca, 2017], but we will here discuss some of the main parts contributing to this proof.

The Procedure of showing Existence and Uniqueness

1. Showing that the elliptic problem in 3.10a) is well posed on the domain.
2. Showing that the ODE 3.10b) is at least locally Lipschitz continuous on the domain.

Part(1) In order to show the existence and uniqueness of a solution pair (μ, u) of the equation 3.10, we start by examining equation 3.10a). This equation can be classified as an elliptic equation. The general procedure for an elliptic equation is to multiplying by a test function, $v \in V$ and integrate over the whole domain:

$$\begin{aligned} -\nabla\cdot(\mu\nabla u) &= f \\ -\int_\Omega \nabla\cdot(\mu\nabla u)v &= \int_\Omega fv \\ a(u,v) &= L(v) \end{aligned} \tag{3.14}$$

Note: That this form is only obtained if $\mu = C \in \mathbb{R}^n$. As the μ is not assumed to obtain a constant value here, see [Facca, 2017] for the proof done on non-constant μ.

The eq. 3.14 is directly applicable to the Lax Milgram Lemma:

Theorem 3.2 (Lax Milgram Lemma). *Assuming that $a(.,.)$ is symmetric, bilinear, coercive and continuous, $L(.)$ is linear and bounded, and $(V, ||V||)$ is Banach, then the variational problem:*

$$\textit{Find}\ \ u \in V \ \ \textit{s.t.}\ \ \ a(u,v) = L(v) \ \ \ \forall\ v \in V \tag{3.15}$$

Has a unique solution. Moreover, the following stability estimate holds:

$$||u|| \leq \frac{M_L}{m} \tag{3.16}$$

Where $M_L, m \in \mathbf{R}$

The proof of the Lax Milgram Lemma can be found in [Johnson, 1987].

Part (2) By denoting this solution, u, as $\mathcal{U}$, we need to prove the existence and uniqueness of a solution μ solving the equation

$$\mu' = \mu(|\nabla \mathcal{U}|^2) - 1 \tag{3.17}$$

This equation can also be rewritten as:

$$\partial_t \mu(t) = \mathcal{Q}(\mu(t)) - \mu(t) \tag{3.18}$$

The following proposition states that $\mathcal{U}$ and $\mathcal{Q}$ are locally Lipschitz. See sec. A.5 for the definition of Local Lipschitz continuity.

Proposition 3.3. *The potential and Flux operators $\mathcal{U}$ and $\mathcal{Q}$ are Lipschitz continuous and bounded in $\mathcal{D}(a,b)$ for all a and b, $0 < a < b < \infty$. This yields that for every $\mu \in \mathcal{D}(a,b)$,*

$$\begin{aligned} ||\mathcal{U}(\mu)||_{C^{1,\delta}(\Omega)} &\leq C_1(a,b) \\ ||\mathcal{Q}(\mu)||_{C^{\delta}(\Omega)} &\leq C_2(a,b). \end{aligned} \tag{3.19}$$

Moreover, there exists constants $L_(\mathcal{U})(a,b)$ and $L(\mathcal{Q})(a,b)$ such that, for every μ_1,μ_2 $\in$ $\mathcal{D}(a,b)$:

$$\begin{aligned} ||\mathcal{U}(\mu_1) - \mathcal{U}(\mu_2)||_{C^{\delta}(\Omega)} &\leq C_1(a,b) \\ ||\mathcal{Q}(\mu)||_{c^{\delta}(\Omega)} &\leq C_2(a,b) \end{aligned} \tag{3.20}$$

A full proof can be found in [Facca, 2017].

3.4 Lyapunov Candidate Functional

The Lyapunov Candidate Functional was first introduced in [Facca et al., 2018] as a functional whose minimizer is the equal of the solution of Eq. 3.10.

The Lyapunov Functional, S, is made up of the sum of the energy functional ε_f and a mass functional $\mathcal{M}$ given by:

$$\begin{aligned}\mathcal{S}(\mu) &:= \varepsilon_f(\mu) + \mathcal{M}(\mu) \\ \varepsilon_f(\mu) &:= \sup_{\phi \in Lip(\Omega)} \int_\Omega (f\phi - \mu\frac{|\nabla\phi|^2}{2})dx \\ \mathcal{M}(\mu) &:= \frac{1}{2}\int_\Omega \mu dx\end{aligned} \tag{3.21}$$

The $\varepsilon_f(\mu)$ can equivalently be written as:

$$\varepsilon_f(\mu) = \frac{1}{2}\int_\Omega \mu|\nabla\mathcal{U}_f(\mu)|^2 dx \tag{3.22}$$

Where $\mathcal{U}_f(\mu)$ identifies the solution of eq. 3.10 given μ. It has also been shown that the OT density μ_f is the unique minimizer of the functional $\mathcal{S}$ and that the minimum equals the Wasserstein distance between f^+ and f^- with a cost equal to the Euclidian distance [Facca et al., 2018] (Recall the definition of the Wasserstein Distance from sec. 2.5).

Proposition Given Ω and open, bounded, convex, and connected domain in $\mathbb{R}^d$ with a smooth boundary. Consider f $\in L^1(\Omega)$ with zero mean, then the Beckmann problem (MK equations) and the minimization of $\mathcal{S}$ are equivalent which means:

$$\min_{v\in[L^1(\Omega)]^d}\left\{\int_\Omega |v|dx : div(v) = f\right\} = \min_{\mu \in L^1_+(\Omega)} \mathcal{S}(\mu) \tag{3.23}$$

Where $L^1_+(\Omega)$ denotes the space of non-negative functions in $L^1(\Omega)$. Moreover, the OT density $\mu(\Omega)$ and $f \in \mathcal{S}(\Omega)$.

3.5 Extended Formulation - eDMK

Moreover, [Facca, 2017] showed that introducing a new parameter significantly improved the branching ability of the medium. He also provided indications that this numerically resembles the Branched Optimal Transport theory. The difference is that the dynamical flux, $q = |\mu(t)\nabla u(t)|$ is raised to the power of $\beta_F > 0$. For the remainder of the work, we will use the following equations called the *extended Dynamic Monge Kantorovich* (eDMK):

$$-\nabla \cdot (\mu(t)\nabla u(t)) = f(x) = f^{+}(x) - f^{-}(x) \tag{3.24a}$$

$$\partial_t \mu(t) = (\mu(t)|\nabla u(t)|)^{\beta_F} - \mu(t) \tag{3.24b}$$

$$\mu(0) = \mu_0(x) > 0 \tag{3.24c}$$

with Neumann boundary conditions. Note that for $\beta_F = 1$, the model remains the same as before, while $0 < \beta_F < 1$ and $\beta_F > 1$ will change the equation. We will here only summarize the analysis done by [Facca, 2017].

$\underline{\beta_F = 1}$**:** The solution pair (μ, u) solves the Monge Kantorovich equation. This connection was described in section 3.2. The existence condition and Lyapunov functional were presented in sections 3.3 and 3.4 respectively.

$\underline{0 < \beta_F < 2}$**:** The Lyapunov candidate functional has been shown to be strictly decreasing along the μ trajectory.

$$\mathcal{L}_{\beta_F} = \frac{1}{2}\int_{\Omega} \mu |u|^2 dx + \frac{\beta_F}{2(2-\beta_F)}\int_{\Omega} \mu^{2-\beta_F} dx \tag{3.25}$$

$\underline{0 < \beta_F \leq 1}$**:** The solution, w of

$$\nabla \cdot (-|\nabla w|^{p-2}\nabla w) = f \tag{3.26}$$

with $p = \frac{2-\beta}{1-\beta}$, zero Neumann boundary conditions and $(u, \mu) = (w, |\nabla w|^{p-2})$ is the steady state of the eq. 3.24.

$\underline{1 < \beta_F < 2}$**:** These values yield a loss of regularity, however, numerical experiments indicate a convergence towards Branched Optimal Transport models even though the exact connection is not known.

Despite the loss of regularity, we will continue to build on the model presented in eq. 3.24 as we have through the sections 3.1- 3.5 seen that it models dynamics we wish to describe.

Chapter 4

Describing Splitting Structure

This chapter includes an original derivation of a continuous function that penalizes splitting points arising in structures. The aim is to explore the properties of a splitting point in an image. The chapter is structured as follows:

In section 4.1, we will motivate the need for the functional we are about to derive and discuss how this would fit into current models and how it is useful. The expression will be derived in section 4.2, where we will also present the resulting functional, as well as provide a summarizing table describing the parameters that go into the expression. In section 4.3, we will apply the expression to a few images that will be of the same type of structures we expect to encounter. Finally, we will in section 4.4 discuss the results from section 4.3, as well as any limitations this expression might have. It is important to note that this chapter will only be verified with the few examples in section 4.3. More test will be conducted in chapter 6.

4.1 Justification

What we mean by a splitting point is a branch that either splits into two or more smaller branches, or has any offspring stemming from it (see fig. 4.1a)). Recall the graph figure in fig. 3.1 which had all the splitting points at the nodes.

The motivation for detecting a splitting point arises from the hypothesis that any flow through a tube or vein that splits will slow the speed of the flow [Quohar et al., 2020]. This increases the total cost of distributing to the whole domain. The current mathematical model for biological optimal transport in Ch. 3 does not consider this, and as a result, ends up with some branches that are only mathematically justified. The hope is that by being able to detect the splitting of any branching structure, we would be able to allocate extra cost to the points of interest and hence be able to get rid of the excess branches.

The target here is then to propose a functional locating the splitting points in a 2D structure that can be used as an extra term in the optimal transport problem. The 2D structures will be seen as images, a note on how to upscale the dimensionality will be given in Section 4.4.3.

4.2 Derivation

To mathematically determine the qualities of a branching point, we start by looking at images containing branching structures. For a simple black and white image as we will study here, the value of the image will either be or at least can be scaled to be, one or zero, see figure 4.1a) and hence it is discrete and discontinuous. To ensure that we have continuity, the initial image must be convoluted with a test function, namely any function from the Gaussian scale space with zero mean. The convolution process can be expressed as:

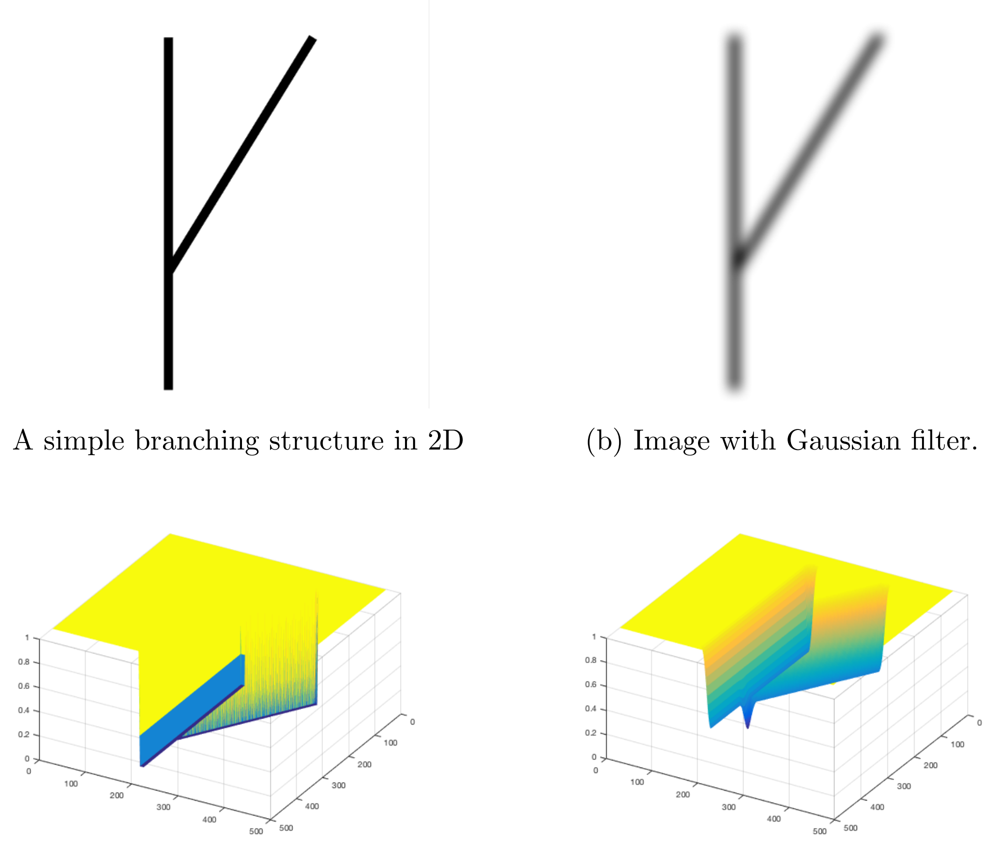
$$f \otimes g = \int_{\infty}^{\infty} f(x-u)g(u)du \tag{4.1}$$

Where the function f is the initial image and g is the test function which is a Gauss filter with zero mean and variable standard deviation, σ. A visual image as to what happens when convolution is applied, see figure 4.1.

One can note that when the Gaussian filter is applied to the structure, it becomes continuous, and the point where the structure splits can be described as a minimum point, see figure 4.1d).

4.2.1 Finding Local Minima

Multiple approaches could be taken to detect local minimum points. Firstly, one could try a discrete approach, and simply compare the locally smallest values on a cell to cell basis using a discretization of the image. This does not guarantee an accurate result however since the minima will mostly reside on a sub-cell position, and a rough partition of the domain would not be able to detect this. See [Kuijper, 2004] for a more in-depth tutorial on detecting extremum points using a hexagonal grid. As well as a discussion of when using the discrete approach could be useful. Secondly, the mathematical definition of something being a local minimum



(a) A simple branching structure in 2D

(b) Image with Gaussian filter.

(c) A 3D view of the initial structure.

(d) Image with Gauss filter.

Figure 4.1: visualization of what happens when a Gaussian filter with standard deviation σ is applied.

is the existence of two positive eigenvalues in a point. It is this definition we will use in the following section.

The literature also supports the use of eigenvalues, as they are closely related to singular values, $\lambda^2 = \sigma$. This means that the singular values are always positive, and we will not be able to differentiate between positives and negatives. This notation can be confusing since both the singular values and standard deviations are denoted as σ. However, except for this paragraph, and Appendix A.1, we will always be discussing the standard deviation when mentioning σ.

A low-rank approximation (See section A.3) is an approximation of the matrix or image using rank-1 components. As stated by [Drineas et al., 2006]: 'Often only a few of the singular vectors are needed to extract the major appearance characteristics of an image'. This means that low-rank approximations work so well because they draw out the characteristics of an image. In other words, low-rank approximations will draw out all top and bottom points to a specified rank n. When only interested in the bottom points, however, we must consider the eigenvalues, as they reveal the orientation of the image value.

4.2.2 Deriving a Function for Continuous Detection

Any point (x, y) inside a domain Ω is a local minima if both of the eigenvalues $eig(D^2u) = \lambda^+, \lambda^-$ are positive, where D^2u is the Hessian of the scalar function $u \in \mathbf{R}^2$ defined as:

$$D^2u = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{4.2}$$

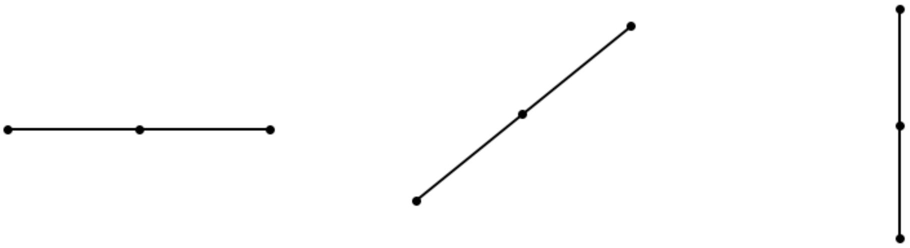


Figure 4.2: Stencils used for D_{xx}, D_{xy} and D_{yy} respectively.

Each element of the matrix in eq. (4.2) is a finite difference approximation of the derivatives.

$$\begin{aligned} D_{xx} &= \frac{1}{h^2}\left(U_{i-1,j} - 2Ui,j + U_{i+1,j}\right) \\ D_{xy} = D_{yx} &= \frac{1}{h^2}\left(U_{i+1,j+1} - 2U_{i,j} + U_{i-1,j-1}\right) \\ D_{yy} &= \frac{1}{h^2}\left(U_{i,j-1} - 2U_{i,j} + U_{i,j+1}\right) \end{aligned} \tag{4.3}$$

Where U_i, j is a discrete grid function approximation of u in the points $(x_i, y_j) \in \Omega$, and h is the mesh size step (Note: granted that $\nabla x = \nabla y$). The elements are described using a stencil like the one described in fig. 4.2. We expect to obtain 2nd order accuracy from these all these discretizations [LeVeque, 2007]. More on approximating derivatives can be found in section A.4.

Both of the eigenvalues will be positive when the smallest eigenvalue is positive, the function for the smallest eigenvalue can, therefore, be defined as:

$$\lambda^{-} = tr(D^2u) - \sqrt{(tr(D^2u))^2 - 4det(D^2u)} \tag{4.4}$$

This function can return all real values, but we are only interested in the positive ones. When creating a functional taking these values $\lambda^{-} \in \mathbb{R}$ we would like the cost functional to only return a positive cost when a branching point is encountered, ie. the branching point in penalized with the cost $f(u)$.

$$f(u) \in \mathbb{R} : \mathbb{R} \to \mathbb{R}^{+}$$

An example of a function that takes the whole real line of values and only returns a positive value or zero is the exponential function. The only clear issue with using this function is the fact that it is not upper-bounded, and could rapidly grow unlimited unless otherwise stated. By using a limiting parameter, β, we can restrain $f(u)$ from becoming very large. β requires that the magnitude of λ^{-} is known a-priori. This parameter can roughly be determined by the boldness of the branching structure, ie. β can be related to the standard deviation σ, used in the convolution discussed in section 4.2.

This will result in that any $\lambda^{-} > \beta$ will still be positive, while the rest will be negative, and by this, limiting the maximum value of the exponential expression.

Another parameter used in the definition of the functional $f(u)$ is α which represents an amplification factor. After β is subtracted from λ^{-} there will be a lot of values surrounding 0. Then one can expect that the output will be very monotonous with a lot of values equal to 1 since $e^0 = 1$. To differentiate between the positive values and the negative values, they are multiplied by α such that the differences will be more clear. All negative values will tend to zero and the positive values will be amplified. Also, we add an extra parameter, γ, in front to scale the magnitude of the whole expression. This parameter becomes important when weighing this term relative to the other components of our full formulation. The resulting functional becomes:

$$f(u) = \int_{\Omega} \gamma \mathrm{e}^{\alpha(\lambda^{-} - \beta)} \, \mathrm{d}\Omega \tag{4.5}$$

Numerical results of the equation stated above are listed in the following section.

4.2.3 Summary Table

Name	Description
γ	Scaling Parameter
α	Amplification parameter
β	Limiting parameter
λ^-	Function returning the smallest value of the eigenvalues in any given point, $\lambda^-(x) \in \mathbb{R}, \text{where} x \in \Omega$
$f(u)$	Functional highlighting the points with large positive eigenvalues

4.3 Examples and Results

All of the following examples use the same parameters but with varying initial images u.

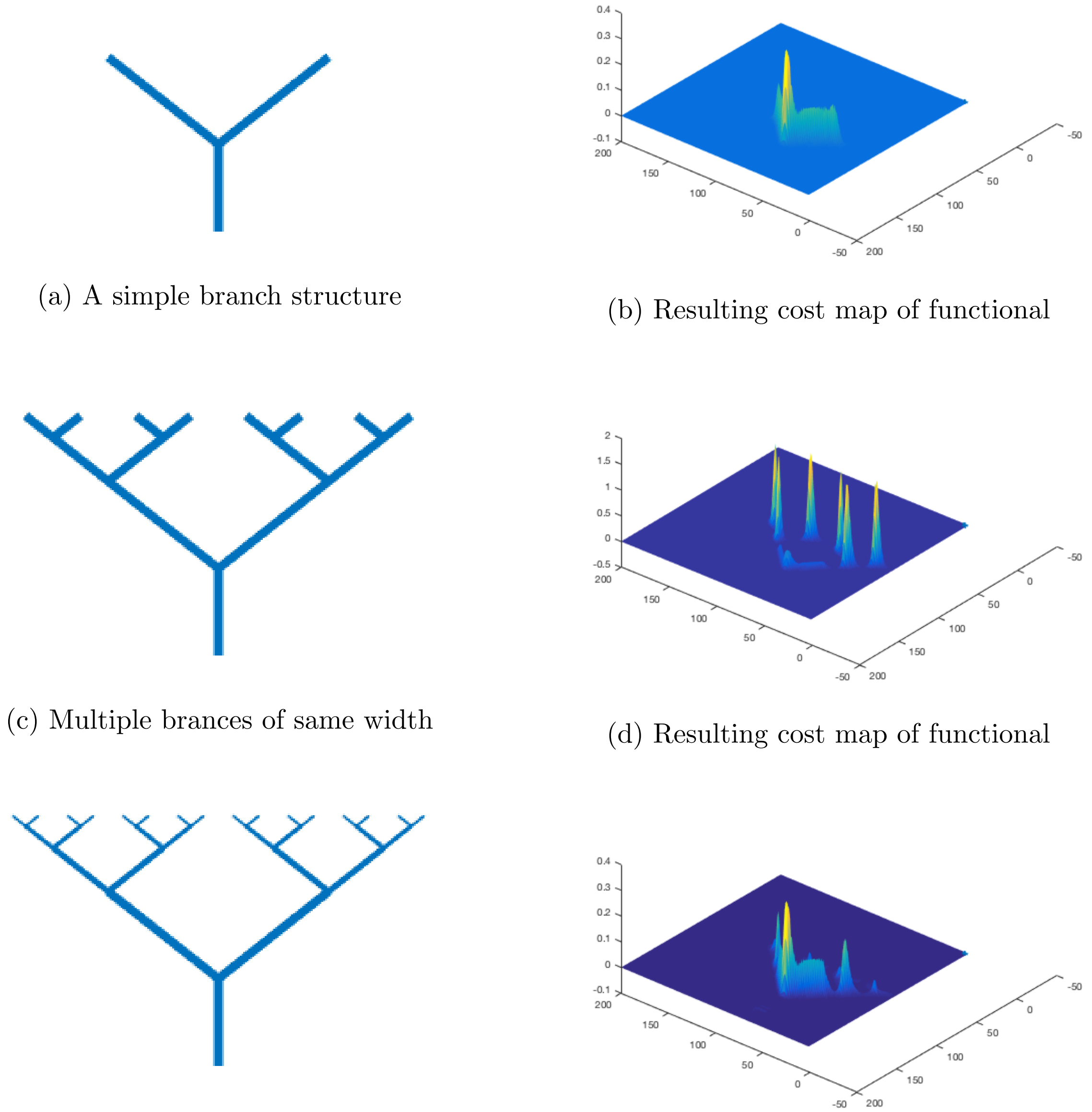
Parameter	Input Data
u	fig. 4.3 a), c) and e) respectively
γ	1
α	500
β	0.016
σ	5

Note: The Gauss filter is applied to all of the images before returning the cost map of the functional. The σ in the table above refers to the standard deviation used for the Gauss filter. Also, since we here only look at the structure by itself, γ is set to 1 throughout this chapter.

From figure 4.3 we see that in all of the examples above, the functional returns a higher value exactly in the splitting points. We regard this as a successful way to continuously penalize a structure only in the splitting points. There are, however, a few comments to be made about the differences in the examples above.

4.4 Discussion

From the examples, we see that the output function is roughly depending on the scaling of the parameters, a value between 0 and 1. The splitting points become visually amplified, so one can think of the functional as an amplification factor of the splitting points of the initial image.



(a) A simple branch structure

(b) Resulting cost map of functional

(c) Multiple brances of same width

(d) Resulting cost map of functional

(e) Multiple branches with decreasing width

(f) Resulting cost map of functional

Figure 4.3: Examples of different types of structures exhibiting a splitting feature and their resulting cost maps from applying functional $f(u)$.

The first image in fig. 4.3 is the image of the initial function u to which the functional $f(u)$ is applied. Note: The displayed image is before adding the Gaussian filter. In the second image, we see that the structure shows something looking like a delta function in exactly the points where the splitting point is, indeed verifying that the value increases in the splitting point.

As seen in the figure above, having a structure that splits multiple times but all while having the same branch thickness will return multiple points of the same magnitude. The only one that differs is the initial point as this does not trigger the exponential function as much as the other ones. This might be because the point is surrounded by more 'air' and is blurred more easily by the Gaussian filter.

As a natural sequel to the previous example we now look at the same structure but with the branches decreasing branch width as the structure keeps splitting, see fig. 4.3e). This might appear more natural in contrast to studying relevant structures as veins or other transport problems. Here the initial splitting point is the largest one and this one triggers the functional more than the sequential points.

4.4.1 More examples

More examples will be done in Ch. 6 on the functional to test things like how the parameters change the output and the cost difference between an 'ideal branch' vs. a non-ideal branch.

4.4.2 Further Implementation

The whole motivation for this functional was to use it in the implementation of optimal transport. As what we essentially want to do is to penalize the splitting points, adding this functional to any image with branching points, the functional will extract the branching point only. For an optimization problem, the optimal cost will occur when the number of branches emitted from the structure is limited. In the following chapters, we will be adding this new term to existing formulations of describing these types of structures to see if it changes the overall appearance of the structures created. Further testing of this expression will be done at the very end of this work.

4.4.3 Limitations

As briefly stated earlier, there are multiple limitations to using this functional. The main one being that the prediction of branching points becomes severely in-accurate for self-crossing structures. In reality, for our main purposes, this will not be of any concern as we do not see any instances of this happening in biological examples, or a simple up-scaling of $n+1$ dimensions could also be applied to extract this crossing.

Also stated was the suggestion of up-scaling to 3D which will be more applicable to studying vascular tissue in the brain and in general 3D structures. This can be done simply by extending the dimension of the x coordinate from 2 to 3 dimensions (x, y, z). Even though this is simple in practice, visualization becomes difficult because a full overview requires 4 dimensions, so this is skipped for the time being.

Another issue mentioned was the significance of differentiating between noise and the structure itself when the branches become very small. Again, de-noising the image will result from applying the Gaussian filter. All positive values will still trigger the functional, but to a very small extent, as was visualized in figure 4.3.

Chapter 5

Expanded Models and Solution Strategies

This chapter will include a proposition of a new functional derived from the dynamics of the structure. This functional, and the one derived from the previous, chapter will be included in the full DMKe model derived in Ch. 3. Finally, we will propose a solution strategy by using Finite Volumes. The result from this chapter will be two original models representing the added splitting penalty we want to obtain. They will both be tested thoroughly in Ch. 6. This chapter is structured as follows:

In sec. 5.1 we derive an original splitting penalty formulation based on the dynamics of the structures we regard throughout this work. This is also added to the DMKe derived in sec. 3.5, and up-scaled to the continuum similarly as presented in 3.2.1.

Section 5.2 will be using the same strategies as 5.1 to take the proposed penalty functional from Ch. 4 into the DMKe model.

In sec. 5.3 we briefly discuss the existence and uniqueness of the two proposed PDEs from sec. 5.1 and 5.2, followed by a proposition of solution strategies in sec. 5.4. Here we will propose the full Finite Volume formulations of the models, solvers of the resulting systems of equations and a mesh.

This chapter uses the approach of [Tero et al., 2006] and [Facca, 2017] to extend the DMKe from eq. 3.24 to include the original splitting penalties. In section 5.4, we use the definitions from [Nordbotten, 2019] to define the grid and solution strategies.

5.1 An unexpected splitting penalty model

By going back to the discrete case from sec. 3.1, one can notice that the only change from adding a no splitting condition is how the conductivity changes over time. In addition to having tubes of smaller flux degenerate, we also want nodes with large pressure differences to be penalized, as these are assumed to be a small branch emerging from a large one.

Looking at the way our discrete model is set up, introducing this new property will still yield the same flux. What will be changed, however, is the conductivity between two nodes who vary widely in size. This means that the new property is making it more expensive to have flow going from a big node to a small node. This can also be seen in image examples where we have shown that every time, a branch emerging form the main branch will trigger the cost functional in a more severe way than a branch emerging from the smaller branched further out in the structure, see figures in sec. 4.3.

With this in mind, it makes sense that it is the conductivity that will be updated to reflect our property. The previous condutivity was the time-varying function:

$$\frac{d}{dt}D_e(t) = g(|Q_e(t)|) - D_e(t) \tag{5.1}$$

Recall that we are still using $g(Q) = Q$ as discussed in Ch. 3. Satisfying the increasing and $g(0) = 0$ conditions.

To add the additional penalty, we need to decrease the conductivity where the pressure differences are high. To ensure that this becomes a positive number, roughly on the desired scale, we use the exponential function, and some scalable parameters, to get a continuous function taking all real numbers and returning positive ones, as described in Ch. 4. This can be expressed as:

$$e^{(p_i - p_j) - \beta} \tag{5.2}$$

Which, when put into the ODE for conductivity, we might want to express this term in terms of flux Q, and conductivity, D.

Note: We do not need to include an absolute sign, as one of the conditions of the structure is that it cannot create a larger branch from a smaller one, and hence $(p_i - p_j) > 0 \; \forall \; i, j \in N$ (nodes), as long as $j > i$ holds. We can at least expect this from the discrete model on the 1D graph, but for the sake of generalizing to the continuum, we will have to include an absolute sign later on.

$$\begin{aligned} Q &= \frac{D_{ij}}{L_{ij}}(p_i - p_j) \\ \implies \quad (p_i - p_j) &= Q_{ij}\frac{L_{ij}}{D_{ij}} \\ \implies \quad e^{(p_i - p_j) - \beta} &= e^{Q\frac{L}{D} - \beta} \end{aligned} \tag{5.3}$$

5.1.1 Parameters

As in Ch. 4, we also here want to introduce a few parameters to this term that help us regulate the magnitude of its output. See Ch. 4 for more explanation behind these parameters. The final term ends up looking like:

$$\gamma e^{\alpha(Q\frac{L}{D} - \beta)} \tag{5.4}$$

$$\begin{aligned} \beta &= \max_{i,j \in N}(Q\frac{L}{D}) \\ \implies \max(e^{(Q\frac{L}{D} - \beta)}) &= 1 \end{aligned} \tag{5.5}$$

When constructing the time varying ODE, subtracting the conductivity signifies the smaller branches degenerating, so that the flux would concentrate along the bigger branches. We still want this to hold, but we also want extra degeneration of branches with splitting. Therefore, we multiply the conductivity with the term just introduced, and it will look as follows:

And that then gives us the time varying ODE

$$\boxed{\frac{d}{dt}D(t) = |Q| - D - \gamma D e^{\alpha(|Q|\frac{L}{D} - \beta)}} \tag{5.6}$$

Which is the discrete case. Rewriting these equations into being valid for the whole dense continuum $\Omega \in \mathbb{R}^d$, yields the following continuous formulation:

$$\begin{aligned} Q &= \mu \nabla u \\ \implies \quad \frac{d}{dt}\mu &= |\mu \nabla u| - \gamma \mu e^{\alpha(\frac{L}{\mu}\mu \nabla u - \beta)} - \mu \\ \frac{d}{dt}\mu &= \mu |\nabla u| - \gamma \mu e^{\alpha(\nabla x \nabla u - \beta)} - \mu \\ &\boxed{\frac{d}{dt}\mu = \mu(|\nabla u| - 1 - \gamma e^{\alpha(\nabla x \nabla u - \beta)})} \end{aligned} \tag{5.7}$$

As the conditions of μ require it to be strictly positive.

$\implies$ The PDE we want to solve becomes:

$$\boxed{\begin{cases} -\nabla \cdot (\mu \nabla u) = f \\ \mu' = \mu(|\nabla u| - 1 - \gamma e^{\alpha(\nabla x \nabla u - \beta)}) \\ \mu(0, x) = \mu_0(x) \end{cases}} \tag{5.8}$$

Now note, we can again begin to discuss how much this resembles our original image filter, plus the scaling of the different terms. We can also discuss if the ∇x needs to be included in the equation at all, or if it could be a part of the β parameter since it is just a threshold to limit the exponential function output.

5.1.2 Branched Transport Model

Recall the parameter β_F shown to encourage branching in Ch. 3. Adding this property to the previous extension yields the following:

$$\boxed{\begin{cases} -\nabla \cdot (\mu \nabla u) = f \\ \mu' = \mu(|\nabla u|)^{\beta_F} - \mu - \gamma \mu e^{\alpha(\nabla x \nabla u - \beta)} \\ \mu(0, x) = \mu_0(x) \end{cases}} \tag{5.9}$$

We will later refer to this model as the **Grad Formulation Model**, as the splitting penalty is based on the gradient of the solution.

5.2 Model for splitting penalty based on image processing

In Ch. 4, we derived a function that adds additional cost to splitting points. This is the feature we want to include in our full model. Similarly to sec. 5.1, we now want to include this term into the existing model.

Again, as our term affects how the fluxes in the tubes are reinforced, the term will affect the conductivity of the flow. The new time-varying ODE becomes:

$$\frac{d}{dt} D(t) = |Q| - D - \gamma e^{\alpha(\lambda^- - \beta)} \tag{5.10}$$

Directly inputted into the eDMK model from eq. 3.24, we end up with:

$$\boxed{\begin{cases} -\nabla \cdot (\mu \nabla u) = f \\ \mu' = \mu(|\nabla u|)^{\beta_F} - \mu - \gamma \mu e^{\alpha(\lambda^- - \beta)} \\ \mu(0,x) = \mu_0(x) \end{cases}} \tag{5.11}$$

Which looks a lot like eq. 5.9, but thresholding on the eigenvalues instead of the gradient. We will later refer to this model as the **Lambdaminus formulation.**

5.3 Existence and Uniqueness

Since we end up having two full models, and both are using the β_F parameter that has different senses of regularity for the different cases, we will not conduct any solid proofs of the existence and uniqueness here. Instead, we will conduct a number of numerical experiments later, that give indications of at least an existing solution, as well as we again refer the reader to the analysis done on the model, not including our new terms.

Although a full analytical solution for these types of problems is yet to be found, we still need to show that there are at least indications as to whether they should exist. The procedure for showing existence and uniqueness is the same as the one we used for the DMK in Ch. 3.

- Showing that the elliptic problem in eq. 5.9 a) and eq. 5.11 a) are well posed on the domain.
- Showing that the ODEs 5.9 b) and 5.11 b) are at least locally Lipschitz on the domain.

See Theorem A.13 for a definition of something being only locally Lipschitz, and sec. A.5 for work on Nonlinear ODEs.

Point 1) remains equal across the chapters as eq. 5.9 a) and eq. 5.11 a) are the same, so we refer the reader to sec. 3.3 for this proof. For point 2), see [Facca, 2017] for a proof of the previous term behind locally Lipschitz. What remains to show is that the terms $\mu e^{\nabla u}$ and $e^{\alpha(\lambda^- - \beta)}$ are locally Lipschitz in terms of μ and u, which suffices as conditions for existence in the cases of $0 < \beta_F \leq 1$ and $\beta_F = 2$.

5.4 Solution Strategies

In order to solve the models, we will proceed to put our equations into the Finite Volumes framework. Advantages from choosing the finite volume approach is that we can ensure mass conservation and that we can expect a lower error from using a rectangular grid. Especially the mass conserving property becomes vital as we are interested in the transport variable, μ. We are also able to easily use the Divergence Theorem on the equations 3.10. Solving these equations using FV is an original idea by [Nordbotten, 2019]. For more on finite volumes and finite elements see [Nordbotten and Celia, 2011] and [Johnson, 1987].

5.4.1 Finite Volume Formulation of the eDMK

Writing out eq. (3.10) into a finite volume approximation, we use the standard framework for converting it. Starting from the Poisson equation 3.10a) we get:

$$\begin{aligned} -\nabla \cdot (\mu \nabla u) &= f \\ \int_{\Omega} \nabla \cdot q dx &= \int_{\Omega} f dx \\ \int_{\delta\Omega} q \cdot n dS &= \int_{\Omega} f dx \end{aligned} \tag{5.12}$$

Where q is the transport flux. [Nordbotten, 2019] introduces $\eta = \log \mu$ to ensure that $\mu = e^{\nu} > 0$ in the distance scheme. Taking the logarithm also ensures that we will end up with a more managable magnitude on the variables. Using the suggested variable changes in the finite volume formulation yields the following:

$$\begin{aligned}
q = -e^{\eta}\nabla u \qquad \eta = ln\mu \\
\frac{d}{dt}\mu = (|\mu\nabla u|)^{\beta_F} - \mu \\
\ln\frac{d}{dt}\mu = \ln(|q|^{\beta_F}) - \ln\mu \\
\ln\frac{d}{dt}\mu = \ln(|q|^{\beta_F}) - \eta \\
e^{\ln\frac{d}{dt}\mu} = e^{\ln(|q|^{\beta_F})} - e^{\eta} \\
\frac{d}{dt}\mu = (|q|^{\beta_F}) - e^{\eta} \\
\frac{1}{e^{\eta}}\frac{d}{dt}e^{\eta} = \frac{(|q|)^{\beta_F}}{e^{\eta}} - \frac{e^{\eta}}{e^{\eta}} \\
\frac{1}{e^{\eta}}\frac{d\eta}{dt}e^{\eta} = \frac{(|q|)^{\beta_F}}{e^{\eta}} - 1 \\
\frac{d}{dt}\eta = \frac{|q|^{\beta_F}}{e^{\eta}} - 1
\end{aligned} \tag{5.13}$$

Paired with the conditions:

$$\begin{aligned}
q \cdot n = 0 \quad on\ \delta\Omega \times [0, T] \\
\eta(x, 0) = \log(\mu_0(x))
\end{aligned} \tag{5.14}$$

Adding a small diffusion term for stabilization and adding an explicit time update [Nordbotten, 2019]:

$$\begin{aligned}
\frac{d}{dt}(\eta - \Delta t\nabla \cdot (\epsilon\nabla\eta)) = \frac{|q|^{\beta_F}}{e^{\eta}} - 1 \\
(I - \Delta t\nabla \cdot (\epsilon\nabla\cdot))\eta^{n+1} = \Delta t(\frac{|q^n|^{\beta_F}}{e^{\eta^n}} - 1) + (I - \Delta t\nabla \cdot (\epsilon\nabla\cdot))\eta^n
\end{aligned} \tag{5.15}$$

The result is a stable coupled system with an explicit Euler expansion as the time update:

$$\begin{aligned}
\int_{\partial\Omega} q \cdot n\ dS = \int_{\Omega} f\ dx \\
(I - \Delta t\nabla \cdot (\epsilon\nabla\cdot))\eta^{n+1} = \Delta t(\frac{|q^n|^{\beta_F}}{e^{\eta^n}} - 1) + (I - \Delta t\nabla \cdot (\epsilon\nabla\cdot))\eta^n \\
q \cdot n = 0 \quad on\ \delta\Omega \times [0, T] \\
\eta(x, 0) = \log(\mu_0(x))
\end{aligned} \tag{5.16}$$

5.4.2 Finite Volume Equations of Grad Formulation

The process of sec. 5.4.1 can also be applied to the new model propsed in eq. 5.9.

Recall the properties:

$$q = \mu \nabla u \quad \mu = e^{\eta} \tag{5.17}$$

Note: If we need to rewrite the equation to not include u, the ∇u could be rewritten according to the property of q as follows:

$$q = -e^{\eta} \nabla u \quad \implies \quad \nabla u = -\frac{q}{e^{\eta}} \tag{5.18}$$

$$\begin{aligned}
\implies \frac{d}{dt}\mu &= |q|^{\beta_F} - \mu - \gamma \mu e^{\alpha(|q|\frac{\nabla x}{\mu} - \beta)} \\
ln(\frac{d}{dt}\mu) &= ln(|q|^{\beta_F}) - ln(\mu) - ln(\gamma \mu e^{\alpha(|q|\frac{\nabla x}{\mu} - \beta})) \\
e^{ln(\frac{d}{dt}\mu)} &= e^{ln(|q|^{\beta_F})} - e^{ln(\mu)} - e^{ln(\gamma e^{\alpha(|q|\frac{\nabla x}{\mu} - \beta}))} \\
\frac{d}{dt}e^{\eta} &= |q|^{\beta_F} - e^{\eta} - \gamma e^{\eta} e^{\alpha(|q|\frac{\nabla x}{e^{\eta}} - \beta)} \\
\frac{1}{e^{\eta}}\frac{d}{dt}(e^{\eta}) &= \frac{|q|^{\beta_F}}{e^{\eta}} - 1 - \gamma e^{\eta} e^{\alpha(|q|\frac{\nabla x}{e^{\eta}} - \beta)} \frac{1}{e^{\eta}} \\
\frac{1}{e^{\eta}}\frac{d\eta}{dt}e^{\eta} &= \frac{|q|^{\beta_F}}{e^{\eta}} - 1 - \gamma e^{\alpha(|q|\frac{\nabla x}{e^{\eta}} - \beta) - \eta + \eta} \\
\frac{d}{dt}\eta &= \frac{|q|^{\beta_F}}{e^{\eta}} - 1 - \gamma e^{\alpha(|q|\frac{\nabla x}{e^{\eta}} - \beta)}
\end{aligned} \tag{5.19}$$

The full system of equations we will end up solving in the finite volume formulation looks as follows (see sec 5.4.1 for transforming the other lines of the equation):

$$\int_{\partial\Omega} q \cdot ndS = \int_{\Omega} f dx \qquad \forall\Omega \tag{5.20a}$$

$$\frac{d}{dt}\eta = \frac{|q|^{\beta_F}}{e^{\eta}} - 1 - \gamma e^{\alpha(\frac{|q|\nabla x}{e^{\eta}} - \beta)} \qquad in\ \Omega \tag{5.20b}$$

$$q \cdot n \qquad \text{on } \partial\Omega \tag{5.20c}$$

$$\eta(x, 0) = \log(\mu_0(x)) \qquad on\Omega \tag{5.20d}$$

Solving The Coupled System Similarly to sec. 5.4.1, we would like to add some stabilization to the system. This changes eq. 5.20b) to:

$$(I - \Delta t \nabla \cdot (\epsilon \nabla \cdot))\eta^{n+1} = \Delta t(\frac{|q^n|^{\beta_F}}{e^{\eta^n}} - 1 - \gamma e^{\alpha(\frac{|q|\nabla x}{e\eta} - \beta)}) + (I - \Delta t \nabla \cdot (\epsilon \nabla \cdot))\eta^n \quad (5.21)$$

5.4.3 Finite Volume Equations of the Lambdaminus Formulation

Applying the same procedure on the **Lambdaminus Formulation** in eq. 5.11, we end up with the following finite volume model:

$$\int_{\partial\Omega} q \cdot n dS = \int_{\Omega} f dx \qquad (5.22a)$$

$$\frac{d}{dt}\eta = \frac{|q|^{\beta_F}}{e^{\eta}} - 1 - \gamma e^{\alpha(\lambda^- - \beta)} \qquad in\ \Omega \qquad (5.22b)$$

$$q \cdot n = 0 \quad \text{on } \delta\Omega \times [0, T] \qquad (5.22c)$$

$$\eta(x, 0) = \log(\mu_0(x)) \qquad on\Omega \qquad (5.22d)$$

Where again, eq. 5.22b) with added stabilization becomes:

$$(I - \Delta t \nabla \cdot (\epsilon \nabla \cdot))\eta^{n+1} = \Delta t(\frac{|q^n|^{\beta_F}}{e^{\eta^n}} - 1 - \gamma e^{\alpha(\lambda^- - \beta)}) + (I - \Delta t \nabla \cdot (\epsilon \nabla \cdot))\eta^n \quad (5.23)$$

Stencil for eigenvalues As we now discuss the flux along edges instead of cell centers, a new stencil is proposed for approximating the eigenvalues on the edges. Notice that this indexation only uses the fluxes on edges in the vertical direction, see fig. 5.1.

We can propose the following stencil for the approximation of the 2nd derivatives:

$$\begin{aligned} \frac{\partial^2 u}{dx^2} &\approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \\ \frac{\partial^2 u}{\partial y^2} &\approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} \\ \frac{\partial^2 u}{\partial y \partial x} &\approx \frac{1}{2k}(\frac{u_{i+1,j+1} - u_{i,j+1}}{h} - \frac{u_{i+1,j-1} - u_{i,j-1}}{h}) \end{aligned} \qquad (5.24)$$

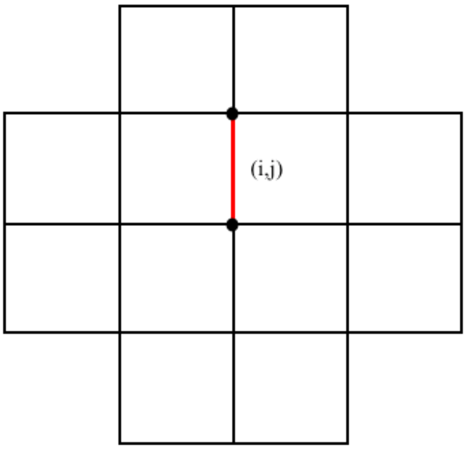


Figure 5.1: Since the flux exists on edges, $u(i, j)$ now denotes the edge marked in red. We denote i as the horizontal index, and j as the vertical index.

Where h and k denote Δx and Δy respectively. As this is again a central difference, we expect to obtain second order accuracy similar to the proposed stencil in Ch. 4.

5.4.4 Spatial Mesh

We need to establish the properties of the mesh we are solving our equations on. The following definitions will be true for all of the formulations stated previously in this chapter. In sec. 3.1, we talked about solving the Monge Kantorovich on a discrete graph structure. We will now discuss how to discretize the continuum in order to obtain a numerical solution.

Say we denote a Finite Volume mesh by the pair $\mathcal{D} = (\mathcal{T}, \mathcal{F})$, which represents the mesh Tesselation and the mesh Faces, s.t.

- The Tesselation, $\mathcal{T}$ describes a partition of the domain Ω ie. a set of non-overlapping units $K \in \mathcal{T}$.
- The faces, $\mathcal{F}$ is a set of all the faces of the partition $\mathcal{T}$.

For every unit $K \in \mathcal{T}$, we can denote its cell center as x_k, its faces as $\mathcal{F}_K$, and its diameter as a 1-dimensional measure d_k. Every face $e \in \mathcal{F}$ can be described with the 1-dimensional measure m_e. The face e has an outward normal with respect to the current cell, denoted as $n_{K,e}$, an Euclidian distance to the cell center, d^e_K, and neighboring cells $\mathcal{T}_e$. Along the boundaries $\mathcal{T}_e$ will include one element while it will have two elements for all internal cells.

We will here and throughout only focus on a 2D rectangular grid, and so many of the important properties usually looked at (such as the angle condition and connectivity of vertices) for triangular or hexagonal grids are of less importance here.

5.4.5 Solvers

The resulting system is a sparse nonlinear set of equations. The solution strategy approach is to do one Newton iteration to emit the nonlinearity and solve the resulting system with Conjugate Gradients. These procedures are both conducted iteratively and interchanging until a certain sense of time convergence is reached. More on both Newton Iteration and Conjugate gradients can be found in section A.5. Further numerical experiments will be stated in Ch. 6, s.a. discussing the convergence of the system and stability. We will also conduct comparisons between this formulation, and the ones we will begin to derive in the following chapters.

Chapter 6

Experiments

This chapter will include several numerical experiments for the proposed models of Ch. 5. The goal is to present experiments that showcase the performance of the expressions derived in earlier chapters. This chapter is structured as follows:

We will in section 6.1 see how the cost functionals alone act on images. This will both serve as complementary tests to the experiments conducted in Ch. 4 and give an insight into how the functionals change as the parameters vary.

Section 6.2 will be testing the full finite volume formulations of the extended Dynamic Monge Kantorovich with added splitting penalties. Here the stability will be shown in the form of convergence plots. Section 6.2.4 will investigate extreme parameters/term weightings whereas section 6.2.5 tests the parameters yielding a more stable convergence. In section 6.3, simulations will be run on frog tongue data using the stable parameters from sec. 6.2.

6.1 Image Filtering

As a large portion of the work was done considering the graph structure as a black and white image, experiments will be conducted on this to test the reasoning of the two models proposed in Ch. 5. Throughout this chapter, eq. 5.22 is referred to as **The Lambdaminus Formulation** and eq. 5.20 as **The Grad formulation**. Recall that Ch. 4 only covered a few examples of how the image filter acted on the images. This section will cover more scenarios and compare the two formulations. This initial section will again discuss the functionals as image filters. The derived functionals will be addressed as follows:

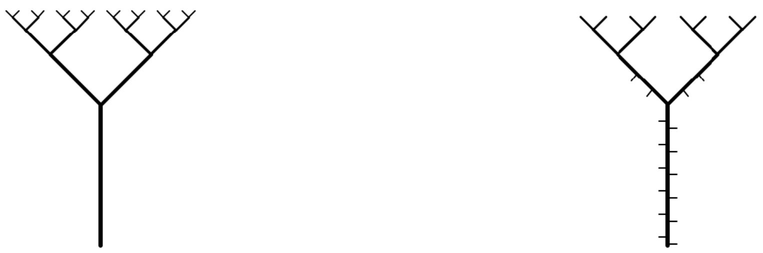
$$\textbf{The Lambdaminus Functional:} f(u) = \gamma e^{\alpha(\lambda^- - \beta)} \tag{6.1a}$$

$$\textbf{The Grad Functional:} f(u) = \gamma e^{\alpha(\nabla u - \beta)} \tag{6.1b}$$

Note: Using the same parameter names might be a little misleading as they are not the same. They are, however, describing the same properties and so the same Greek letters are being used in each formulation. The value of each parameter will be clearly stated in the following examples.

6.1.1 Experiment 1: Comparing The cost of Different Structures

In Ch. 4 we only looked at the structure of the functionals and saw that they penalized the splitting point more than the rest of the structure. This section will discuss *how much* each point is penalized.



(a) Decreasing fractal branch-like structure. (b) More unnatural looking structure.

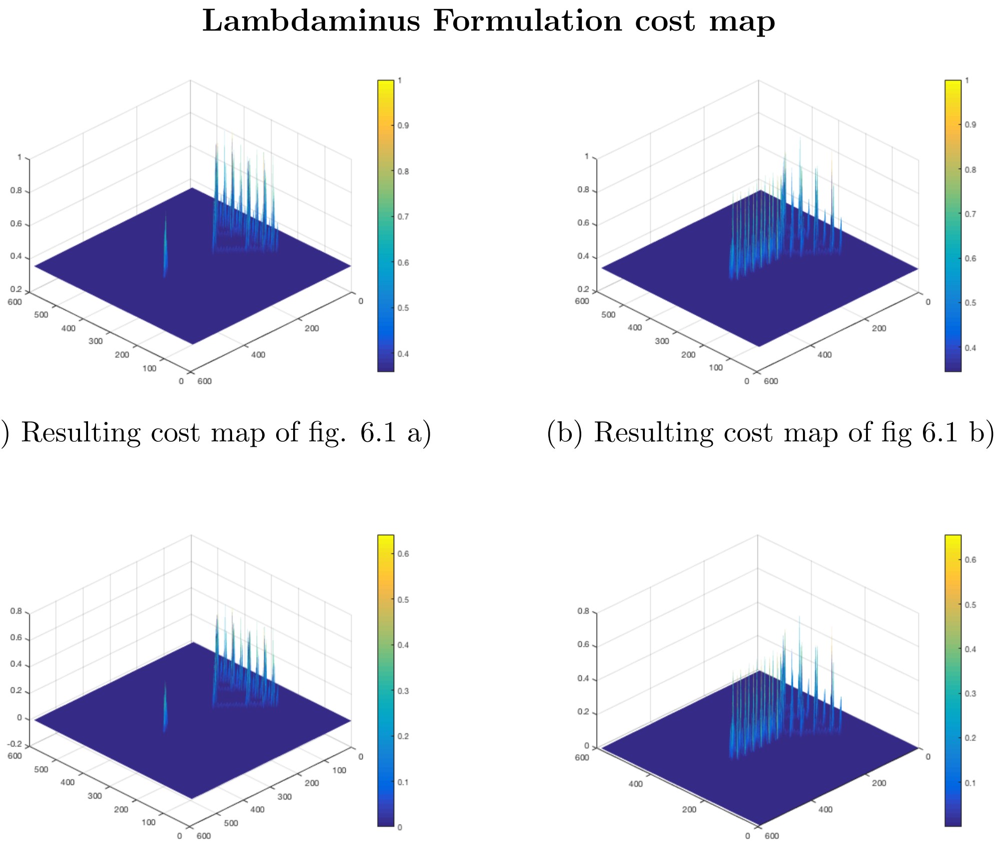
Figure 6.1: Images used for inputting into the functional.

As the initial motivation was to reduce the number of small branches stemming from the main branch, we need to test whether such a structure would yield a higher cost overall. Recall that we are minimizing a function, and so, we expect it to always opt for the cheapest alternative. We then need to test whether the ideal looking branch is the cheapest alternative.

Take two structures as in figure 6.1. They have the same number and sizes of lines (*branches*). The thing that sets them apart is that figure 6.1 a) has the smallest lines at the very ends of the branches whereas figure 6.1 b) has the smallest lines along the main stem. From the definitions of Optimal Transport in Ch. 2, one can expect the cost of fig. 6.1a) to be cheaper.

Parameter	Lambda Formulation	Grad Formulation
u_1	fig. 6.1 a)	fig. 6.1 a)
u_2	fig. 6.1 b)	fig. 6.1 b)
γ	1	1
α	10	1
β	$\max(\lambda^-)$	$\max(\nabla x \nabla u)$
$f(u)$	eq. 6.1a)	eq: 6.1b)

Table 6.1: Table of parameters used in experiment 1



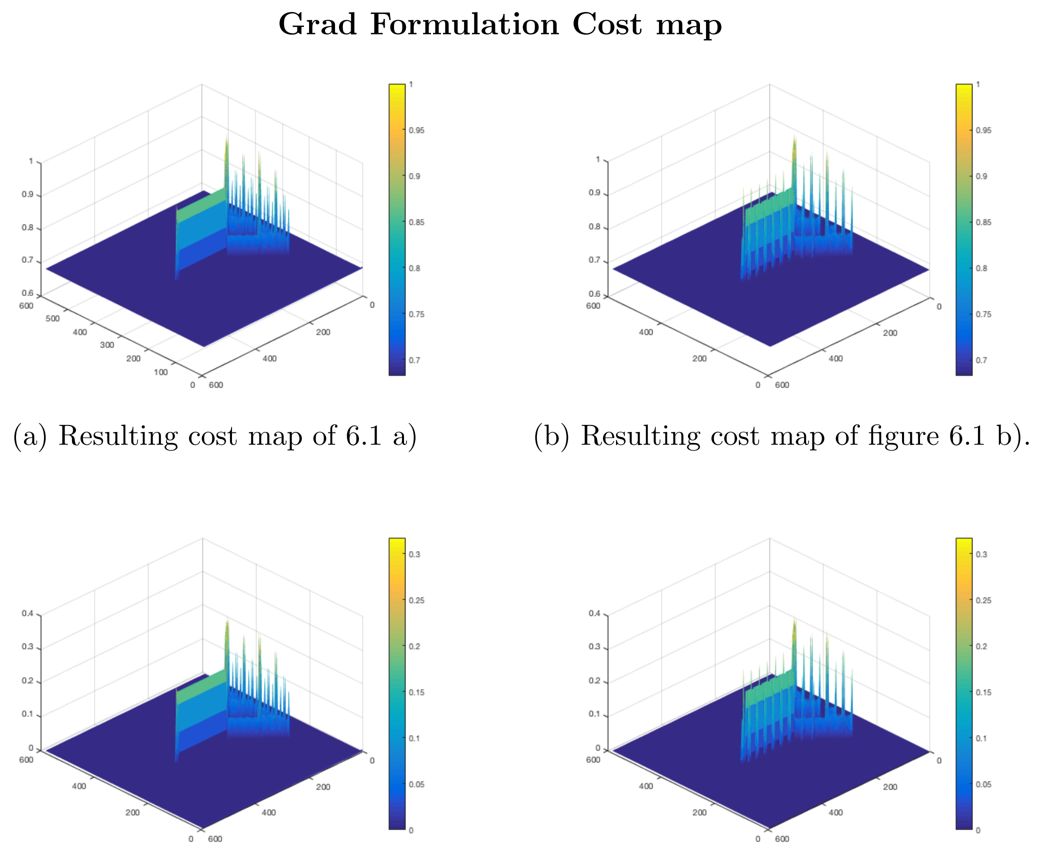
(a) Resulting cost map of fig. 6.1 a)

(b) Resulting cost map of fig 6.1 b)

(c) Total cost: $C = 467.9701$

(d) Total cost: $C = 549.5614$

Figure 6.2: Illustration of returned cost map using the Lambdaminus formulation, see table 6.1. Bottom figures are scaled to start at the zero axis in order to easier compare the two.



(a) Resulting cost map of 6.1 a)

(b) Resulting cost map of figure 6.1 b).

(c) Total cost: $C = 803.7851$

(d) Total cost: $C = 829.4769$

Figure 6.3: Cost map return using gradient formulation on the two figures 6.1. The top two images are directly returned while the bottom two are scaled to start at zero. This is to simplify the comparison between them, but would not influence the overall solution. Note the added noise/inconsistency along the main stem.

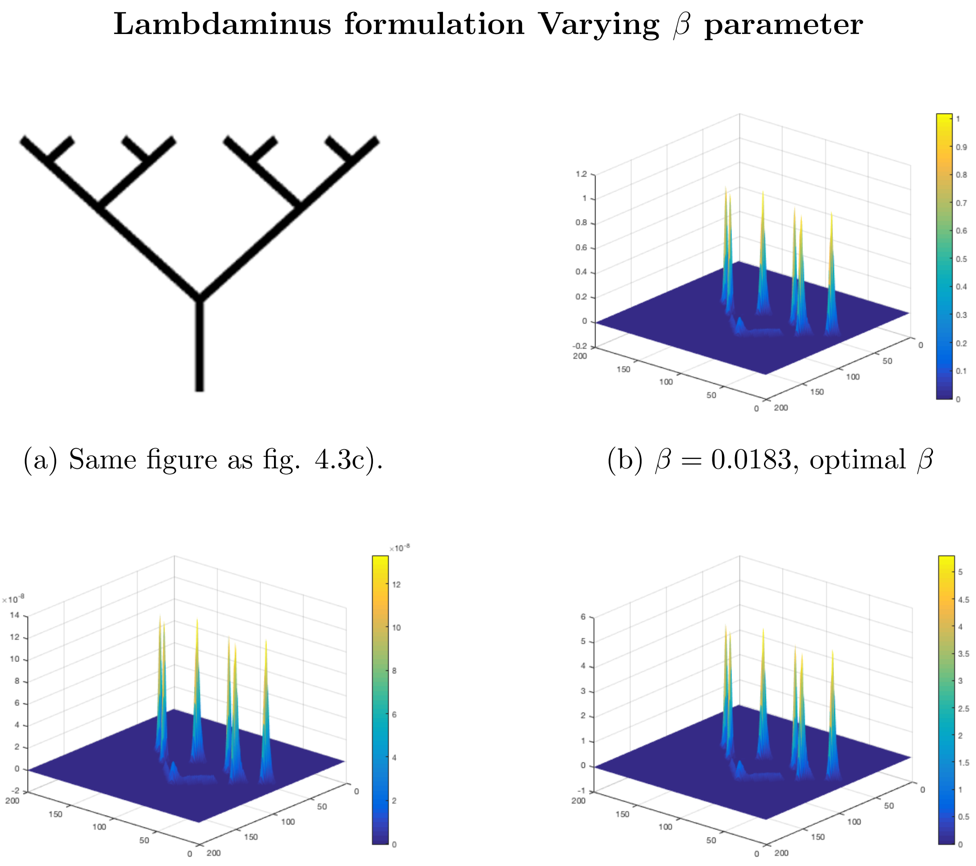
From figures 6.2 and 6.3, we see that both formulations return an increased cost from figure 6.1 b). The figures are the outputs of the functionals respectively when using the parameters listed in table 6.1.

6.1.2 Experiment 2: Varying β parameter

In this section, we will look at what happens to the structures produced when we change the parameters. This will also serve as an explanation of the reasoning behind choosing the exact parameters used in the previous examples, and the ones conducted in sec. 4.3. The following table include the inputted parameter values used to produce figures 6.4 and 6.5.

Parameters - Experiment 2		
Parameter	Eigenvalue Approach	Gradient Approach
u	fig. 6.4 a)	fig 6.5 a)
γ	1	1
σ	5	5
α	500	500
β	Varying, see fig. 6.6	Varying

Table 6.2: Table of parameters used in experiment 2



Lambdaminus formulation Varying β parameter

(a) Same figure as fig. 4.3c).

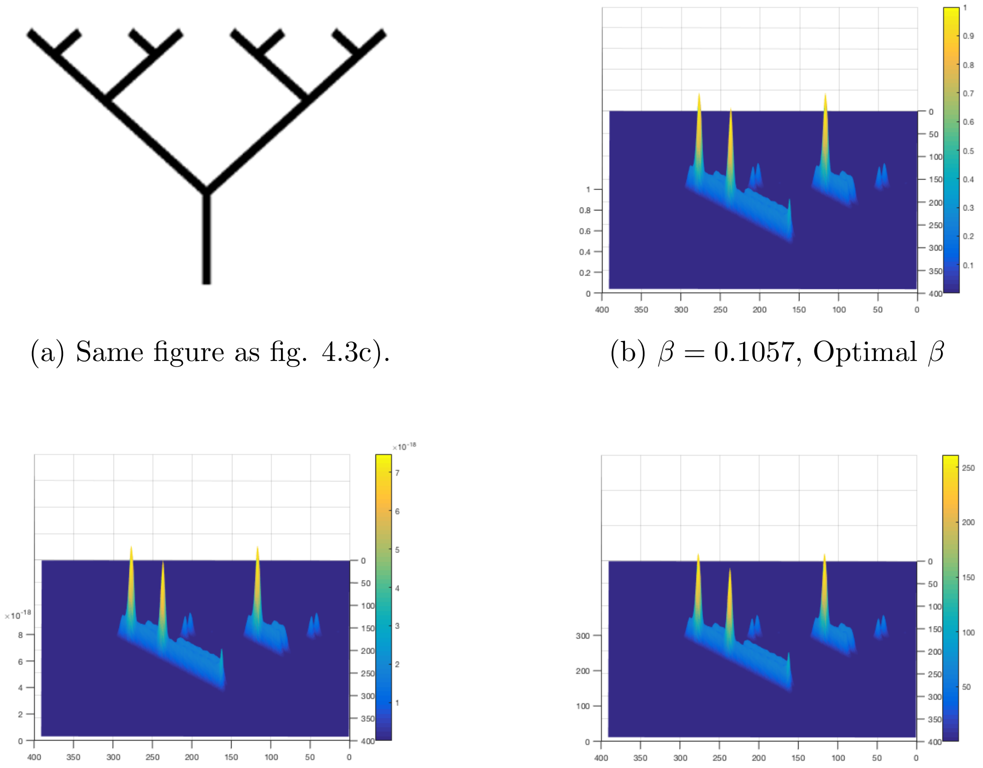
(b) $\beta = 0.0183$, optimal β

(c) $\beta = 0.05$

(d) $\beta = 0.015$

Figure 6.4: Image in 6.4a) and the returning output of the Lambdaminus functional with varying the β parameter. The optimal β value in image 6.4b represents the β value yielding a max output value 1.

Grad Formulation Varying β parameter



(a) Same figure as fig. 4.3c).

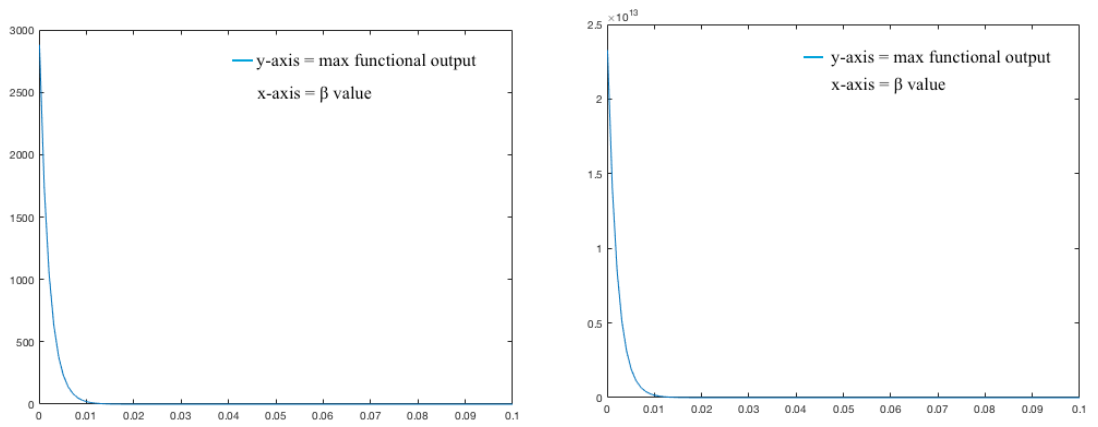
(b) $\beta = 0.1057$, Optimal β

(c) $\beta = 0.5$

(d) $\beta = 0.05$

Figure 6.5: Inputted image, fig. 6.4a), and the returning output of the Grad functional with varying the β parameter. The optimal β value in image 6.5b) represents the β value yielding a max output value 1.

From figures 6.4 and 6.5, one can notice that the β parameter only changes the order of magnitude of the structure, not the overall shape of it. Therefore, it will penalize the splitting points the most regardless of the value of β.



(a) Lambdaminus formulation plot.

(b) Gradient Approach plot.

Figure 6.6: A plot of β values along the x-axis vs. the maximum output of the functionals along the y-axis. Note that the only β values that yields a reasonable output (ie. a constant size C), are the β values used in figures 6.4 and 6.5.

Notice that the span of β values used in figures 6.4 and 6.5 is not very large. Figure 6.6 demonstrates what happens to the magnitude of the functional output of β values outside this interval. When coupling this to the rest of the model in section 6.2, the optimal β threshold parameter will automatically limit the max output to 1. We can then not expect that the optimal value of β in the following examples will correspond to the ones presented on these images, as they are solely dependant on the input.

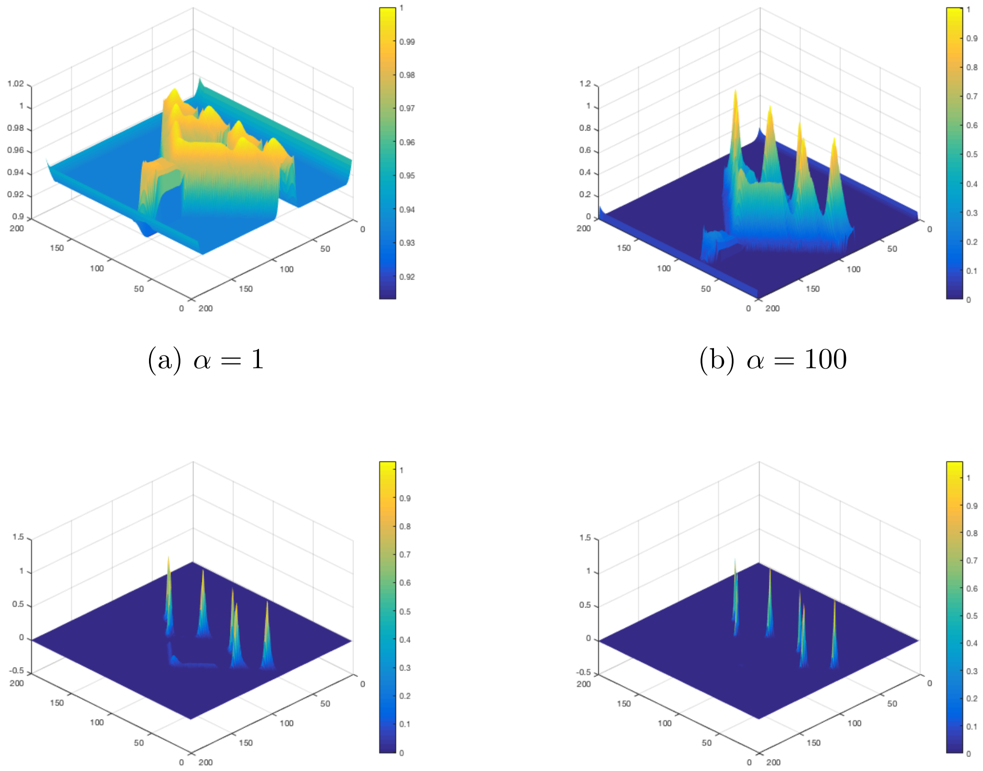
6.1.3 Experiment 3: Varying α parameter

This experiment uses the already optimal value of β discussed in sec. 6.1.2. Now we will see how the α parameter changes the outputs of the functionals. Figures 6.7 and 6.8 shows the resulting functional outputs given the parameters in table 6.3.

Parameters - Experiment 3		
Parameter	Eigenvalue Approach	Gradient approach
u	fig. 6.5a)	fig. 6.5a)
γ	1	1
α	Varying	Varying
β	0.0183	0.0656
$f(u)$	eq. 6.1a)	6.1b)

Table 6.3: Table of parameters used in experiment 3.

Lambdaminus Formulation Varying α parameter



(a) $\alpha = 1$

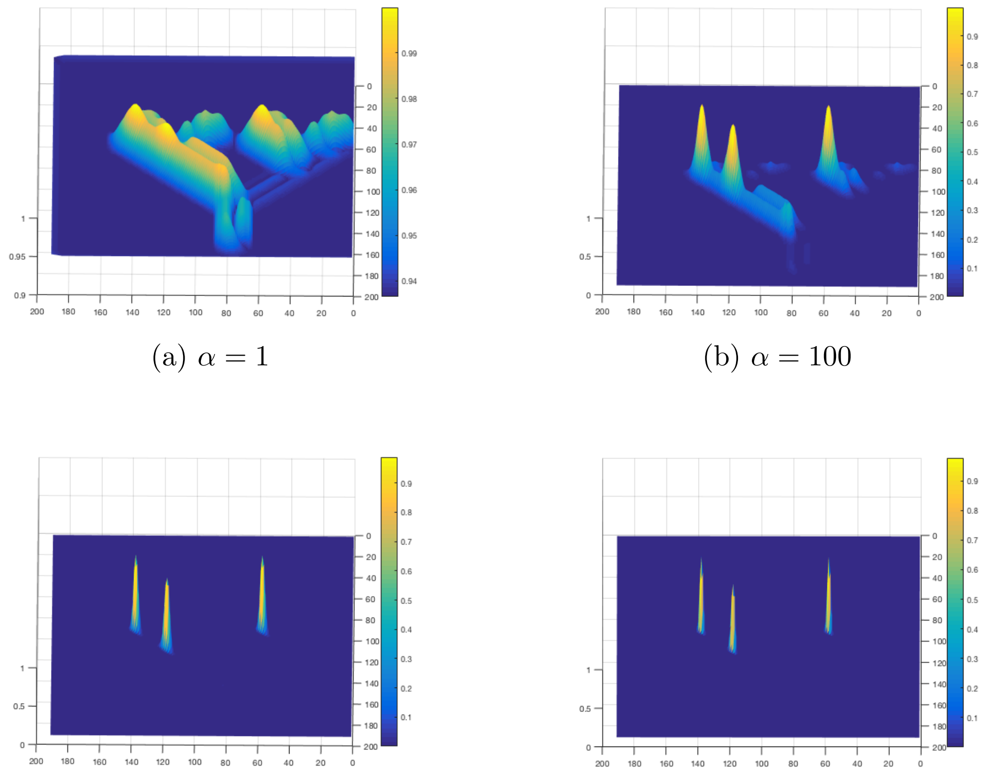
(b) $\alpha = 100$

(c) $\alpha = 500$

(d) $\alpha = 1000$

Figure 6.7: Returned Lambdaminus functional output with varying α parameter. Plots a)-d) all have a max value of 1, but their structure is different.

Grad Formulation Varying α parameter



(a) $\alpha = 1$

(b) $\alpha = 100$

(c) $\alpha = 500$

(d) $\alpha = 1000$

Figure 6.8: Returned Grad functional output with varying α parameter.

From the figures 6.7 and 6.8, we see that increasing the α parameter focuses the cost penalty to only the splitting points found. The problem, however, is that they find different points. This will be discussed more closely in sec. 7.1.1.

6.2 Testing the FV Models

In this section we will set up a small test problem that runs both the eDMK model 5.16 with a finite volume solution method described in sec. 5.4 , and the newly full derived models 5.22 and 5.20, with the added splitting penalties. The goals are to see on a small scale if any changes happen to the produced optimal maps found by the models using the same data and through this section become confident in using the models on a higher resolution image in section 6.3. Both in the fact that it looks like our desired results, and that we have a sense of numerical stability and convergence.

6.2.1 Setup

The following experiment runs a small test problem. This was mainly done to increase total run-time, but also keeping the scale large enough to visually see the forming branching properties from the models. It was also important to re-scale images to a rectangular grid to ensure that the code would be general enough to handle any grid size.

The input data is a point source, an equally distributed sink map with a total value equal to the sink, and an initial configuration map for the flow. As we can see in fig. 6.9, the initial flow map is chosen at random.

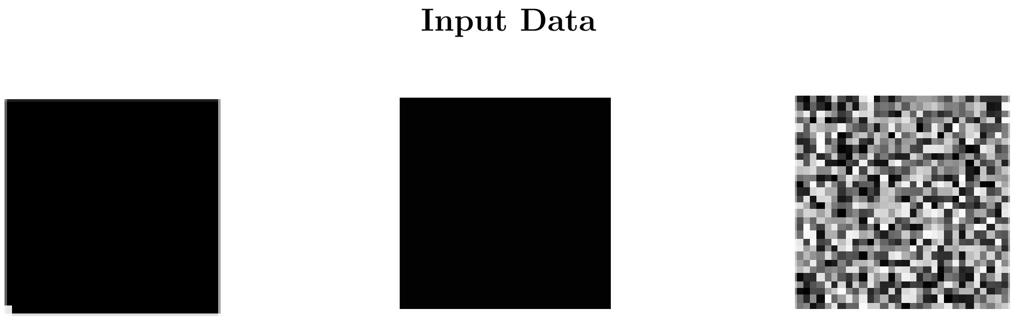


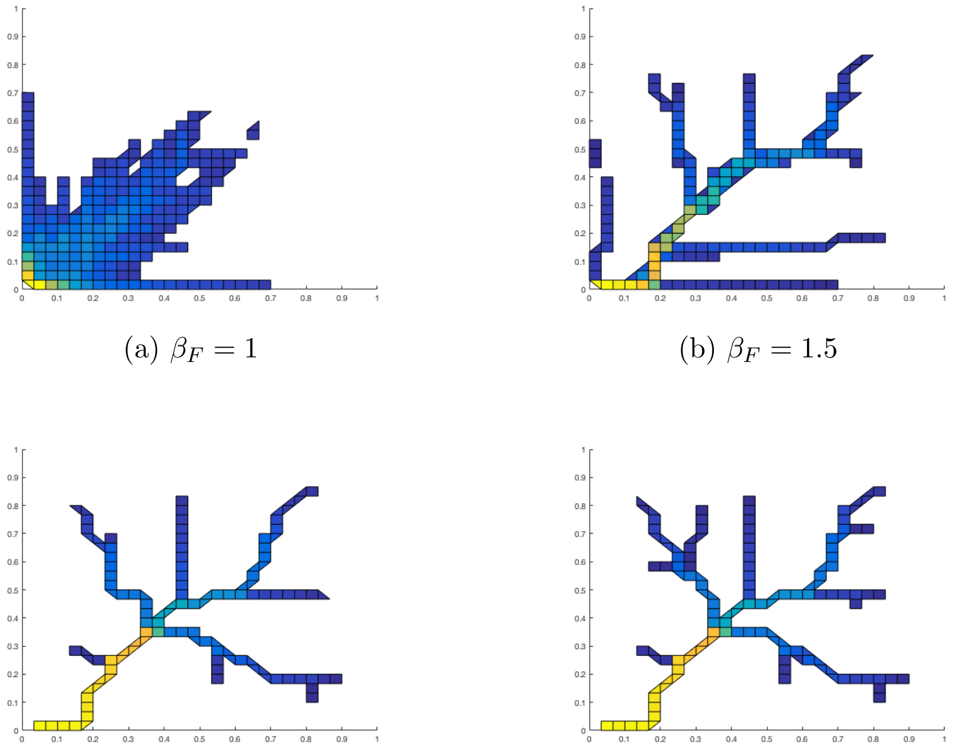
Figure 6.9: Images of the input data specified. From left to right: The source, the sink and the initial flow map.

Note: The use of a random configuration like this will act a little differently every time. The main point, however, is to compare the different formulations which were all done using the same configuration. We can, therefore, expect to obtain a slight difference in our configurations every simulation. This difference might also be due to the non-uniqueness described in section 3.2.2 in addition to the randomness in data.

6.2.2 Testing The β_F parameter

Figure 6.10 is the given output for different values of the β_F parameter. This is first to test that it works with the finite volume formulation and to test how changing the β_F parameter changes the output. Again, see [Facca, 2017] for more on this parameter.

Regular FV DMK with varying β_F parameter



(a) $\beta_F = 1$

(b) $\beta_F = 1.5$

(c) $\beta_F = 1.85$

(d) $\beta_F = 2$

Figure 6.10: The Finite Volume Test problem from eq. 5.16 using same initial data but different values of β_F.

The plots above obtain the results we expected according to [Facca, 2017]. The code therefore also works while using the Finite Volume formulation. We can then test with inputting the splitting penalty conditions. Throughout the rest of the experiments, we will be using $\beta_F = 1.85$.

Note: It is worth to mention that there is a threshold set to which parts of the figure are visible. This is because we are mostly concerned about the overall structural changes in the figures. This threshold will be kept consistent throughout this section.

6.2.3 Testing Added Splitting Penalties

Based on the parameters tested in sec. 6.1, we run the models 5.16, 5.22 and 5.20 using the parameters from table 6.4. The computed solutions of the different models are plotted in fig. 6.11.

Parameters			
Parameter	Regular DMK	Eigenvalue Approach	Gradient approach
γ	-	1	1
α	-	1	1
β_F	1.85	1.85	1.85
β	-	Optimal	Optimal
Equation	eq. 5.16	eq. 5.22	eq. 5.20

Table 6.4: Table of parameters for testing the proposed models.

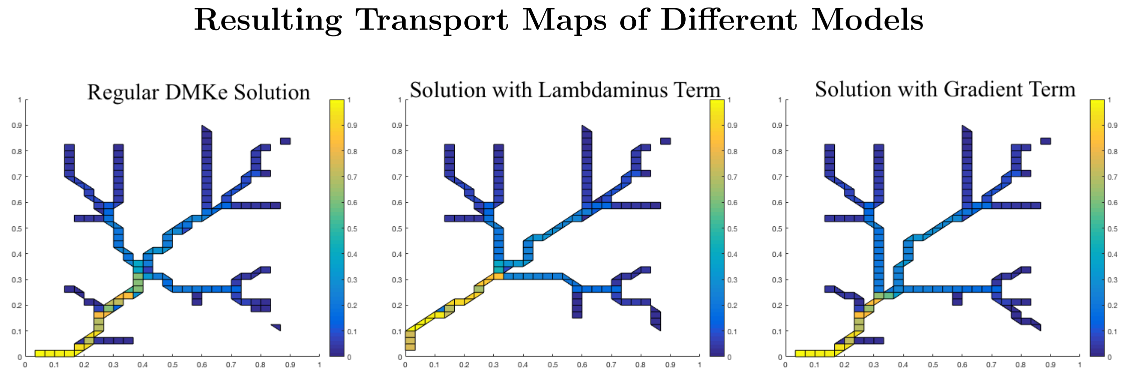


Figure 6.11: Resulting plots of first testing the splitting penalty properties. The plots obtained by the different models are figured from left to right: The Regular DMKe, The Lambdaminus Formulation and The Grad Formulation.

From fig. 6.11 we see that the new models keep a similar base structure as the original, but both of our proposed formulations have managed to remove small branches along the main stem. It does appear that the Lambdaminus formulation was more successful in removing the larger unwanted branches. This will be discussed more thoroughly in sec. 7.1.

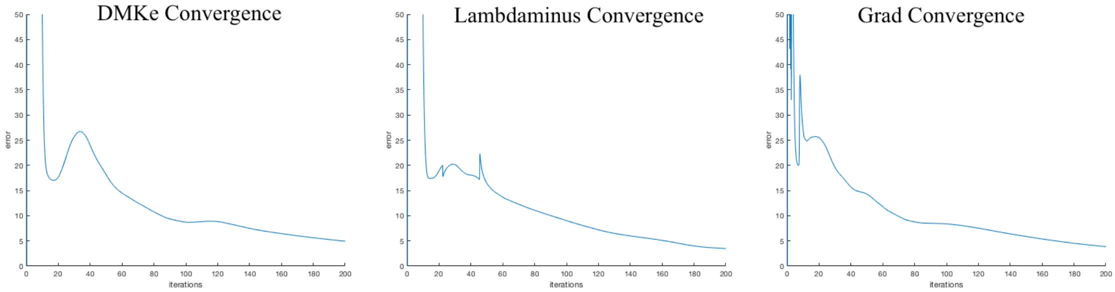


Figure 6.12: Returned Convergence, difference between two consecutive time steps.

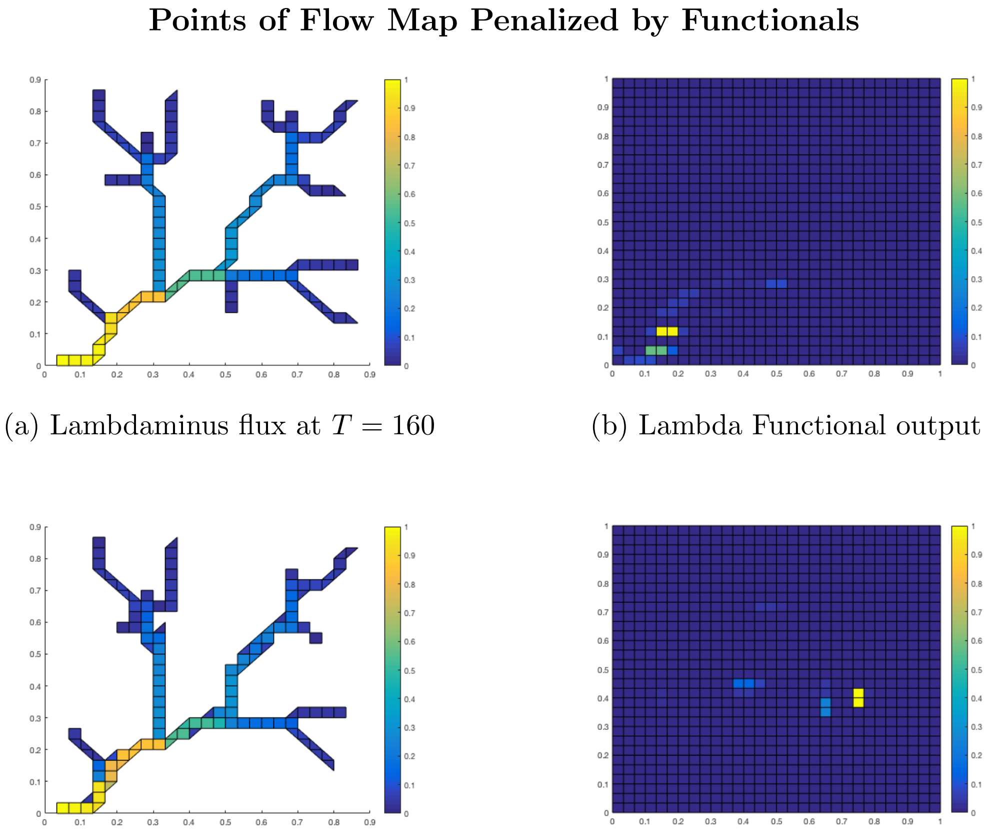
By looking at the convergence plots in fig. 6.12, we see that the added terms do add some inconsistencies in the convergence rate. Such sharp peaks are a concern as this could quickly spike the values resulting in an ill-posed system. Figure 6.12 shows that the grad formulation is more sensitive to this than the lambda formulation.

From the fig. 6.13, we see that the functionals have only picked out the largest point to penalize. This is due to the fact that we have large differences in the points. In contrast to the image example, we have not here applied the Gauss filter to the initial conditions. The contrast between the largest point found and the remaining points may then get very large, which is the reason that we only see one point on the plots in fig. 6.13.

To correct this, we can use the Gauss function on the transmissibility map itself. The Gauss function looks as follows:

$$G(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{6.2}$$

Applying this with standard deviation, $\sigma = 5$ as used in the image examples in sec. 6.1, should yield a more similar map. See fig. 6.14 below for new functional maps. One can also argue that by the property of exponential functions, this standard deviation could be accounted for in the already existing α parameter, which has previously been regarded as a *sharpening* parameter. By setting it to a value lower than 1 it transforms into a diffusing parameter.



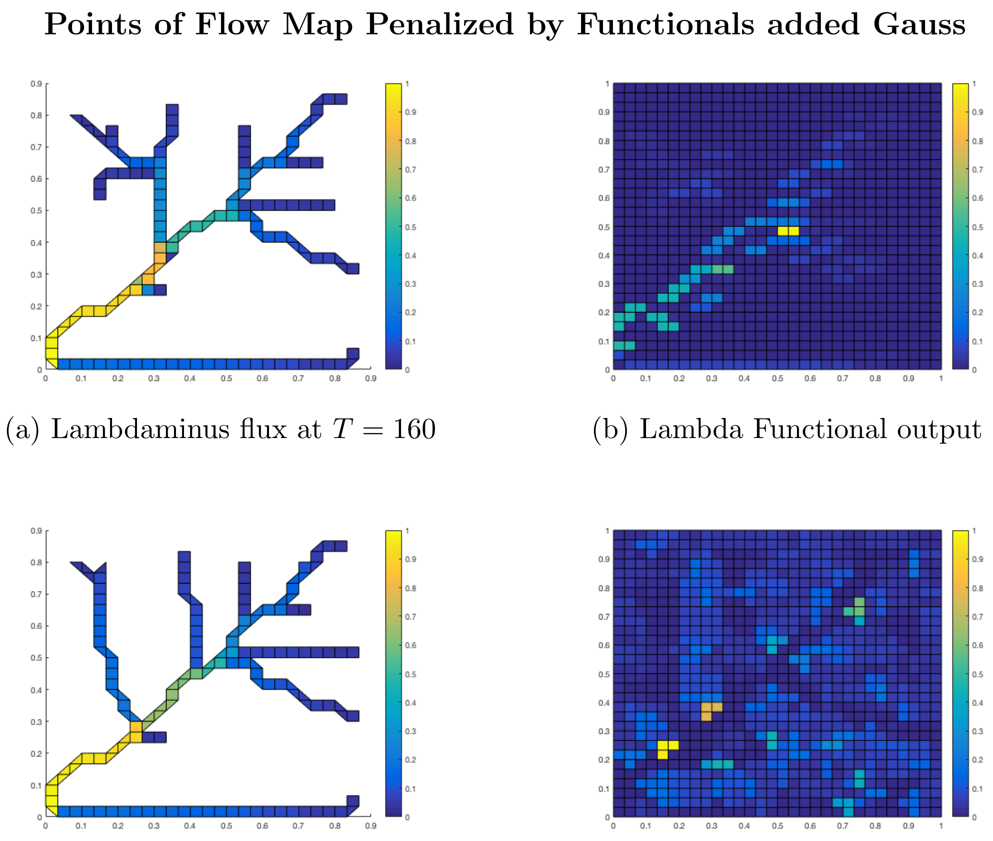
(a) Lambdaminus flux at $T = 160$

(b) Lambda Functional output

(c) Grad flux at $T = 160$

(d) Grad Functional Output

Figure 6.13: The flow maps and their respective functional output. Note that they are penalizing different points due to discretization. Also note that at $T = 160$, the systems have already started converging which is why the structures are slightly different even when using the same input data.



Points of Flow Map Penalized by Functionals added Gauss

(a) Lambdaminus flux at $T = 160$

(b) Lambda Functional output

(c) Grad flux at $T = 160$

(d) Grad Functional Output

Figure 6.14: The flow maps and their respective functional output. This is after adding a Gaussian filter.

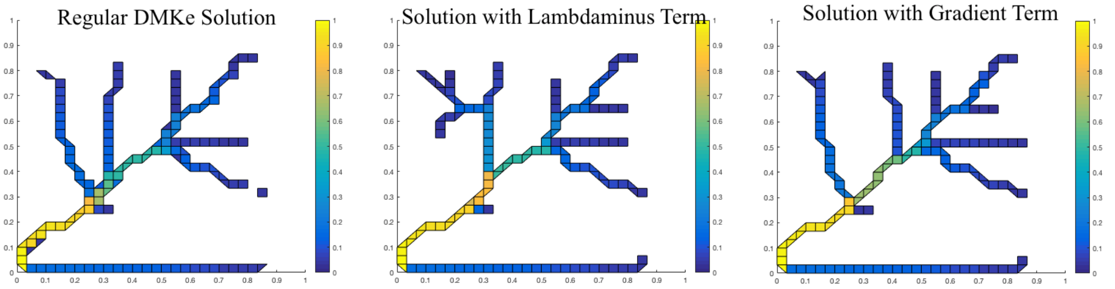


Figure 6.15: Resulting plots of testing the splitting penalty properties with added Gauss filter.

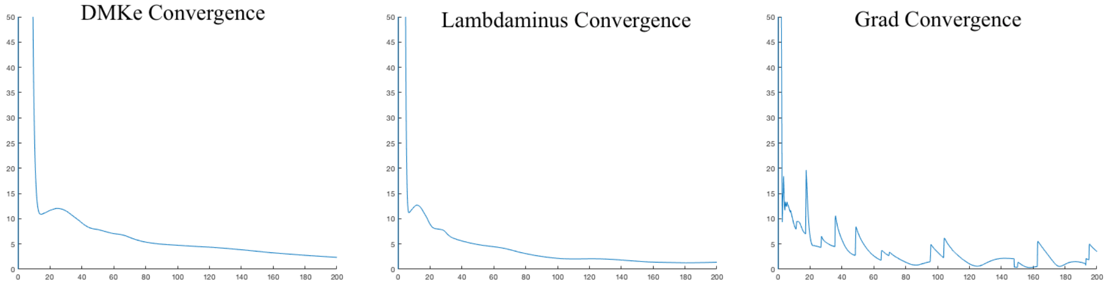


Figure 6.16: Convergence plots of different models.

6.2.4 Testing Extreme Parameter Values

Here we will test with adding the largest weighing of γ possible and still running. **Note:** The random configuration of the initial data creates uncertainty here, as one map might get lucky and run. What we mean by the largest weighing of the term, is the one that consecutively runs. The parameters in table 6.5 were the largest ones to still run.

Parameters			
Parameter	Regular DMK	Eigenvalue Approach	Gradient approach
γ	-	$\gamma = 5$	$\gamma = 15$
α	-	$\alpha = 0.000001$	$\alpha = 0.0001$
β_F	1.85	1.85	1.85
β	-	Optimal	Optimal
Equation	eq. 5.16	eq. 5.22	eq. 5.20

Table 6.5: Table of parameters used to test max weight of extra terms.

These had the resulting convergence plots shown in fig. 6.18.

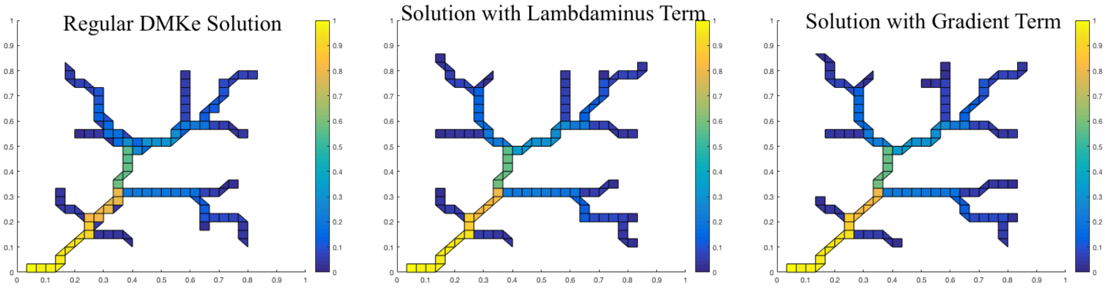


Figure 6.17: Resulting plots of testing highest splitting penalty weight.

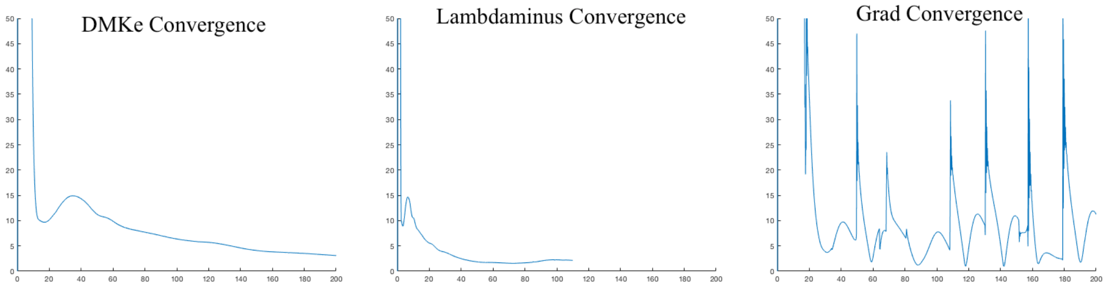


Figure 6.18: Convergence plots of different models for highest possible weighing of penalty term.

Note: That although the Grad Formulation has the most unstable convergence map with high periodic spikes (see fig. 6.17 c)), the Lambdaminus Formulation did not manage to run until final time even with a much smoother map.

6.2.5 Testing Stable Parameter Values

Here we will focus on running models that alter the initial image to the desired outcome, while still maintaining a stable enough convergence plot. These are the parameter values that will be used for the run on a larger scale simulation. See tab. 6.6, fig. 6.19 and fig. 6.20.

Parameters			
Parameter	Regular DMK	Eigenvalue Approach	Gradient approach
γ	-	2	2
α	-	0.000001	0.0001
β_F	1.85	1.85	1.85
β	-	Optimal	Optimal
Equation	eq. 5.16	eq. 5.22	eq. 5.20

Table 6.6: Table of parameters shown successive simulations of semi-stable convergence.

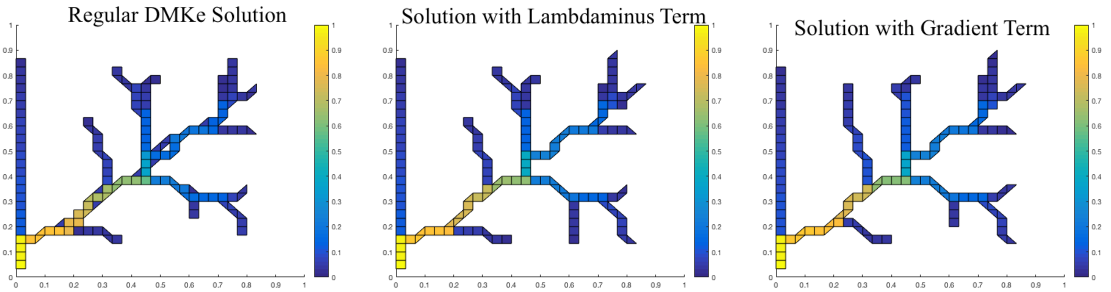


Figure 6.19: Resulting plots of first testing the splitting penalty properties.

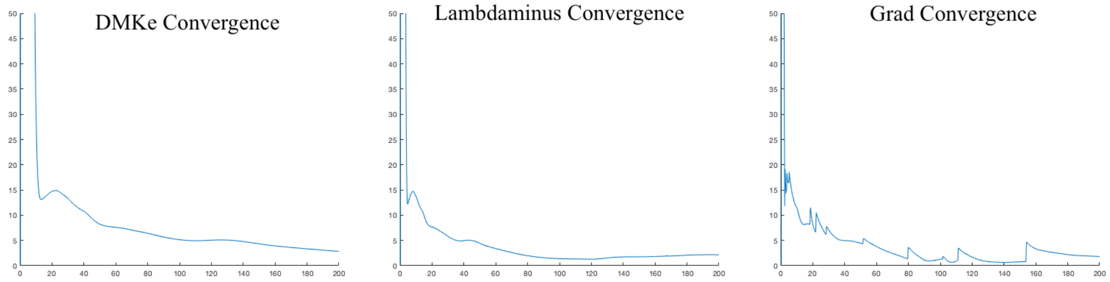


Figure 6.20: Returned Convergence using more stable parameters.

6.3 Applying to Frog Tongue Data

This section will test how the models built in previous chapters performs at simulating the vascular structure of real input data. We will run tests using the most stable parameters testes in section 6.2. See table 6.6 for a list of these parameters and values.

6.3.1 Data

The data used for setting up this problem is the image of the frog tongue from [Cohnheim, 1872] segmented by [Hanson and Lundervold, 2013]. The results are the drawn-out vein characteristics used as initial data seen in fig. 6.21 b).

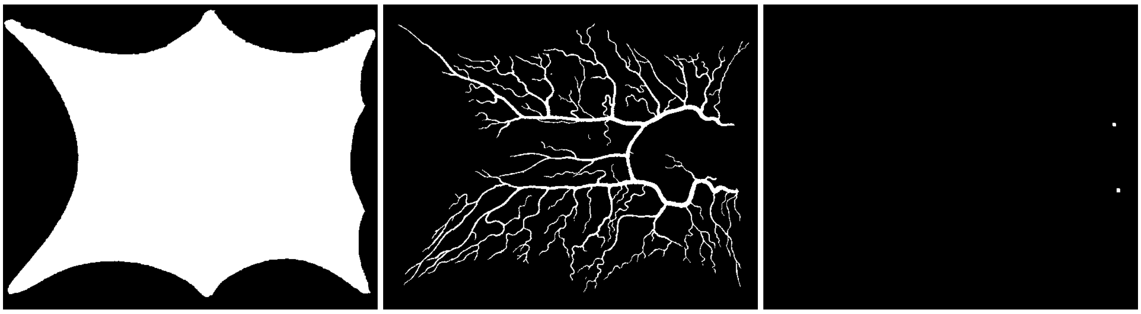


Figure 6.21: Images of the input data specified. From left to right: The domain, the initial flow map and the source.

6.3.2 Simulation

Fig. 6.22 shows the returned structures calculated by the model with a viewpoint 0.7, ie. only the main branches are visible on the plot.

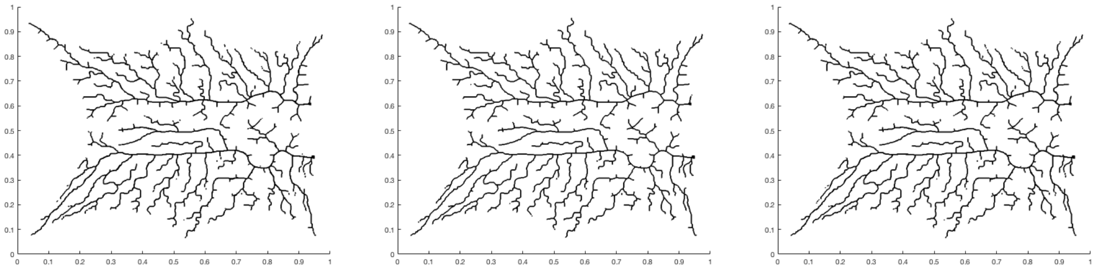


Figure 6.22: Plots of the main structures returned by models. From left to right: The eDMK, The Lambdaminus Formulation and The Grad Formulation.

All of the figures resemble the input data, but there is no visible difference made by the added penalties on this threshold. Getting a closer look by setting the visibility to 0.01:

A zoomed-in version of this plot can be found in figure 6.24, where one can notice a difference between the models.

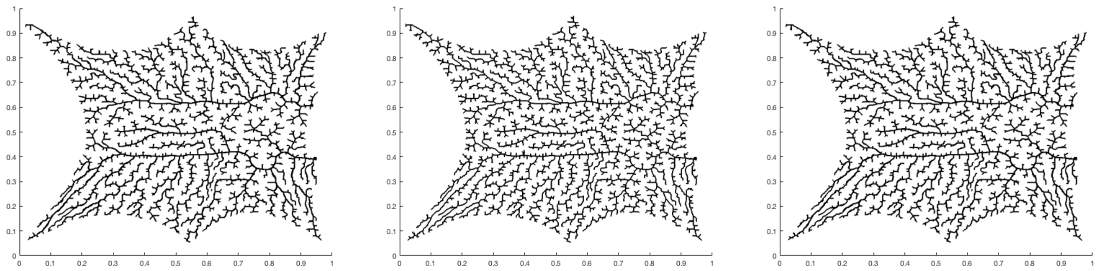


Figure 6.23: Viewpoint 0.01. From left to right: The eDMK, The Lambdaminus Formulation and The Grad Formulation.

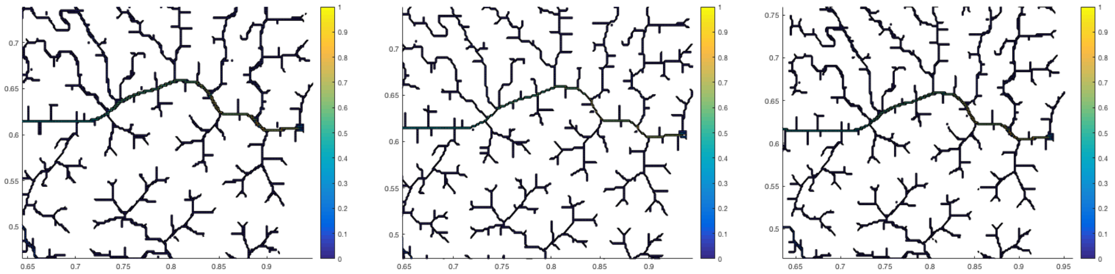


Figure 6.24: Zoomed in version of previous figure. From left to right: The eDMK, The Lambdaminus Formulation and The Grad Formulation.

6.3.3 Random Initial Data

As the previous experiment seems to end up with a structure closely resembling the initial data, we now look at what happens when we initialize at random, similarity to what was done in section 6.2.

Note that also here the same random configuration is used in all formulations. The plots in fig. 6.26 are the returned results of the simulation.

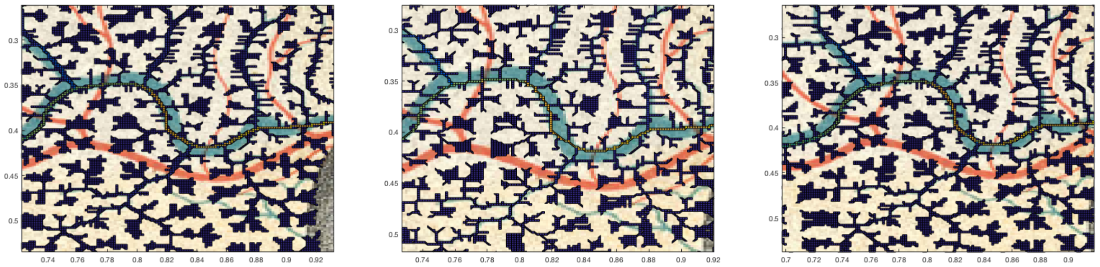
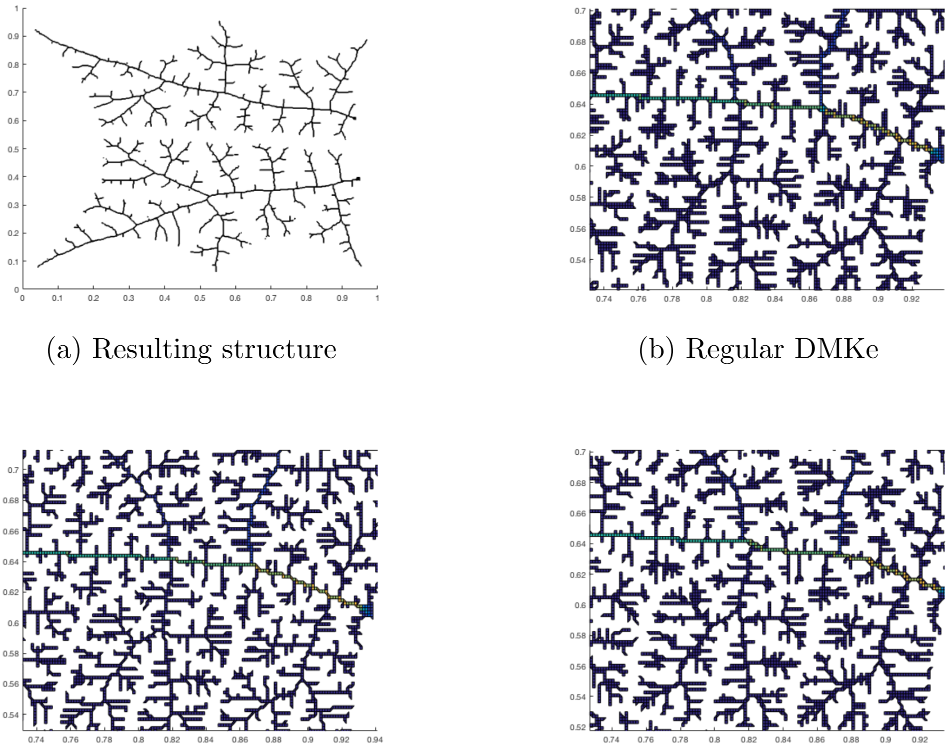


Figure 6.25: Zoomed in version of previous figure mapped onto vascular image. From left to right: The eDMK, The Lambdaminus Formulation and The Grad Formulation.

Frog Tongue Simulation With Random Initial Data



(a) Resulting structure

(b) Regular DMKe

(c) Lambdaminus Formulation

(d) Grad Formulation

Figure 6.26: The resulting structures from running on the frog tongue using a random initial configuration of the flow pattern. Figure a) is the overall resulting structure by limiting the viewpoint. This structure is equal across formulations, similar to what was shown in fig. 6.22. Figures b)-d) show a zoomed-in version of the same locations in the images. Again, notice that the two new formulations yield a refinement of the original one.

Chapter 7

Discussion

This chapter serves as a discussion and summary of the experiments in Ch. 6. We will discuss how the experiments turned out versus the initial goal, and include a list of relevant limitations. This chapter is sectioned as follows:

Section 7.1 will discuss the overall performance of the new models relative to the eDMK. It will discuss the results of each section in the previous chapter separately. Section 7.2 summarises the limitations encountered, which give rise to further work of interest within this topic. Finally, section 7.3 provides the conclusion of the thesis.

7.1 Discussion

7.1.1 Discoveries made from Image Filtering Experiments

In sec. 6.1, experiments were done to see whether the functionals could detect splitting points in an image. Both of the functionals were successful in returning a higher cost at the splitting points compared to the rest of the image.

By further comparison between figures 6.4 and 6.5, we see that the Grad Formulation only has detected 3/6 of the splitting points, whereas the Lambdaminus formulation has targeted 5/6 of all splitting points in the image. This is due to the discretization, as the Grad Formulation only approximates the gradient by two points, whereas the eigenvalues are approximated by second derivatives using a central difference scheme. This leads to a higher directional dependency for the Grad Formulation, which is the reason why it does not find the same amount of points. Note that this is due to the discretization, not the continuous equation itself. The effect of the discretization weakness can also be seen in fig. 6.13. It would be interesting to see how a better-suited discretization would improve this.

As for the parameters, both the α and β parameters were shown important to the output of the functional. In Ch. 4 we mentioned that the β is always automatically set to the max value and acts as a threshold to limit the max output of the functional to 1. The downside to this, however, is that it will still penalize the minimum point in any image even though this point might not represent a splitting point. There has not been put a huge emphasis on this here, as the natural branching properties of the rest of the model always has shown to yield branches spanning the domain.

7.1.2 Discussing Test Problem Cases

From section 6.2, both of the new formulations show signs of removing small details otherwise included in the finite volume solution of the eDMK. Note that all experiments conducted in this section ran until $T = 200$. This was done mainly from the convenience of maintaining a consistent size throughout, but also to see how the convergence plots acted over time. Even by setting the stopping criteria as $tol = 1$, did not strike in before $T = 200$. All resulting structures end up having a line crossing the domain diagonally. This relates to optimal transport, as the smallest path between two points is straight through.

This section focused on varying the α and γ parameters, as β was already determined from sec. 6.1. After adding a small diffusion to the problem to both improve numerical stability and increase the amounts of points penalized, the test cases showed the ability to limit the number of small flow outside of the main structure. The main structure, however, stayed roughly the same.

It can be seen in fig. 6.18, that the simulation only ran for small enough γ, as the larger weightings made the model unstable. From fig. 6.18, we see that even for the γ tested, the Lambdaminus Formulation was unable to run until completion. This suggests that perhaps the wrong solution strategy was chosen for this term, or that the extreme cases do not yield a solution.

Aside from the computational stability, the extreme cases would be very interesting to look at. We can imagine that a structure heavily focused on the no-splitting-conditions would not include any splitting points at all, as these points would be very expensive to build. This might yield zig-zag structures spanning the whole domain. Again, the automatic β could create problems in this case.

7.1.3 Frog Tounge Simulation

By running the new models using the frog tongue data in sec. 6.3, we notice that the overall structure stays the same for all formulations. By a closer comparison in fig. 6.25, one can notice differences made on a more detailed level. The differences made here correspond to the results seen from the previous examples. As a result, both of the models were able to refine the structure and end up with more defined branches.

To ensure that these results do not only come from rigid initial conditions, section 6.3.3 initialized the same model with a random flow map. The results in fig. 6.26, show that the branch structure turned out a little differently than fig. 6.24, but all the models again yielded the same overall structures. By a closer examination, also here, the new models, and especially the Lambdaminus model, refined the tree structure by removing small branches next to the main stem. This is the most visible near the beginning of the structure, as this is the area focused on by the functionals.

7.2 Limitations

7.2.1 Dimensions

As discussed in Ch. 4, the proposed cost functionals are very sensitive to crossings in an image. Recall the definition of splitting point in the image sense is given in section 4.1. When defining something as a splitting point or not, it will, at least in 2D, count an overlap as an additional splitting point.

The working dimension has been 2D throughout. In the description of the grid setup in sec. 5.4.4, it becomes clear that the model will struggle to compute arbitrary 3D data. This becomes a restriction as there are many interesting 3D cases of branching structures to look at. This could be computed by running on each 2D slice individually.

7.2.2 Parameter dependency

Having a lot of parameters gives us a lot of control, but also a lot of time can be spent fine-tuning these parameters to ensure that we get the optimal output. Making them dependant on the input itself has eliminated some of these issues, but creates new ones as the functional will always retrieve the smallest point as mentioned in sec. 7.1.1.

In sec. 7.1 we discussed that the models only penalized small branches on the substructures. By changing the parameters, the models might be able to make changes to larger branches as well.

7.2.3 Discretizations

Again, one of the main limitations is that the discretizations used creates a biased directional dependence.

7.2.4 Computational Stability

The plots from sec. 6.2 show that there are some numerical stability issues. This also yields a limitation of the cases tested. Ideas to improve numerical stability is to perhaps introduce a pre-conditioner to matrix A, as the matrix might be ill-conditioned. Another thing that could improve the stability would be to lower the time-step.

7.2.5 Analytic proof

As mentioned in sec. 3.5, the regularity of the problem disappears for $1 < \beta_F < 2$. Because these are the values that show the best branching properties, and the values used throughout the experiments, we have not been able to conduct formal proof that the solution exists. We do, however, see a strong inclination towards numerical convergence, and this is the reasoning behind relying on numerical evidence.

7.3 Conclusion

The proposed functionals specify a penalty for points with large gradients or large eigenvalues. By implementing these into the existing eDMK model, the points removed are the small points next to the main branches. Results of the tests show that they yield a refinement to the branches, but not a change to the overall structure as hoped for.

The presented theory is an interesting approach to the current model of branched structures. More work needs to be done on the applications, but it is thought that by changing the parameters, the functionals will be able to influence larger branches as well. When working through some of the limitations, this approach yields a more visually optimal result. Hence, this answers the motivating question of the ability to find such a penalty function.

Bibliography

[Bonifaci et al., 2012] Bonifaci, V., Mehlhorn, K., and Varma, G. (2012). Physarum can compute shortest path. *Journal of Theoretical Biology*, (309):121–133.

[Cohnheim, 1872] Cohnheim, J. (1872). *Untersuchungen über die embolischen Prozesse*. A. Hirschwald.

[Drineas et al., 2006] Drineas, P., Kannan, R., and W. Mahoney, M. (2006). Fast monte carlo algorithms for matrices ii: Computing a low rank approximation to a matrix. *Society for Industrial and Applied Mathematics*, 36(1):158–183.

[Facca, 2017] Facca, E. (2017). *Biologically Inspired Formulation of Optimal Transport Problems*. PhD thesis, Università Degli Studi Di Padova, Padova. .

[Facca et al., 2018] Facca, E., Daneri, S., Cardin, F., and Putti, M. (2018). Numerical solution of monge-kantorovich equations via a dynamic formulation. *ArXiv*, 1709(2):8.

[Gilbert, 1967] Gilbert, E. N. (1967). Minimum cost communication networks. *The Bell System Technical Journal*, 46(9):2209–2227.

[Hamfeldt, 2019] Hamfeldt, B. (2019). Optimal transport - introduction to optimal transport.

[Hanson and Lundervold, 2013] Hanson, E. A. and Lundervold, A. (2013). Local/non-local regularized image segmentation using graph-cuts. *International Journal of Computer Assisted Radiology and Surgery*, 8(3):1073–1084.

[Johnson, 1987] Johnson, C. (1987). *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover.

[Kantorovich, 2006] Kantorovich, L. V. (2006). On a problem of monge. *Journal of Mathematical Sciences*, 133(4). Originally Published in Uspekhi Mat. Nauk, 3, No. 2, 225-226(1948).

[Kuijper, 2004] Kuijper, A. (2004). On detecting all saddle points in 2d images. *Elsevier*, 4(2):201–213.

[LeVeque, 2007] LeVeque, R. J. (2007). *Finite Difference Mehods for Ordinary and Partial Differential Equations.* Siam.

[Li, 2004] Li, J. K.-J. (2004). *Dynamics of the Vascular System*, volume vol.1. World Scientific Publishing Co.

[Moler, 2004] Moler, C. (2004). *Numerical Computing with MATLAB.* Society for Industrial and Applied Mathematics, 3rd edition.

[Monge, 1781] Monge, G. (1781). *Mémoire sur la théorie des déblais et des remblais.* De l'Imprimerie Royale.

[Nordbotten, 2019] Nordbotten, J. M. (2019). Finite volume discretization for the extended dynamic monge-kantorovich equations. Unpublished.

[Nordbotten and Celia, 2011] Nordbotten, J. M. and Celia, M. (2011). *Geological Storage of CO2*, volume vol.1. Wiley.

[Quohar et al., 2020] Quohar, U. N. A., Munthe-Kaas, A. Z., Nordbotten, J. M., and Hanson, E. A. (2020). A multi-scale flow model for blood regulation in a realistic vascular system. preprint on webpage at `https://www.researchsquare.com/article/rs-13683/v1`.

[Santambrogio, 2015] Santambrogio, F. (2015). *Optimal Transport for Applied Mathematicians*, volume 1 of *87*. Birkhäuser Basel, 1 edition. Series title: Progress in Nonlinear Differential Equations and Their Applications.

[Tero et al., 2006] Tero, A., Kobayashi, R., and Nakagaki, T. (2006). A mathematical model for adaptive transport network in path finding by true slime mold. *Journal of Theoretical Biology*, 244(4):553–564.

[Trefethen, 1997] Trefethen, L. N. (1997). *Numerical linear algebra.* Society for Industrial and Applied Mathematics.

[Villani, 2009] Villani, C. (2009). *Optimal Transport.* Springer-Verlag Berlin Heidelberg, 1st edition.

Appendices

Appendix A

Additional Theory

The Appendix includes properties and theories that are mentioned in the thesis. They are summarized here to shorten previous sections and serves as a checklist to read through before and alongside the other chapters. The sections in the Appendix are structured as follows: Section A.1 discusses matrix properties such as orientation, eigenvalues, and singular values. Section A.2 discusses norms, which relates to cost functions used in Optimal Transport. Section A.3 describes the components of the SVD and its usages for Image Processing. Section A.4 introduces Finite Differences. Finally, section A.5 will describe the solution strategies and properties for nonlinear ODEs.

The Appendix utilizes many textbooks encountered in graduate courses in applied mathematics. The main resources for the Appendix are [LeVeque, 2007], [Trefethen, 1997] and [Moler, 2004]. No proofs will be stated here, but they can be located in their sources.

A.1 Matrix Properties

A matrix $A \in \mathbb{C}^{m \times n}$ has full rank m where $m \geq n$. We also have that the **column rank** of A is n and the **row rank** m. This means that the rank of the matrix is equal to its column rank if it is linearly independent. This is summarized in the following theorems.

Theorem A.1. *A matrix $A \in \mathbb{C}^{m \times n}$ with $m \geq n$ has full rank if and only if it maps two distinct vectors to the same vector.*

Theorem A.2 (Fundamental Theorem of Linear Algebra). *For $A \in \mathbb{C}^{m \times n}$, the following conditions are equivalent:*

1. *A has an inverse* A^{-1}
2. *rank*$(A) = m$,
3. *range*$(A) = \mathbb{C}^m$,
4. *null*$(A) = \{0\}$,
5. *0 is not an eigenvalue of* A,
6. *0 is not a singular value of* A,
7. *det*$(A) \neq 0$.

Definition A.3 (Sparse). *A matrix A is said to be sparse if many of its entries are said to have a magnitude lower than machine epsilon.* ⌟

Definition A.4 (Symmetric). *A matrix* A *is said to be symmetric if* $A = A^T$, *where* A^T *denotes the transpose of the matrix* A. ⌟

Definition A.5 (Positive Definite Matrix). *A matrix A is said to be Positive Definite if* $\lambda_i > 0,\ \forall\ \lambda_i = \lambda_1, \ldots, \lambda_n \in eig(A)$ ⌟

Proofs can be found in [Trefethen, 1997].

A.1.1 Eigenvalues

Definition A.6 (Eigenvalue). *An eigenvalue and an eigenvector of a square matrix* A *are a scalar* λ *and a nonzero vector* x *so that*

$$Ax = \lambda x \tag{A.1}$$

⌟

Definition A.7 (Singular Value). *A singular value and pair of singular vectors of a square or rectangular matrix* A *are a non-negative scalar* σ *and two nonzero vectors* u *and* v *so that*

$$\begin{aligned} Av &= \sigma u, \\ A^H u &= \sigma v. \end{aligned} \tag{A.2}$$

⌟

Where superscript H denotes the *Hermetian Transpose.*

Eigenvalues are very important for solving systems of ordinary differential equations. 'The values of λ can correspond to frequencies of vibration, or critical values of stability parameters, or energy levels of atoms' [Moler, 2004].

A.2 Norms

A norm is defined as a measure of length or distance. Different norms measure magnitudes in different ways, and hence some will be better suited for specific applications than others. Here we will introduce the most common ones along with where one can expect to naturally encounter them.

Definition A.8 (Norm). *[Trefethen, 1997] A norm is a function* $||.|| : \mathbb{C}^m \to \mathbb{R}$ *that assigns a real-valued length to each vector. In order to conform to a reasonable notion of length, a norm must satisfy the following three conditions. For all vectors* x *and* y *and for all scalars* $a \in \mathbb{R}$*,*

1. $||x|| \geq 0, and||x|| = 0 \implies x = 0$ *, (Nonzero condition)*
2. $||x + y|| \leq ||x|| + ||y||$*, (Triangle inequality)*
3. $||ax|| = |a| \; ||x||$*, (Scaling property).*

⌟

The 1-Norm

$$|.|_1 = \sum_{i \in n} abs(.) \tag{A.3}$$

The 1-norm is also sometimes referred to as the Manhattan norm. This is because it visually represents the measure of distance actually traveled for taxi drivers driving around blocks.

The 1-norm then gives the absolute distance between two consecutive points as seen on a grid.

The 2-Norm

$$||.||_2 = \sqrt{\sum_{i=1}^{N} |.|^2} \tag{A.4}$$

The 2-norm is also often referred to as the Euclidean norm. This norm is the most common to talk about when discussing general distance between two objects, as this norm will equal the length of a straight line drawn between two objects.

The ∞ norm

$$||x||_\infty = \max_{1 \leq i \leq m} |x_i|, \tag{A.5}$$

The p-norm

$$||x||_p = (\sum_{i=1}^{m} |x_i|^p)^{\frac{1}{p}} \quad (1 \leq p < \infty) \tag{A.6}$$

A.3 Singular Value Decomposition

After defining the singular values in section A.1, we can define the Singular Value Decomposition as follows:

Definition A.9 (Singular Value decomposition). *The Singular Value Decomposition of any $m \times n$ matrix A can be expressed as*

$$A = U\Sigma V^T \tag{A.7}$$

Where U and V are orthogonal matrices, and Σ is a diagonal matrix whose entries σ_k are called singular values. ⌟

More on the SVD can also be found in [Trefethen, 1997] or [Moler, 2004].

A.3.1 Low Rank Approximation

A low-rank approximation, also called Principal Component Analysis, approximates full rank matrices by multiple rank one matrices (recall the definition of the rank of a matrix from section A.1). Let

$$A = U\Sigma V^T \tag{A.8}$$

Be the SVD of a $m \times n$ real matrix A. Then SVD can be rewritten as:

$$A = E_1 + E_2 + ... + E_p, \tag{A.9}$$

where $p = min(m, n)$, and E_k are rank one matrices. Each of the matrices, E_k, can be expressed as:

$$E_k = \sigma_k u_k v_k^T \tag{A.10}$$

where $\sigma = \text{diag}(\Sigma)$, and u_k , v_k are the k-th columns of matrix U and V.

The norm of each component matrix is the corresponding singular value

$$||E_k|| = \sigma_k. \tag{A.11}$$

The number of rank-one matrices used in the approximation (starting from E_1) is the rank number approximation of the matrix.

Low-rank approximations are used in a number of fields such as statistics, archaeology, and image processing. Because of the variation in fields, the notation also varies. Note that this will often be described as finding the eigenvalues and eigenvectors from $A^T A$ since

$$A^T A V = V \Sigma^2. \tag{A.12}$$

This notation transforms the cross-product matrix into the correlation matrix known from statistics. The following figure showcases the low-rank approximations of an initial image:

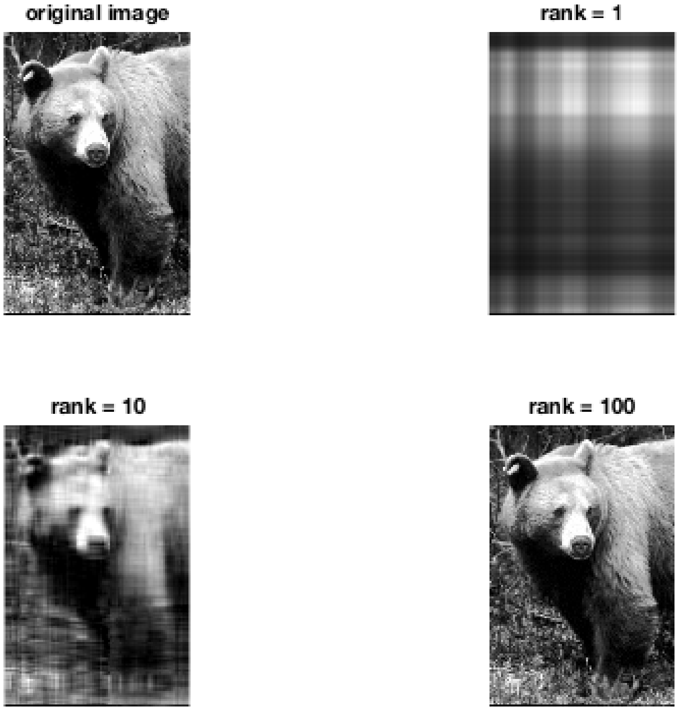


Figure A.1: Visual representation of image and its low rank approximations. The first image is the original image; *Black Bear*. Copyright photos courtesy of Robert E. Barber, Barber Nature Photography (REBarber@msn.com)

Figure A.1 shows that the first low approximation has pointed out all the major dark and light parts in the image. The low rank approximation with rank 100 is

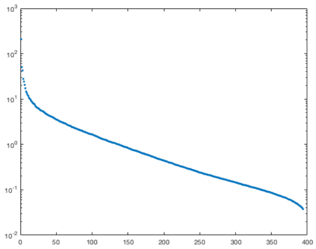


Figure A.2: Singular values of image in fig. 2.4.

almost as clear as the initial image, which has a rank of 394, as it is a 600 × 394 pixel image. This means that a majority of the characteristics can be stored with 1/4 of the singular values.

When looking at the singular value curve in fig. A.2, the curve starts out steep as the first values hold a lot of information. For other images this curve will look differently. See [Moler, 2004] for another experiment like this.

A.4 Finite Differences

The study of finite differences is to approximate the derivatives or partial derivatives in an equation. By substituting derivatives with their discrete approximations, one can obtain the numerical solutions of ODEs and PDEs. This section will describe the most common approximations and will follow the notation from [LeVeque, 2007].

First Order Approximations

$$D_+ = \frac{df(x)}{dx} = \frac{f(x+h) - f(x)}{h} \tag{A.13}$$

This is called the **Forward difference** where h (also sometimes denoted ϵ) is a small discretization step, here in space.

Note: The approximations of the derivatives yield a small error. These errors can be found by performing a Taylor expansion to get information about the quality of the approximation relative to the step size used. See [LeVeque, 2007] for more on this.

Backward difference

$$D_- = \frac{df(x)}{dx} = \frac{f(x) - f(x-h)}{h} \tag{A.14}$$

Centered Difference

$$D_2(f(x)) = \frac{df(x)}{dx} = \frac{1}{2}\frac{f(x+h) - f(x-h)}{h} \tag{A.15}$$

Second Order Approximations

$$\frac{d^2f}{dx^2} = \frac{1}{h^2}(2f(x) - f(x-h) - f(x+h)) \tag{A.16}$$

This approximation can also be derived from just applying first order approximations twice:

$$\begin{aligned} D_+(D_-f(x)) &= \frac{1}{h}(D_+(f(x)) - D_+(f(x-h))) \\ = \frac{1}{h}(\frac{1}{h}(f(x+h)) - f(x) &- \frac{1}{h}f(x) - f(x-h)) \\ = \frac{1}{h}(\frac{1}{h}(f(x+h) &+ f(x-h) - 2f(x)) \\ &= D_2(f(x)) \end{aligned} \tag{A.17}$$

A.5 Nonlinear ODEs

As all of the ODEs in this work will be nonlinear, this section will discuss how to treat these types of problems.

A nonlinear ODE will have the form

$$u'' = f(u), \tag{A.18}$$

where $f(u)$ is a nonlinear function (f.ex $e^u, u^2, sin(u)$ etc.). We can discretize this equation similarly as we would an ordinary one:

$$\frac{1}{h^2}(u_{i+1} - 2u_i + u_{i-1}) - f(u_i) = 0 \tag{A.19}$$

with $i = 1, \dots, n,\ \ h = T/(n+1)$. We can also set the boundary conditions to be $u_0 = \alpha$, $u_{n+1} = \beta$. This now becomes a system of n equations and n unknowns. But it will also be a nonlinear system of the form:

$$F(u) = 0 \tag{A.20}$$

Where $F : \mathbb{R}^n \to \mathbb{R}^n$. Instead of a direct method, we must use an iterative method such as Newtons method.

Newtons method If $u^{[k]}$ is an approximation of the solution u at timestep k, then *Newtons method* is derived via the Taylor expansion

$$F(u^{k+1}) = F(u^k) + F'(u^k)(u^{k+1} - u^k) + \frac{1}{2}F''(u^k)(u^{k+1} - u^k)^2 + \dots \tag{A.21}$$

Setting $F(u^{k+1}) = 0$ as desired (because of eq. A.20) and dropping the higher order terms, results in

$$0 = F(u^k) + F'(u^k)(u^{k+1} - u^k) \tag{A.22}$$

And we end up with the Newton update

$$u^{k+1} = u^k + \delta^k, \tag{A.23}$$

where δ^k solves the system

$$J(u^k)\delta^k = -F(u^k). \tag{A.24}$$

and J is the *Jacobian matrix*, $J(u) \equiv F'(u)$. Note that for every time-step we have to solve a tridiagonal system until it has reached the final time. This is similar to the single tridiagonal system solved in the linear case. Note that we need an initial guess to use the Newton iteration. The following can be said about the stability:

Definition A.10. *The nonlinear difference method $F(u) = 0$ is stable in some norm $||.||$ if the matrices $(\hat{J}^h)^{-1}$ are uniformly bounded in this norm as $h \to 0$, i.e., there exists constants C and h_0 such that*

$$||(\hat{J}^h)^{-1}|| \leq C \text{ for all } h < h_0. \tag{A.25}$$

⌟

Note: 'Stability of the difference method does not imply that Newton's method will converge from a poor guess' [LeVeque, 2007].

See [LeVeque, 2007] p. 38 for a more specific example.

A.5.1 Lipschitz Continuity

Lipschitz continuity plays an important role in determining whether an ODE has a solution or not.

Definition A.11 (Lipschitz Continuous). *We say that the function f(u,t) is Lipschitz continuous over some domain*

$$\mathcal{D} = \{(u,t) : |u - \eta| \leq a \; t_0 \leq t \leq t_1\} \tag{A.26}$$

if there exists some constant $L \geq 0$ so that

$$|f(u,t) - f(u,t)| \leq L|u - u| \tag{A.27}$$

for $\forall$ $(u,t), (u,t) \in \mathcal{D}$. ⌟

The Lipschitz constant, L, retrieves information on how smooth the nonlinearity in the equation is. It also expresses how much a solution diverges or converges at a certain point, which tells how much the function changes. Note that it does not differ between rapid convergence and rapid divergence.

If $f(u,t)$ is differentiable with respect to u in $\mathcal{D}$ and its derivative, $f_u = \frac{\partial f}{\partial u}$, is bounded, the Lipschitz constant can be expressed as:

$$L = \max_{(u,t)\in\mathcal{D}} |f_u(u,t)| \tag{A.28}$$

Existence and Uniqueness

Theorem A.12. *If f is Lipschitz continuous over a region, $\mathcal{D}$, then there is a unique solution to the BVP atleast up to a time $T = \min(t_1, t_0 + a/S)$, where*

$$S = \max_{(u,t)\in\mathcal{D}} |f(u,t)|. \tag{A.29}$$

This can be generalized to only requiring that a function is *Locally Lipschitz.*

Locally Lipschitz Although it is easier to show that an equation has an existing and unique solution when it is globally Lipschitz continuous, we also sometimes need the weaker form of the Lipschitz continuity. This will enable us to include more functions of which we can be certain that their solution exists before we start looking for it numerically, as will be described in the following section.

Theorem A.13 (Local Lipschitz Continuity). *Let X be a normed space and $f : X \to \mathbf{R}$. If f is continuously differentiable in a neighborhood V of a point $x_0 \in X$, then f is locally Lipschitz at x_0.*

In other words, given x_0, use the continuity of f' to find a neighborhood U of x_0 st. $||f'(x) - f'(x_0)|| \leq 1 \; \forall \; x \in U$. $\implies$ $||f'(x)|| \leq ||f'(x_0)|| + 1$ in U $\implies$ f is Lipschitz on U.

A.5.2 Solvers

Although nonlinear equations might have an existing unique solution, they generally do not have an analytical solution. This means that we need to solve nonlinear equations numerically.

Note: Any well-posed ODE can be solved numerically, but linear ODEs also often have an analytical solution as well. If both are known, the analytical solution can be used to verify the accuracy of the numerical solution.

For any nonlinear equation, we need to solve it numerically and march in time. Methods that can be used for this are Explicit Euler update, Taylor expansion, and both Runge Kutta and Newton that build on Taylor expansion, see [LeVeque, 2007].

A.5.3 Iterative Solvers for Linear Systems

Large matrices can be computationally expensive to solve directly, and so iterative solvers are very beneficial. For large matrices arising from discretizations of differential equations, we generally end up with large sparse systems easier solved with iterative solvers. For a full overview of iterative methods, see literature such as [LeVeque, 2007], [Trefethen, 1997]. Here we will only include Conjugate Gradients, as this is the algorithm used to solve the resulting system of equations in this thesis.

Conjugate Gradient Algorithm Conjugate gradients are used to solve $Au = b$ when A is a symmetric, positive definite matrix. The algorithm works as follows:

Algorithm 1: Conjugate Gradient Algorithm.

Input : Initial guess u_0, possibly zero vector.

Output : Solution approximation u_k after k iterations.

```
// Initial residual.
$r_0 = f - Au_0$;
$p_0 = r_0$ ;
for $k = 1, 2, \ldots$ do
    $w_{k-1} = Ap_{k-1}$;
    $\alpha_{k-1} = (r_{k-1}^T r_{k-1})/(p_{k-1}^T w_{k-1})$ ;
    $u_k = u_{k-1} + \alpha_{k-1} w_{k-1}$ ;
    if $||r_k|| < tol$ then stop ;
    $\beta_{k-1} = (r_k^T r_k)/(r_{k-1}^T r_{k-1})$;
    $p_k = r_k + \beta_{k-1} r_{k-1}$ ;
end
```