Master's thesis

2019

Esten Nicolai Wøien

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Esten Nicolai Wøien

A Semi-Discretized Method for Optimal Reparametrization of Curves

June 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

A Semi-Discretized Method for Optimal Reparametrization of Curves

Esten Nicolai Wøien

Applied Physics and Mathematics
Submission date: June 2019
Supervisor: Markus Grasmair

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

In this thesis, we develop a new method for solving the optimal reparametrization problem within the square root velocity framework. The method is based on a dynamic programming approach, but with a more accurate update equation than previous methods. While previous methods are fully discretized, the new method is only semi-discretized. This is utilized to give both a better convergence rate and a lower computational complexity compared to similar methods.

To construct the method, we introduce new auxiliary variables, and establish differential equations characterizing the optimal reparametrizers. The resulting method is linear in the reparametrizers and quadratic in the distance estimate. In certain situations, these convergence rates can be improved to quadratic and super-quadratic, respectively, by the use of extrapolation. This is supported by numerical experiments.

Sammendrag

I denne oppgaven utvikler vi en ny metode for optimal omparametrisering av kurver ved bruk av rothastighetstransformasjonen (*the square root velocity transform*). Metoden bruker dynamisk programmering, men med en bedre håndtering av gruntilfellene enn tidligere metoder. Mens tidligere metoder er fullstendig diskretisert, er den nye metoden kun delvis diskretisert. Dette utnyttes til å oppnå både en bedre konvergensrate og lavere asymptotisk kjøretid sammenlignet med tilsvarende metoder.

Under utviklingen av metoden introduser vi nye hjelpevariabler og nye differensialligninger som karakteriserer optimale løsninger. Metoden er lineær i omparametriseringsfunksjonene, og kvadratisk i avstandsestimatet. I enkelte tilfeller kan henholdsvis kvadratisk og super-kvadratisk konvergens oppnås ved hjelp av ekstrapolasjon. Dette underbygges av numeriske eksperimenter.

Preface

This thesis concludes my five years of studies of applied physics and mathematics at NTNU, with specialization in industrial mathematics.

I would like to thank my supervisor, Markus Grasmair, for great guidance through the writing process. We have had many interesting discussions, and you have always been open for a question or ten.

Esten Nicolai Wøien
Trondheim
June 12, 2019

Table of Contents

List of Figures

1 Introduction

Shape analysis is the field concerned with the analysis of geometric shapes. The field includes topics such as object recognition and classification, and it is accordingly important within applications such as computer vision and medical imaging.

Within shape analysis, it is important to have well-posed definitions of geometric shapes. Although the concept of a geometric shape can be defined in multiple ways, it is common to consider shapes represented by curves or surfaces, and we will in this thesis consider curves in particular. Here, it is important that the concept of a shape is invariant with respect to to the parametrization of the curves. To ensure this, we define a *shape* as an equivalence class in the space of parametric curves, where two curves are considered equivalent if one curve can be continuously reparametrized to the other. Representing shapes by curves is useful since this gives access to tools from differential geometry. For example, it is common to define similarities and dissimilarities between shapes through Riemannian metrics. Again, it is important that such metrics are invariant to the parametrization of the curves.

A popular choice of a Riemannian metric is the so-called *elastic metric* [1, 2, 3, 4]. This metric considers elastic deformations of the curves by measuring the bending and stretching required to deform one curve into another. In this thesis, we will consider a specific elastic metric, defined through the *Square Root Velocity Transform* as introduced in [4]. This metric has a key property: If two curves are optimally parametrized, we have explicit formulae for the geodesics between the shapes of the curves. Here, we consider two curves to be optimally parametrized if the geodesics between their shapes can be computed using the curves themselves, i.e., without reparametrization.

The concept of optimal parametrization can be formulated as a variational problem. We consider two parametric curves (representing two shapes), and define optimality as the reparametrizations of the curves which minimize some metric on the space of parametric curves. This is commonly denoted as the curve registration problem since a solution provides a registration of points on each curve.

Each curve is typically parametrized using some abstract “time” parameter. Accordingly, the optimal reparametrization problem can be seen as a problem of finding a matching between the time domains of the curves in question. Further, we assume that the time domains are monotone, meaning that we want to find a monotone matching between the domains. For such problems, dynamic programming methods are typically available. This holds for the optimal reparametriation problem where both gradient based method and dynamic programming based methods [5] have been used.

The main contribution of this thesis is a new dynamic programming approach which is more accurate than previous dynamic programming based methods. The new method agrees with the previous approach in that the time domains of the curves are discretized. Where the two methods differ is in the base cases of the dynamic programming. The previous method searches for the optimal velocity of the reparametrizations among a discrete set of velocities. In other words, the previous method is a fully discretized method. In our new approach, we construct the base cases using a continuous optimization problem, meaning that the method as a whole is only semi-discretized. The new construction of the base cases is used to both obtain a better convergence rate and a lower computational cost.

The thesis consists of three parts. In Chapter 2, we start by defining the optimal reparametrization problem as a variational problem. We review useful reformulations to simplify the problem, and reformulations necessary to ensure existence of solutions. Then, we review previous results, and provide a new result characterizing the optimal reparametrizations. The chapter is concluded by defining auxiliary variables used to construct the dynamic programming method. Although these variables have been used to derive the previous dynamic programming based methods, there has been little to no emphasis on the properties of the variables. Under certain regularity assumptions of the auxiliary variables, we provide differential equations governing both the auxiliary variables and the optimal reparametrizations. Additionally, we demonstrate how the auxiliary variable can be defined through a hyperbolic partial differential equation. Lastly, conditions for the appearance of shocks and differential equations governing the evolution of shock paths are established.

In Chapter 3 we show how the auxiliary variables can be used to construct a numerical solver. The method is motivated by the special case of the general reparametrization problem where we assume both curves to be linear. This is then used as a base case to construct a dynamic programming method where we assume the curves to be piecewise linear. Through the differential equations derived in the previous chapter, we demonstrate how the dynamic programming method can be interpreted as a finite difference scheme, and we show how to retrieve the optimally reparametrized curves. We also demonstrate how the approximated solution can be used to compute geodesics. Assuming that the auxiliary variable is absolutely continuous, the resulting method has a linear convergence rate for the optimally reparametrized curves and a quadratic convergence for the similarity / dissimilarity estimates. We show how extrapolation can be used to improve these convergence rates to quadratic and super-quadratic, respectively, if no shock solutions are present.

In Chapter 4, we demonstrate the convergence rates of the solvers empirically. We consider both simple problems where analytic solutions are available, and more interesting problems, where exact solutions must be estimated. Where shock solutions do not appear, the theoretical convergence rates are verified. In this case, however, extrapolation does not improve the asymptotic convergence rates.

2 | A Shape Space Metric

In this chapter, we review current theory on the existence and charactrization of optimal reparametrizers within the square root velocity framework. We then expand the theory on the characterization of the optimal solutions, and define auxiliary variables related to a measure of similarity between two curves. Lastly, we will see how these auxiliary variables can be used to construct differential equations governing the optimal reparametrizers, and how the auxiliary variables are governed by a hyperbolic PDE.

2.1 The Shape Space for Parametric Curves

A parametric curve is a mapping $c : I \to \mathbb{R}^d$ which belongs to a certain regularity class. The assumed regularity class varies from application to application — we will for now assume that the curves are C^2-continuous. Further, we will only consider open curves, hence the unit interval $I = [0, 1]$ is a natural choice of domain. For closed curves, it is common to choose the unit circle $I = S^1$.

On the space of parametric curves, we are interested in defining a metric which encapsulates the geometric properties of the curves. It would be natural to define this metric on the images of the curves since the images really do include all geometric aspects. This, however, is a hard task and the smoothness properties of the curves are much easier to exploit when the curves are of parametric form, rather than defined by their images. Additionally (see Figure 2.1), the image does always contain all the information of the curve. Therefore, we are interested in an equivalence class other than the class of curves with the same image.

Consider the set of curves that are reparametrizations of one another. Here, a reparametrization is defined as a right composition $c \mapsto c \circ \varphi$ for some $\varphi \in \mathrm{Diff}(I)$, where $\mathrm{Diff}(I)$ denotes the set of orientation preserving C^2-diffeomorphisms from I to I. The orientation preserving property can be ensured by the constraint $\dot{\varphi} > 0$. Additionally, we define shapes modulo translations: If two curves only differ by a translation, they should belong to the same shape. This can be ensured by only considering curves starting at the origin. Lastly, we will only consider curves with non-zero velocity everywhere. If we would allow zero-velocity curves, this would allow sharp corners which contravenes the assumed smoothness properties. Therefore, we will define our space of parametric curves as

$$\mathcal{C} := \operatorname{Imm}(I, \mathbb{R}^d) := \{c \in C^2(I, \mathbb{R}^d) \mid c(0) = 0, |\dot{c}| > 0\}.$$

The notation Imm is a natural choice since the set consists of immersions. We then define the *shape* of a curve $c \in \mathcal{C}$ as the equivalence class

$$[c] := \{c \circ \varphi \mid \varphi \in \operatorname{Diff}(I)\}.$$

This is actually an equivalence class since $\operatorname{Diff}(I)$ is a group which ensures that $[c \circ \varphi] = [c]$. Consequently, the entire equivalence class can be identified from any single representative c. We will denote the set of shapes $[c]$ as the shape space $\mathcal{S}$.

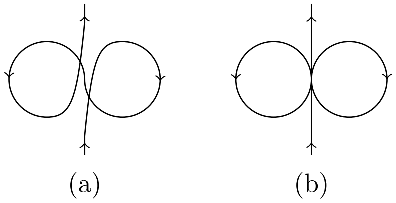


Figure 2.1: Curves with given starting positions where we can (a) and cannot (b) uniquely determine the orientation from their images.

The shape and the image of a curve are two separate concepts. Since the image of a parametric curve does not depend on its parametrization, all curves that belong to the same shape have the same image, as desired. However, the reverse it not true in general. This is easy to see for open curves where we simply reverse the orientation of the curve. Then, the reversed curve and the curve itself have the same image, but there is no orientation-preserving reparametrization from the reversed curve to the curve itself. Additionally, we cannot determine whether the curve in Figure 2.1b goes through the right or left loop first. Therefore we cannot determine the orientation of this curve, as opposed to the orientation of the curve in Figure 2.1a, which is unique due to the assumed smoothness properties.

Since we want to define a metric on the space of parametric curves which encapsulates the geometric features of the curves, it is natural to define it on the shape space S. Now, consider any metric $d_{\mathcal{C}}(c_1, c_2)$ defined on the space of parametric curves. We can then define a metric on $\mathcal{S}$ by minimizing $d_{\mathcal{C}}$ over all representatives of the two shapes. Such a metric can be defined as

$$d_{\mathcal{S}}([c_1], [c_2]) = \inf_{b_1 \in [c_1], b_2 \in [c_2]} d_{\mathcal{C}}(b_1, b_2),$$

or equivalently

$$d_{\mathcal{S}}([c_1], [c_2]) := \inf_{\varphi_1, \varphi_2 \in \operatorname{Diff}(I)} d_{\mathcal{C}}(c_1 \circ \varphi_1, c_2 \circ \varphi_2). \tag{2.1}$$

With this definition we only require a well defined metric $d_{\mathcal{C}}$. However, this metric must be constructed with care. A tempting choice could be the L^2-metric. However, defining $d_{\mathcal{C}}(c_1, c_2) = \|c_1 - c_1\|_{L^2}$ does not induce a well defined shape space metric. It has been shown that for any pair of curves, the L^2-distance between the curves vanishes when we minimize over all reparametrizations of the curves [6, 7, 8]. In other words, using the L^2-metric as the metric on the space of parametric curves will result in a shape space "metric" which satisfies $d_{\mathcal{S}}([c_1], [c_2]) = 0$, regardless of c_1 and c_2. Therefore, we need to construct other metrics for the parametric curves.

2.2 The Square Root Velocity Transform

Consider the *square root velocity transform* (SRVT) as introduced in [4]. In our context, this can be seen as a mapping $R : \operatorname{Imm}(I, \mathbb{R}^d) \to C^1(I, \mathbb{R}^d \setminus \{0\})$ given by

$$R(c)(t) = \frac{\dot{c}(t)}{\sqrt{|\dot{c}(t)|}},$$

with associated inverse

$$R^{-1}(q)(t) = \int_0^t q(s)|q(s)|dt.$$

Throughout this thesis we will use the notation $q = R(c)$. The mapping R acts on the entire space $\operatorname{Imm}(I, \mathbb{R}^d)$ and is hence an injection. Additionally, since we only consider curves starting at the origin, we do not need to consider the initial value of the integral. This ensures that both the left and right inverse of R is defined everywhere, meaning that R is a bijection between $\operatorname{Imm}(I, \mathbb{R}^d)$ and $C^1(I, \mathbb{R}^d \setminus \{0\})$. Hence, all relevant information of a curve c is captured by its SRVT q. The SRVT is used to construct a metric on the space of parametric curves which takes the form

$$d_{\mathcal{C}}(c_1, c_2) = \|q_1 - q_2\|_{L^2}.$$

This metric on the space of parametric curves can further be used to define a shape space metric through (2.1). To do so, we need to know how the SRVT behaves under reparametrization. We have that

$$R(c \circ \varphi) = \frac{(\dot{c} \circ \varphi)\dot{\varphi}}{\sqrt{|(\dot{c} \circ \varphi)\dot{\varphi}|}} = \frac{(\dot{c} \circ \varphi)}{\sqrt{|(\dot{c} \circ \varphi)|}}\sqrt{\dot{\varphi}} = (R(c) \circ \varphi)\sqrt{\dot{\varphi}}.$$

This implies that the shape space metric induced by the SRVT can be defined as

$$d(c_1, c_2) = \inf_{\varphi_1, \varphi_2 \in \operatorname{Diff}(I)} \left\| (q_1 \circ \varphi_1)\sqrt{\dot{\varphi}_1} - (q_2 \circ \varphi_2)\sqrt{\dot{\varphi}_2} \right\|_{L^2}. \tag{2.2}$$

The motivation behind the SRVT induced metric comes a specific Riemannian metric. A Riemannian metric on $\operatorname{Imm}(I, \mathbb{R}^d)$ is given by an inner product on each tangent space $T_c \operatorname{Imm}(I, \mathbb{R}^d)$ for $c \in \operatorname{Imm}(I, \mathbb{R}^d)$. Consider a curve c and an element of h of the tangent space $T_c \operatorname{Imm}(I, \mathbb{R}^d)$. Since $T_c \operatorname{Imm}(I, \mathbb{R}^d)$ can be identified with $C^2(I, \mathbb{R}^d)$, h can be seen as a C^2-continuous curve from I to $\mathbb{R}^d$. We reparametrize h according to the arc length of c. Then, we consider derivatives of h of the form $D_s h$, where we define $D_s := |\dot{c}|^{-1}\partial_t$. Further, we decompose $D_s h$ into its tangential and normal components relative to the parametrization of c. Specifically, denote

$$\begin{aligned}(D_s h)^{\perp} &= \langle D_s h, D_s c\rangle D_s c,\\ (D_s h)^{\top} &= D_s h - (D_s h)^{\perp}.\end{aligned}$$

Note that $D_s c = \dot{c}/|\dot{c}|$ is in fact the unit tangent to the curve c. With this notation, we define the *elastic metric* as

$$G_c(h,k) := \int_I a^2 \big\langle (D_s h)^{\perp}, (D_s k)^{\perp} \big\rangle + b^2 \big\langle (D_s h)^{\top}, (D_s k)^{\top} \big\rangle ds$$

for some positive constants a^2 and b^2. This was first introduced in [3]. Note that this is an arc length integral with $ds = |\dot{c}|dt$, which ensures that the metric is invariant under reprametrizations of c. There is a quite nice interpretation of this metric. The first part of the integrand, weighted by a^2, considers the tangential components of h and k, and can therefore be seen as a measure of stretching. Similarly, the second part, weighted by b^2, can be seen as a measure of bending, as it is only concerned with the normal components of h and k. Additionally, since these parts are independently weighted, the weights can be chosen to favor either bending or stretching.

The weights are commonly chosen to be $a^2 = 1$ and $b^2 = 1/4$. It has been shown that the pullback of the L^2-norm via the SRVT is the elastic metric using these weights [4]. This is an especially useful result since geodesics in the L^2-topology are easy to compute. In fact, the geodesic between any $q_1, q_2 \in L^2(I, \mathbb{R}^d)$ is simply given by $\tau \mapsto (1-\tau)q_1 + \tau q_2$. This, however, is not a well defined geodesic on the space of immersions. If there exist $t, \tau \in I$ such that $\tau q_1(t) + (1-\tau)q_2(t) = 0$, this construction of the geodesics allows zero-velocity curves, which contradicts the curves being immersions. In Section 2.3, we will additionally see that through reparametrization, we must allow the reparametrized curves $c_1 \circ \varphi_1$ and $c_2 \circ \varphi_2$ to have zero velocity to ensure that optimal reparametrizations exist. In other words, geodesics between certain curves will be outside the space of parametric curves (with nonzero velocity everywhere). To cope with this, geodesic completion has been discussed in the square root velocity framework in [9], and for more general Sobolev metrics in [10]. Additionally, we refer to [9, 4, 3] for further readings on the elastic metric and choice of SRVT.

There are certain properties of the SRVT which are of interest. Firstly, consider the arc length of the curve c, which we denote as $L(c)$. We have that

$$L(c) := \int_I |\dot{c}| dt = \int_I \left| \frac{\dot{c}}{\sqrt{|\dot{c}|}} \right|^2 dt = \|q\|_{L^2}^2.$$

In other words, the squared L^2-norm of a square root velocity transformed curve equals the length of the original curve. This implies that the space of unit length curves is mapped by the SRVT to the unit sphere of L^2-functions. Additionally, since the length of a curve is not dependent of its parametrization, we have that $\|(q \circ \varphi)\sqrt{\dot{\varphi}}\|_{L^2}^2 = \|q\|_{L^2}^2$ for any $\varphi \in \mathrm{Diff}(I)$. This motivates two reformulations.

2.2.1 Redundancy of the Problem

The invariance property of the SRVT leads to redundancy of the variational problem. Since the functional is invariant under joined reparametrizations, we will never have uniqueness of solutions (if solutions exist). If $(\varphi_1^*, \varphi_2^*)$ is a solution, then for any $\psi \in \mathrm{Diff}(I)$, the joined reparametrizatzers $(\varphi_1^* \circ \psi, \varphi_2^* \circ \psi)$ be a solution as well. It is therefore common to apply certain constraints to the search space to remove this redundancy. One idea is to only reparametrize one of the curves and define the variational problem as

$$d(c_1, c_2) = \inf_{\varphi \in \mathrm{Diff}(I)} \left\| q_1 - (q_2 \circ \varphi)\sqrt{\dot{\varphi}} \right\|_{L^2}.$$

This is a common idea, and it is a valid reformulation since the search space $\mathrm{Diff}(I)$ is a group. However, we will in the next section see that we need to allow zero-derivatives of the paths, i.e. either allow $\dot{\varphi}_1 = 0$ or $\dot{\varphi}_2 = 0$ to ensure existence of solutions. This breaks the group property of the search space, which implies that we cannot consider reformulations such as the one above.

Still, additional constraints might come in handy to cope with the redundancy, and we will eventually consider the constraint

$$\dot{\varphi}_1 + \dot{\varphi}_2 = 2.$$

Note that this is equivalent to $\varphi_1(t) + \varphi_2(t) = 2t$. In other words, if $(\varphi_1(t), \varphi_2(t)) = (x_0, y_0)$, then t is uniquely defined as $t = \frac{1}{2}(x_0 + y_0)$. This property holds for all (φ_1, φ_2) which pass through the point (x_0, y_0). In the next chapter, we will optimize over all paths which pass through the point (x_0, y_0) where this constraint will be useful. However, the constraint is optional and we will for now not assume this nor any other additional constraints to hold. We will only assume $\dot{\varphi}_1 + \dot{\varphi}_2 = 2$ wherever explicitly stated.

2.2.2 Maximization of the Inner Product

The connection between the length of the curve and the L^2-norm of the SRVT, $L(c) = \|q\|_{L^2}^2$ motivates another reformulation of the variational problem (2.2). By expansion of the square, we have that

$$\begin{aligned} \|q_1 - q_2\|_{L^2}^2 &= \|q_1\|_{L^2}^2 + \|q_2\|_{L^2}^2 - 2\langle q_1, q_2 \rangle_{L^2} \\ &= L(c_1) + L(c_2) - 2\langle q_1, q_2 \rangle_{L^2}. \end{aligned}$$

Since $L(c_1)$ and $L(c_2)$ are invariant to the parametrization of c_1 and c_2, the above equality can be used to reformulate the variational problem as a maximization of the inner product, rather than a minimization of the norm. Specifically, define

$$s(c_1, c_2) := \sup_{\varphi_1, \varphi_2 \in \mathrm{Diff}(\ I)} F(\varphi_1, \varphi_2), \tag{2.3}$$

where

$$F(\varphi_1, \varphi_2) := \left\langle (q_1 \circ \varphi_1)\sqrt{\dot{\varphi}_1}, (q_2 \circ \varphi_2)\sqrt{\dot{\varphi}_2} \right\rangle_{L^2}.$$

Here, and throughout the rest of the thesis, the notation $F(\varphi_1, \varphi_2)$ assumes fixed q_1 and q_2.

We denote $s(c_1, c_2)$ as the *similarity* between the curves c_1 and c_2 since a larger value of s is associated with a smaller distance. There is a strictly monotone (in fact linear) mapping between the functionals of the variational problems (2.2) and (2.3), which means that any local solution of one of the problems will be a local solution of the other. This holds even though the linear mapping is strictly decreasing, as the two optimization problems differ in that one is a maximization problem, while the other is minimization problem. In other words, the problems can be said to be equivalent, and choice of either one of them is only a matter of preference. However, we experience that the equations that arise when maximizing the similarity are more compact and intuitive than when minimizing the distance. Therefore, we will in this thesis consider methods for solving (2.3). If the distance is specifically of interest, it can be retrieved through the equality

$$d(c_1, c_2)^2 = L(c_1) + L(c_2) - 2s(c_1, c_2). \tag{2.4}$$

We can also find bounds for the distance and the similarity. First of all, both the distance and the similarity are nonnegative. The distance is trivially nonnegative since it is defined as the infimum of a norm. To see why the similarity is nonnegative, consider the following pair of functions:

$$\psi_1(t) = \begin{cases} 0, & t \in \left[0, \frac{1}{2}\right], \\ 2t - 1, & t \in \left(\frac{1}{2}, 1\right], \end{cases} \qquad \psi_2(t) = \begin{cases} 2t, & t \in \left[0, \frac{1}{2}\right], \\ 1, & t \in \left(\frac{1}{2}, 1\right]. \end{cases}$$

For all t, these functions satisfy $\dot{\psi}_1\dot{\psi}_2 = 0$, which ensures that $F(\psi_1, \psi_2) = 0$. Although these functions are not diffeomorphic, they are absolutely continuous which means that they can be arbitrarily well approximated by diffeomorphisms. Further, since our functional F is continuous, we can construct diffeomorphisms (φ_1, φ_2) for which $F(\varphi_1, \varphi_2)$ is arbitrarily close to zero. This implies that the supremum and hence also $s(c_1, c_2)$ is nonnegative. Combining the nonnegativity of the distance and the similarity with the equality (2.4), we obtain the following bounds:

$$\begin{aligned} 0 \leq \ & d(c_1, c_2)^2 \leq L(c_1) + L(c_2), \\ 0 \leq \ & 2s(c_1, c_2) \ \leq L(c_1) + L(c_2). \end{aligned}$$

2.3 Reformulation to Ensure Existence of Solutions

From the current definition of the problem, we do not allow zero derivatives of the reparametrizers, i.e. $\dot{\varphi}_1 = 0$ or $\dot{\varphi}_2 = 0$. Unfortunately, this implies that the problem will in many cases not attain a solution. The easiest way to see this is to consider the extreme case where $\langle q_1(x), q_2(y)\rangle < 0$ for all x, y. Then, for any pair diffeomorphisms φ_1, φ_2, we have that $F(\varphi_1, \varphi_2) < 0$. However, as we have seen, we can approximate a pair of functions which satisfy $\dot{\psi}_1\dot{\psi}_2 = 0$ arbitrarily well, meaning that we can get arbitrarily close to $F(\varphi_1, \varphi_2) = 0$. In other words, we have that

$$\sup_{\varphi_1,\varphi_2\in \mathrm{Diff}(I)} F(\varphi_1, \varphi_2) = 0.$$

It is clear that there is no pair of diffeomorpshism for which the supremum is attained, and it is therefore of interest to reconstruct the search space to possibly ensure the existence of a solution.

Rather than considering diffeomorphic reparametrizations, we will assume the reparametrizatizers to be absolutely continuous, and we will additionally allow their derivatives to be zero. In other words, we will consider reparametrizers of the form

$$\begin{aligned}\Phi([t_0, t_1], [x_0, x_1]) = \big\{\varphi \in \mathrm{AC}([t_0, t_1], [x_0, x_1]) \,|\varphi(t_0) &= x_0,\\ \varphi(t_1) &= x_1,\\ \dot{\varphi} &\geq 0 \text{ a.e.}\big\},\end{aligned}$$

and redefine the problem as

$$s(c_1, c_2) = \sup_{\varphi_1,\varphi_2\in\Phi(I)} F(\varphi_1, \varphi_2). \tag{2.5}$$

We will use the abbreviation $\Phi(I) = \Phi(I, I)$. This problem has been thoroughly studied in [9] where a proof is provided that the problem has a solution for all C^1-continuous curves c_1, c_2 with nonzero velocity almost everywhere. Although the optimization problems have different search spaces, the original search space $\mathrm{Diff}(I)$ is dense in $\Phi(I)$. Further, since F is continuous in φ_1 and φ_2, the problems will therefore have the same supremum.

Remark 1. Note that this reformulation is in fact compatible with the constraint $\dot{\varphi}_1 + \dot{\varphi}_2 = 2$. To see this, observe that any pair of functions $\varphi_1, \varphi_2 \in \Phi(I)$ can be seen as a curve $(\varphi_1, \varphi_2) \in \mathrm{AC}(I, I \times I)$. Since this curve is absolutely continuous, we are free to choose a constant speed parametrization of the curve. By measuring the speed in the L^1-norm, we get that $\varphi_1 + \varphi_2$ must be constant, as desired.

2.3.1 Concatenation of Reparametrization Paths

Another useful property of absolutely continuous functions is that the concatenation of absolutely continuous functions is also absolutely continuous. For some $\varphi \in \Phi([t_0, t_1], [x_0, x_1])$ and $\vartheta \in \Phi([t_1, t_2], [x_1, x_2])$, we define their concatenation by

$$\varphi \oplus \vartheta : t \mapsto \begin{cases} \varphi(t), & t \in [t_0, t_1),\\ \vartheta(t), & t \in [t_1, t_2].\end{cases}$$

Note that we require the endpoint of the first curve to be equal the start point of the second curve, both in argument (t_1) and value (x_1). Now, since piecewise absolutely continuous functions are absolutely continuous (given that they are continuous), we have that $\varphi \oplus \vartheta \in \Phi([t_0, t_2], [x_0, x_2])$.

We also have that F is additive under concatenation. To see this, we start by generalizing the functional to be domain dependent by defining

$$F_{[t_0,t_1]}(\varphi_1, \varphi_2) := \int_{t_0}^{t_1} \langle q_1 \circ \varphi_1, q_2 \circ \varphi_2 \rangle \sqrt{\dot{\varphi}_1 \dot{\varphi}_2} dt.$$

Observe that the full functional can then be defined as $F = F_I$. Further, assume that

$$\begin{aligned} \varphi_1 &\in \Phi([t_0, t_1], [x_0, x_1]), & \vartheta_1 &\in \Phi([t_1, t_2], [x_1, x_2]), \\ \varphi_2 &\in \Phi([t_0, t_1], [y_0, y_1]), & \vartheta_2 &\in \Phi([t_1, t_2], [y_1, y_2]), \end{aligned}$$

Then, F is additive under concatenation in the sense that

$$F_{[t_0,t_2]}(\varphi_1 \oplus \vartheta_1, \varphi_2 \oplus \vartheta_2) = F_{[t_0,t_1]}(\varphi_1, \varphi_2) + F_{[t_1,t_2]}(\vartheta_1, \vartheta_2).$$

This property is very useful as we want to construct the optimal reparametrizers iteratively. Note that this result could be derived with diffeomorphisms as well. To do so, however, we require additional constraints to the functions to ensure that the concatenated functions are still diffeomorphic. These additional constraints are avoided by absolutely continuous functions, emphasizing why absolutely continuous functions might be more suitable.

2.4 Characterisations of Optimal Paths

Although we do not in general have explicit formulae for the solutions to the optimization problem (2.5), we can say quite a bit about the general behaviour of the solutions. In this section, we will prove that an optimal solution path satisfies $\dot{\varphi}_1(t) = 0$ or $\dot{\varphi}_2(t) = 0$ if and only if $\langle q_1(t), q_2(t) \rangle \leq 0$. But before we get there, we need to consider a few auxiliary results.

Consider the decomposition of the unit interval given by

$$\begin{aligned} A(\varphi_1, \varphi_2) &= \{t \in I : \langle q_1(\varphi_1(t)), q_2(\varphi_2(t)) \rangle \geq 0\}, \\ B(\varphi_1, \varphi_2) &= \{t \in I : \langle q_1(\varphi_1(t)), q_2(\varphi_2(t)) \rangle < 0\}. \end{aligned}$$

It is clear that $I = A(\varphi_1, \varphi_2) \cup B(\varphi_1, \varphi_2)$ for all $\varphi_1, \varphi_2 \in \Phi(I)$. Additionally, since $q_1 \circ \varphi_1$ and $q_2 \circ \varphi_2$ are continuous, then A must be closed and that B must be open. In the following lemma, we will show that if $B(\varphi_1, \varphi_2)$ is nonempty, we can construct another path such that the integral over $B(\varphi_1, \varphi_2)$ can be neglected. The idea is as follows: If $B(\varphi_1, \varphi_2)$ is nonempty, there exists some open interval (t_0, t_1) such that the inner product between the reparametrized SRVTs is negative. In Figure 2.2 this is drawn as the path between $(\varphi_1(t_0), \varphi_2(t_0)) = (x_0, y_0)$ and $(\varphi_1(t_1), \varphi_2(t_1)) = (x_1, y_1)$. Since the interval is open, we have that

$$\int_{t_0}^{t_1} \langle q_1 \circ \varphi_1, q_2 \circ \varphi_2 \rangle \sqrt{\dot{\varphi}_1 \dot{\varphi}_2} dt \leq 0.$$

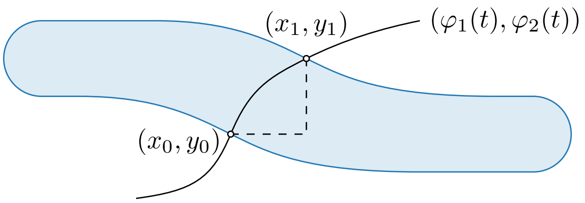


Figure 2.2: A path passing through a region where $\langle q_1(x), q_2(y)\rangle < 0$, shaded blue.

By replacing (φ_1, φ_2) by a piecewise horizontal or vertical path (drawn as a dashed line in the figure), we are enforcing $\dot{\varphi}_1\dot{\varphi}_2 = 0$. This in turn ensures that the above integral becomes exactly zero, which means that we can neglect the interval (t_0, t_1).

Lemma 2.4.1. [9, Lemma 16] *For all $\varphi_1, \varphi_2 \in \Phi(I)$, there exists $\psi_1, \psi_2 \in \Phi(I)$ such that*

$$\int_I \langle q_1 \circ \psi_1, q_2 \circ \psi_2\rangle \sqrt{\dot{\psi}_1\dot{\psi}_2}dt = \int_{A(\varphi_1,\varphi_2)} \langle q_1 \circ \varphi_1, q_2 \circ \varphi_2\rangle \sqrt{\dot{\varphi}_1\dot{\varphi}_2}dt.$$

Proof. The proof is taken directly from [9, Lemma 16]. If $B(\varphi_1, \varphi_2)$ is empty, the lemma is trivially proven by setting $\psi_1 = \varphi_1$ and $\psi_2 = \varphi_2$. Therefore, assume that $B(\varphi_1, \varphi_2)$ is nonempty. Since B is open, it can be constructed as the union of a countable set of open intervals, which we denote as $B = \bigcup_k I_k$ where $I_k = (t_k^-, t_k^+)$. Additionally, we split the intervals in half by defining $I_k^- = (t_k, \frac{1}{2}(t_k^- + t_k^+)]$ and $I_k^+ = (\frac{1}{2}(t_k^- + t_k^+), t_k^+)$. We construct (ψ_1, ψ_2) in the following way:

$$\psi_1(t) = \begin{cases} \varphi_1(2t - t_k^-), & t \in I_k^-, \\ \varphi_1(t_k^+), & t \in I_k^-, \\ \varphi_1(t) & \text{otherwise}, \end{cases}$$

$$\psi_2(t) = \begin{cases} \varphi_2(t_k^-), & t \in I_k^-, \\ \varphi_2(2t - t_k^+), & t \in I_k^-, \\ \varphi_2(t) & \text{otherwise}. \end{cases}$$

This construction ensures that $\psi_1, \psi_2 \in \Phi(I)$. Additionally, for all $t \in I_k^-$ we have that $\dot{\psi}_1 = 0$ and for all $t \in I_k^+$ we have that $\dot{\psi}_2 = 0$. In other words, for all $t \in B$, we have that $\dot{\psi}_1\dot{\psi}_2 = 0$. This gives

$$\int_{B(\varphi_1,\varphi_2)} \langle q_1 \circ \psi_1, q_2 \circ \psi_2\rangle \sqrt{\dot{\psi}_1\dot{\psi}_2}dt = 0$$

Since $I = A(\varphi_1,\varphi_2) \cup B(\varphi_1,\varphi_2)$, we obtain

$$\begin{aligned}\int_I \langle q_1 \circ \psi_1, q_2 \circ \psi_2\rangle \sqrt{\dot{\psi}_1\dot{\psi}_2}dt &= \int_{A(\varphi_1,\varphi_2)} \langle q_1 \circ \psi_1, q_2 \circ \psi_2\rangle \sqrt{\dot{\psi}_1\dot{\psi}_2}dt \\ &= \int_{A(\varphi_1,\varphi_2)} \langle q_1 \circ \varphi_1, q_2 \circ \varphi_2\rangle \sqrt{\dot{\varphi}_1\dot{\varphi}_2}dt\end{aligned}$$

concluding the proof. □

This lemma tells us that if a path passes through a region for which we have that $\langle q_1(x), q_2(y)\rangle < 0$, we can always alter the path such that this region does not contribute negatively towards the objective function. A direct consequence is given in the following result.

Corollary 2.4.2. *We have that*

$$\sup_{\varphi_1,\varphi_2\in\Phi(I)} F(\varphi_1,\varphi_2) = \sup_{\varphi_1,\varphi_2\in\Phi(I)} F^+(\varphi_1,\varphi_2) \tag{2.6}$$

where F^+ is given by either of the following equivalent definitions

$$\begin{aligned}F^+(\varphi_1,\varphi_2) &:= \int_{A(\varphi_1,\varphi_2)} \langle q_1 \circ \varphi_1, q_2 \circ \varphi_2\rangle \sqrt{\dot{\varphi}_1\dot{\varphi}_2}dt \\ &:= \int_I \max\left\{\langle q_1 \circ \varphi_1, q_2 \circ \varphi_2\rangle, 0\right\} \sqrt{\dot{\varphi}_1\dot{\varphi}_2}dt.\end{aligned}$$

Proof. For all $\varphi_1,\varphi_2 \in \Phi(I)$, we have that $F^+(\varphi_1,\varphi_2) \geq F(\varphi_1,\varphi_2)$. However, as seen in Lemma 2.4.1, we can for all $\varphi_1,\varphi_2 \in \Phi(I)$ construct $\psi_1,\psi_2 \in \Phi(I)$ such that $F(\psi_1,\psi_2) = F^+(\varphi_1,\varphi_2)$. Hence

$$\sup_{\psi_1,\psi_2\in\Phi(I)} F(\psi_1,\psi_2) = \sup_{\varphi_1,\varphi_2\in\Phi(I)} F^+(\varphi_1,\varphi_2),$$

concluding the proof. □

Note that Corollary 2.4.2 applies both when optimizing over diffeomorphisms and optimizing over absolutely continuous functions since $\mathrm{Diff}(I)$ is dense in $\Phi(I)$ and that both F and F^+ are continuous. The corollary tells us that the similarity, which can now be expressed as the supremum over F^+, does not “see” the regions where the curves are negatively correlated. However, if (φ_1,φ_2) solves the right hand side of (2.6), it does not necessarily solve the left hand side. This is because the positive functional F^+ only ensures that we do not need to consider the regions where the curves are negatively correlated. From this formulation, we cannot say anything about the behavior of the solution in these regions.

A similar argument to the proof of Lemma 2.4.1 can also be used to prove positivity of the similarity, assuming that the curves are somewhere positively correlated. This is formally described in the following lemma:

Lemma 2.4.3. *We have a positive similarity $s(c_1,c_2) > 0$ if and only if there exist a point (x,y) such that $\langle q_1(x), q_2(y)\rangle > 0$.*

Proof. Step 1: proving that a positive similarity implies a at some point positive inner product. Assume to the contrary that $\langle q_1(x), q_2(y)\rangle \leq 0$ for all x, y. Via Corollary 2.4.2, we have that $s(c_1, c_2) = 0$. This contradicts $s(c_1, c_2) > 0$, concluding this part of the proof.

Step 2: proving that a at some point positive inner product implies a positive similarity. Since q_1 and q_2 are continuous, there exist some open rectangle $(x_0, x_1) \times (y_0, y_1)$ for which $\langle q_1(x), q_2(y)\rangle > 0$. Now, construct φ_1, φ_2 in the following way:

$$\dot\varphi_1 = \begin{cases} \frac{x_0}{t_1}, & t \in [0, t_1), \\ 0, & t \in [t_1, t_2], \\ \frac{x_1 - x_0}{t_3 - t_2}, & t \in (t_2, t_3), \\ \frac{(1-x_1)}{t_4 - t_3}, & t \in [t_3, t_4), \\ 0, & t \in [t_4, 1], \end{cases} \qquad \dot\varphi_2 = \begin{cases} 0, & t \in [0, t_1), \\ \frac{y_0}{t_2 - t_1}, & t \in [t_1, t_2], \\ \frac{y_1 - y_0}{t_3 - t_2}, & t \in (t_2, t_3), \\ 0, & t \in [t_3, t_4), \\ \frac{(1-y_1)}{1 - t_4}, & t \in [t_4, 1], \end{cases}$$

for some $0 < t_1 < t_2 < t_3 < t_4 < 1$. Using initial conditions $\varphi_1(0) = \varphi_2(0) = 0$, we have that $\varphi_1, \varphi_2 \in \Phi(I)$, meaning that the path is feasible. Further, we have that $\langle q_1(\varphi_1(t)), q_2(\varphi_2(t))\rangle \sqrt{\dot\varphi_1(t)\dot\varphi_2(t)} > 0$ for all $t_2 < t < t_3$, and $\sqrt{\dot\varphi_1(t)\dot\varphi_2(t)} = 0$ otherwise. This gives

$$F(\varphi_1, \varphi_2) = \int_{t_2}^{t_3} \langle q_1(\varphi_1(t)), q_2(\varphi_2(t))\rangle \sqrt{\dot\varphi_1(t)\dot\varphi_2(t)} dt > 0.$$

Therefore, the supremum must also be positive, concluding the proof. □

Although Lemma 2.4.3 might be trivial, it is a nice result. If there is some positively correlated parts of the curves, the similarity between the curves will be positive, and vice versa.

The last concept we need before introducing the main result in this section, is defining the variation of F. The variation of F measures the change in F for small changes in the reparametrization path (φ_1, φ_2) in a feasible direction (γ_1, γ_2). We say that the direction is feasible if for sufficiently small h, $(\varphi_1 + h\gamma_1, \varphi_2 + h\gamma_2)$ is still inside the search space. In other words, for (γ_1, γ_2) to be a feasible direction, we need that $\varphi_1 + h\gamma_1, \varphi_2 + h\gamma_2 \in \Phi(I)$. This is ensured by the following requirements:

(i) $\gamma_1, \gamma_2 \in \mathrm{AC}(I, \mathbb{R})$.

(ii) $\gamma_1(0) = \gamma_2(0) = \gamma_1(1) = \gamma_2(1) = 0$.

(iii) If $\dot\varphi_1(t) = 0$, then $\dot\gamma_1(t) \geq 0$. If $\dot\varphi_2(t) = 0$, then $\dot\gamma_2(t) \geq 0$.

(iv) $\dot\gamma_1 + \dot\gamma_2 = 0$.

The last condition is optional, and is only used to ensure $\dot\varphi_1 + h\dot\gamma_1 + \dot\varphi_2 + h\dot\gamma_2 = 2$ which we will ignore. We can now state and prove the main result in this section.

Theorem 2.4.4. *If (φ_1, φ_2) solves (2.5), then*

(a) for a.e. $t \in I$ s.t. $\dot\varphi_1(t), \dot\varphi_2(t) > 0$, we have that $\langle q_1(\varphi_1(t)), q_2(\varphi_2(t))\rangle \geq 0$.

(b) for a.e. $t \in I$ s.t. $\langle q_1(\varphi_1(t)), q_2(\varphi_2(t))\rangle > 0$, we have that $\dot\varphi_1(t), \dot\varphi_2(t) > 0$.

Proof of (a). Assume to the contrary that there is a set with nonzero measure for which $\dot{\varphi}_1(t), \dot{\varphi}_2(t) > 0$ and $\langle q_1(\varphi_1(t)), q_2(\varphi_2(t)) \rangle < 0$. Then, we have that

$$\int_I \langle q_1 \circ \varphi_1, q_2 \circ \varphi_2 \rangle \sqrt{\dot{\varphi}_1 \dot{\varphi}_2} dt < \int_{A(\varphi_1, \varphi_2)} \langle q_1 \circ \varphi_1, q_2 \circ \varphi_2 \rangle \sqrt{\dot{\varphi}_1 \dot{\varphi}_2} dt$$

In other words, $F(\varphi_1, \varphi_2) < F^+(\varphi_1, \varphi_2)$. However, via Lemma 2.4.1, we can construct a feasible path ψ_1, ψ_2 such that $F(\psi_1, \psi_2) = F^+(\varphi_1, \varphi_2)$. Hence (φ_1, φ_2) cannot be optimal. □

Proof of (b). Assume to the contrary that there is a set with nonzero measure for which $\langle q_1(\varphi_1(t)), q_2(\varphi_2(t)) \rangle > 0$.

Case 1: $\dot{\varphi}_1(t)\dot{\varphi}_2(t) = 0$ for almost every $t \in I$. This implies that $F(\varphi_1, \varphi_2) = 0$. However, by assumption, there is at least one point (x, y) such that $\langle q_1(x), q_2(y) \rangle > 0$. Via Lemma 2.4.3, we have that

$$\sup_{\psi_1, \psi_2 \in \Phi} F(\psi_1, \psi_2) > 0 = F(\varphi_1, \varphi_2).$$

Hence, (φ_1, φ_2) is not optimal.

Case 2: there is a set with nonzero measure for which $\dot{\varphi}_1, \dot{\varphi}_2 > 0$. We want to prove this by contradiction. By assuming that Theorem 2.4.4b does not hold, we will show that there exist a feasible direction (γ_1, γ_2) for which the objective function increases, meaning that (φ_1, φ_2) cannot be optimal. An informal description of the construction of the construction of this direction, together with an informal summary of the following proof, can be found after the completion of the proof.

Define

$$\begin{aligned} C_1 &= \{t \in I \mid \dot{\varphi}_1(t) = 0, \langle \varphi_1(t), \varphi_2(t) \rangle > 0\}, \\ C_2 &= \{t \in I \mid \dot{\varphi}_2(t) = 0, \langle \varphi_1(t), \varphi_2(t) \rangle > 0\}, \\ D_1 &= \{t \in I \mid \dot{\varphi}_1(t) = 0\}, \\ D_2 &= \{t \in I \mid \dot{\varphi}_2(t) = 0\}. \end{aligned}$$

Observe that $C_1 \subseteq D_1$ and $C_2 \subseteq D_2$. As established in Section 2.3, we can safely assume that $\dot{\varphi}_1(t) + \dot{\varphi}_2(t) > 0$ for all t. Hence, we can safely assume that C_1 and C_2 are essentially disjoint. This also holds for D_1 and D_2.

With this notation, we want to prove that $C := C_1 \cup C_2$ has measure zero. Therefore, assume to the contrary that C has nonzero measure. We construct the direction (γ_1, γ_2) in the following way:

$$\dot{\gamma}_1(t) = \begin{cases} 1, & t \in C_1 \\ 0, & t \in D_1 \setminus C_1, \\ 0, & t \in D_2, \\ -k_1 \dot{\varphi}_1(t), & \text{otherwise,} \end{cases} \qquad \dot{\gamma}_2(t) = \begin{cases} 1, & t \in C_2 \\ 0, & t \in D_2 \setminus C_2, \\ 0, & t \in D_1, \\ -k_2 \dot{\varphi}_2(t), & \text{otherwise,} \end{cases}$$

for some $k_1, k_2 > 0$. We choose k_1 such that the constraint $\gamma_1(1) = \int \dot{\gamma}_1 = 0$ is satisfied. Inserting γ_1, we obtain

$$\begin{aligned}\int \dot{\gamma}_1 dt &= \int_{C_1} 1dt + \int_{D_1 \setminus C_1} 0dt - k_1 \int_{I \setminus D} \dot{\varphi}_1 dt \\ &= |C_1| - k_1 \int_{I \setminus D} \dot{\varphi}_1 dt = 0.\end{aligned}$$

Here, we define $D := D_1 \cup D_2$. By assumption, $I \setminus D$ has nonzero measure, which implies that the last integral is strictly positive, ensuring that k_1 is well-defined. The same argument holds for k_2. Additionally, we have that $\dot{\varphi}_1 + h\dot{\gamma}_1 = \dot{\varphi}_1(1 - hk_1)$ and $\dot{\varphi}_2 + h\dot{\gamma}_2 = \dot{\varphi}_2(1 - hk_2)$ which are nonnegative for sufficiently small h. Lastly, since φ_1 and φ_2 are absolutely continuous, so are γ_1 and γ_2, which means that (γ_1, γ_2) is a feasible direction.

We decompose the functional into

$$F(\varphi_1 + h\gamma_1, \varphi_2 + h\gamma_2) = \mathcal{I}_1 + \mathcal{I}_2 + \mathcal{I}_3$$

where $\mathcal{I}_1$, $\mathcal{I}_2$ and $\mathcal{I}_3$ are the integration over D_1, D_2 and $I \setminus D$, respectively. In other words, we define

$$\begin{aligned}\mathcal{I}_1 &:= \int_{D_1} \langle q_1(\varphi_1 + h\gamma_1), q_2(\varphi_2 + h\gamma_2)\rangle \sqrt{(\dot{\varphi}_1 + h\dot{\gamma}_1)(\dot{\varphi}_2 + h\dot{\gamma}_2)}dt, \\ \mathcal{I}_2 &:= \int_{D_2} \langle q_1(\varphi_1 + h\gamma_1), q_2(\varphi_2 + h\gamma_2)\rangle \sqrt{(\dot{\varphi}_1 + h\dot{\gamma}_1)(\dot{\varphi}_2 + h\dot{\gamma}_2)}dt, \\ \mathcal{I}_3 &:= \int_{I \setminus D} \langle q_1(\varphi_1 + h\gamma_1), q_2(\varphi_2 + h\gamma_2)\rangle \sqrt{(\dot{\varphi}_1 + h\dot{\gamma}_1)(\dot{\varphi}_2 + h\dot{\gamma}_2)}dt.\end{aligned}$$

Consider the first integral $\mathcal{I}_1$. Recall for all $t \in D_1$ we have that $\dot{\varphi}_1 = 0$ and by construction $\dot{\gamma}_1 = 0$. Hence, we have that

$$\dot{\varphi}_1 + h\dot{\gamma}_1 = \begin{cases} 1, & t \in C_1, \\ 0, & t \in D_1 \setminus C_1, \end{cases} \qquad \dot{\varphi}_2 + h\dot{\gamma}_2 = \dot{\varphi}_2.$$

In other words, the integration over $D_1 \setminus C_1$ vanishes and we are left with

$$\mathcal{I}_1 = \sqrt{h} \int_{C_1} \langle q_1(\varphi_1 + h\gamma_1), q_2(\varphi_2 + h\gamma_2)\rangle \sqrt{\dot{\varphi}_2}dt.$$

Since q_1 and q_2 are continuously differentiable, the integrand converges linearly with h. Further, since $\dot{\varphi}_2$ is integrable, the whole integral converges linearly with h. We obtain a similar result for $\mathcal{I}_2$, replacing C_1 with C_2 and $\dot{\varphi}_2$ with $\dot{\varphi}_1$. Since $C = C_1 \cup C_2$ is non-negligible, the sum of the integrals over C_1 and C_2 are strictly positive, and there exists a positive constant K such that

$$\mathcal{I}_1 + \mathcal{I}_2 > K\sqrt{h}$$

for sufficiently small h.

For the third integral, $\mathcal{I}_3$, we have that $\dot{\varphi}_1 + h\dot{\gamma}_1 = \dot{\varphi}_1(1-k_1h)$ and $\dot{\varphi}_2 + h\dot{\gamma}_2 = \dot{\varphi}_2(1-k_2h)$. Inserting this gives

$$\mathcal{I}_3 = \sqrt{(1-k_1h)(1-k_2h)} \int_{I\setminus D} \langle q_1(\varphi_1 + h\gamma_1), q_2(\varphi_2 + h\gamma_2)\rangle \sqrt{\dot{\varphi}_1\dot{\varphi}_2} dt.$$

We have that $\sqrt{(1-k_1h)(1-k_2h)}$ is continuously differentiable at $h=0$, with Taylor expansion $1 - \frac{1}{2}(k_1+k_2)h + o(h)$. Further, we have the following Taylor expansion

$$\langle q_1(\varphi_1 + h\gamma_1), q_2(\varphi_2 + h\gamma_2)\rangle = \langle q_1(\varphi_1), q_2(\varphi_2)\rangle + \mathcal{O}(h).$$

Since $\varphi_1, \varphi_2, \gamma_1, \gamma_2$ are absolutely continuous and that q_1 and q_2 are continuously differentiable, this $\mathcal{O}(h)$ has a uniform constant with respect to t. Since the integral is always finite, we obtain

$$\begin{aligned}\mathcal{I}_3 &= (1 - \tfrac{1}{2}(k_1+k_2)h + o(h)) \int_{I\setminus D} \big(\langle q_1(\varphi_1), q_2(\varphi_2)\rangle + \mathcal{O}(h)\big) \sqrt{\dot{\varphi}_1\dot{\varphi}_2} dt \\ &= \int_{I\setminus D} \langle q_1(\varphi_1), q_2(\varphi_2)\rangle \sqrt{\dot{\varphi}_1\dot{\varphi}_2} dt + \mathcal{O}(h) \\ &= F(\varphi_1, \varphi_2) + \mathcal{O}(h).\end{aligned}$$

We have here used that both $\dot{\varphi}_1$ and $\dot{\varphi}_2$ and hence also $\sqrt{\dot{\varphi}_1\dot{\varphi}_2}$ are integrable. Combining all three integrals, this gives

$$\begin{aligned}F(\varphi_1 + h\gamma_1, \varphi_2 + h\gamma_1) &= \mathcal{I}_1 + \mathcal{I}_2 + \mathcal{I}_3 \\ &= F(\varphi_1, \varphi_2) + K\sqrt{h} + \mathcal{O}(h) \\ &> F(\varphi_1, \varphi_2)\end{aligned}$$

for sufficiently small h. In other words, we can find another feasible path with a higher objective value. This contradicts that (φ_1, φ_2) is optimal, concluding the proof. □

The feasible direction in case 2 of the proof Theorem 2.4.4b is visualized in Figure 2.3. The continuous line represents the proposed solution path (φ_1, φ_2), and the dashed line represents a small step $(h\gamma_1, h\gamma_2)$ added to the proposed path. We want to particularly consider the regions where either $\dot{\varphi}_1 = 0$ or $\dot{\varphi}_2 = 0$. This is where the path is either horizontal or vertical, which is accented in the figure. Where the path is either horizontal or vertical, and the inner product is greater than zero (case (a) and (b)), we nudge the path to become slightly less horizontal or vertical. Where the path is either horizontal or vertical, but the inner product is negative (case (c) and (d)), we do not alter the slope of the path. Although the *position* of the path might be slightly altered, it is only important that the slope is not changed. For the remaining part of the path, we only do slight alterations to ensure that the path remains feasible, while ensuring that no parts of the curve becomes horizontal or vertical which was not already horizontal or vertical.

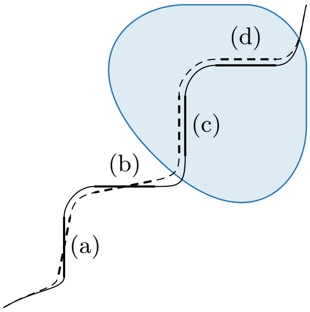


Figure 2.3: Example of the feasible variation constructed in case 2 of proof of Theorem 2.4.4b. The solid line represents (φ_1, φ_2), the dashed line represents $(\varphi_1 + h\gamma_1, \varphi_2 + h\gamma_2)$ and the shaded region represents where $\langle q_1(x), q_2(y)\rangle < 0$. The parts of (φ_1, φ_2) which are either horizontal or vertical (a, b, c, d) are accented.

Since the integrand in the functional is proportional to $\sqrt{\dot{\varphi}_1\dot{\varphi}_2}$, the slight alterations in (a) and (b) will contribute with a positive change in the functional value which is proportional to $\sqrt{h}$. All remaining parts will contribute with a change in the functional value proportional to h, which can be neglected for small h. Therefore, we have constructed a feasible direction for which the functional increases, meaning that the proposed path is not optimal.

Theorem 2.4.4 is to our knowledge a new result, and it provides insight into how solutions to the problem behave. In regions where the inner product is negative, we must have a vertical or horizontal slope to avoid a negative contribution to the objective value. In regions where the inner product is positive, we must have positive derivatives of the reparametrizers. Having an either vertical or horizontal slope could in this case be seen as “waste” of a positive correlation between the curves.

2.5 Auxiliary Similarity Metrics

In the following, we will present three auxiliary metrics which gives additional insight into the problem. These metrics are constructed as generalizations of the similarity metric (2.5). Since Diff(I) is dense in $\Phi(I)$, the metrics could be introduced using diffeomorphisms as well. However, as we have seen, we choose to define the metrics using $\Phi(I)$ to ensure the existence of solutions.

2.5.1 Partial Similarity

An important idea in this thesis, is that of defining a similarity on only parts of each curve. Specifically, we want to define a similarity metric, assuming smaller domains $[x_0, x_1]$ and $[y_0, y_1]$ for curve c_1 and c_2, respectively. To be consistent with previous definitions, we require $0 \le x_0 \le x_1 \le 1$ and $0 \le y_0 \le y_1 \le 1$. We can apply this idea through the changing the reparametrizations in the definition of the similarity. Since the support of the reparametrizations acts as the domain of each curve, we can simply change the support of φ_1 and φ_2 to $[x_0, x_1]$ and $[y_0, y_1]$. In other words, we assume $\varphi_1 \in \Phi([0,1],[x_0,x_1])$ and $\varphi_2 \in \Phi([0,1],[y_0,y_1])$ and define the *partial similarity* as

$$P(x_0, y_0, x_1, y_1) := \sup_{\substack{\varphi_1 \in \Phi([0,1],[x_0,x_1]),\\ \varphi_2 \in \Phi([0,1],[y_0,y_1])}} F(\varphi_1, \varphi_2),$$

First of all, note that $P(0,0,1,1) = s(c_1, c_2)$. This is as we expect, since in this case, we do not restrict the domain of either c_1 or c_2. Since P is defined on arbitrary sub-rectangles of the unit square, it can be used to incrementally construct the total similarity s. This will be done using a dynamic programming approach in the next chapter.

There are certain interesting properties of the partial similarity P. First of all, P nonnegative for the same reason that $s(c_1, c_2)$ is nonnegative. Furthermore, if $x_0 = x_1$ or $y_0 = y_1$, we have that $P(x_0, y_0, x_1, y_1) = 0$. This is because $\varphi \in \Phi([0,1],[x_0,x_0])$ satisfies $\dot{\varphi} = 0$ everywhere, which ensures that the functional evaluates to zero.

Additionally, for $x_0 \leq x_1 \leq x_2$ and $y_0 \leq y_1 \leq y_2$, we have that

$$P(x_0, y_0, x_2, y_2) \geq P(x_0, y_0, x_1, y_1) + P(x_1, y_1, x_2, y_2). \tag{2.7}$$

This property is due to the fact that we can concatenate the solutions to the optimization problems in $P(x_0, y_0, x_1, y_1)$ and $P(x_1, y_1, x_2, y_2)$. Further, the second inequality (2.7) reduces to an equality if and only if there is an optimal path which passes through the point (x_1, y_1).

Remark 2. The property (2.7) is actually related to the triangle inequality. If we would have defined P by minimizing the norm, rather than maximizing the inner product, the inequality sign in (2.7) would be reversed. Hence, the property could be seen as a triangle inequality for monotone triples of points (x_0, y_0), (x_1, y_1), (x_2, y_2).

2.5.2 Cumulative Similarity

Consider the *cumulative similarity*, which we define by the abbreviation $S(x,y) = P(0,0,x,y)$. Written out, this reads

$$S(x,y) := \sup_{\substack{\varphi_1 \in \Phi([0,1],[0,x]),\\ \varphi_2 \in \Phi([0,1],[0,y])}} F(\varphi_1, \varphi_2), \tag{2.8}$$

For now, we consider (x,y) as a point in $\mathbb{R}^d$. The denotion of cumulative similarity comes from the fact as S measures the cumulative integration of the inner product over optimal paths starting at $(0,0)$. Although S is only a special case of the more general variable P, the cumularive similarity S has certain properties which will become very useful. First of all, S is monotone increasing, which is a direct consequence of (2.7). Further properties of S are discussed in the next chapter. Additionally, there is one reformulation which should be mentioned. If $\varphi \in \Phi([0,1],[0,1])$, then we have that $t \mapsto x\varphi(t) \in \Phi([0,1],[0,x])$. This can be used to reformulate S in the following way:

$$S(x,y) := \sup_{\varphi_1,\varphi_2 \in \Phi(I)} F(x\varphi_1, y\varphi_2). \tag{2.9}$$

2.5.3 Restricted Similarity

We want to consider a similarity measure when we restrict the solution to pass through a point (x,y). Such points are often called landmarks and landmark-guided shape analysis has been studied both in the context of curves [5] and surfaces [11]. However, we want to define the restricted similarity metric for all (x,y), regardless of whether the point is a landmark or not. We define the *restricted similarity* as

$$S|_{x,y} := \sup_{\varphi_1,\varphi_2 \in \Phi(I)} F(\varphi_1,\varphi_2) \quad \text{s.t.} \quad \exists t:\ \varphi_1(t) = x, \varphi_2(t) = y.$$

Although we will not use this variable in the development of our method, this is a great tool for evaluating the fitness of the solution. If an optimal solution passes through the point $(\varphi_1(t),\varphi_2(t)) = (x,y)$ for some $t \in [0,1]$, then $S|_{x,y} = s(c_1,c_2)$. In other words, $S|_{x,y}$ stays constant, equal to the total similarity along the path $(x,y) = (\varphi_1(t),\varphi_2(t))$. For that reason, the restricted similarity can be used to determine if we have found a solution. If a path (φ_1,φ_2) is locally optimal, then $S|_{x,y}$ should be non-increasing in all directions orthogonal to the path for all $(x,y) = (\varphi_1(t),\varphi_2(t))$. Lastly, as a direct consequence of (2.7), we have that

$$S|_{x,y} = P(0,0,x,y) + P(x,y,1,1).$$

This comes from the fact that the inequality (2.7) has equality if and only if the optimal path passes through the point (x,y) (denoted as (x_1,y_1) in equation (2.7)). In the definition of $S|_{x,y}$ we force all paths to pass through this point, hence we can apply (2.7). The property hints as to why a dynamic programming approach might be suitable. If we know that the optimal paths passes through the point (x,y), the optimization problem can be reduced to finding the optimal reparametrizations on the regions $[0,x] \times [0,y]$ and $[x,1] \times [y,1]$ independently.

2.6 Differential Properties of S

Solutions to variational problems are often governed by differential equations. This is also true for the optimal reparametrization problem, and the auxiliary variable S is very helpful in this regard. We start with one of the fundamental properties of S:

Proposition 2.6.1. *S is continuous.*

Proof. This follows directly from the continuity of F and that x, y and φ_1, φ_2 are coupled linearly. Formally, we have the following argument. Writing out the functional F, the definition of S as given in (2.9) can be reformulated as

$$S(x,y) = \sup_{\varphi_1,\varphi_2\in\Phi(I)} \langle R(xc_1\circ\varphi_1), R(yc_2\circ\varphi_1)\rangle_{L^2}$$

For simplicity, we will consider absolutely continuous curves which belongs to $\mathrm{AC}_0 = \{c \in \mathrm{AC}(I,\mathbb{R}^d), c(0) = 0\} \supset \mathcal{C}$. This allows us to use certain results from [9]. We have that

(i) $(c, x) \mapsto xc$ is a continuous map from $\mathrm{AC}_0 \times \mathbb{R}$ to AC_0,

(ii) $(c, \varphi) \mapsto c \circ \varphi$ is a continuous map from $\mathrm{AC}_0 \times \Phi(I)$ to AC_0 [9, Proposition 7],

(iii) $c \mapsto R(c)$ is a continuous map from AC_0 to L^2 [9, Lemma 4].

This ensures that the mapping

$$(c_1, c_2, \varphi_1, \varphi_2, x, y) \mapsto \langle R(xc_1\circ\varphi_1), R(yc_2\circ\varphi_1)\rangle_{L^2}$$

is continuous. Now, consider any sequence $(x_n, y_n)_n \to (x, y)$. Since the above mapping is continuous, we have that

$$\begin{aligned}\lim_{n\to\infty} S(x_n,y_n) &= \lim_{n\to\infty}\sup_{\varphi_1,\varphi_2}\langle R(x_nc_1\circ\varphi_1), R(y_nc_2\circ\varphi_1)\rangle_{L^2}\\ &= \sup_{\varphi_1,\varphi_2}\lim_{n\to\infty}\langle R(x_nc_1\circ\varphi_1), R(y_nc_2\circ\varphi_1)\rangle_{L^2}\\ &= \sup_{\varphi_1,\varphi_2}\langle R(xc_1\circ\varphi_1), R(yc_2\circ\varphi_1)\rangle_{L^2}\\ &= S(x,y),\end{aligned}$$

concluding the proof. □

We want to slightly alter the notation for the optimal reparametrizers for a better synergy with S. Specifically, we will from now on denote optimal paths as (x, y). This is motivated by the definition of S where φ_1 is analogous to x and φ_2 is analogous to y. Using this notation, we have a particularly useful property of optimal paths:

$$S(x(t), y(t)) = \int_0^t \langle q_1(x(s)), q_2(y(s))\rangle \sqrt{\dot{x}(s)\dot{y}(s)}ds. \tag{2.10}$$

Using Corollary 2.4.2, we can reformulate this to

$$S(x(t), y(t)) = \int_0^t Q(x(s), y(s))\sqrt{\dot{x}(s)\dot{y}(s)}ds, \tag{2.11}$$

where we define $Q(x,y) := \max\{\langle q_1(x), q_2(y)\rangle, 0\}$. Through this equation we have an explicit expression for S along an optimal path. Note the connection to the method of characteristics, where one solves a PDE along certain paths called characteristics. We will come back to this later.

Using (2.11), we can compute the temporal derivative of S along optimal paths. The fundamental theorem of calculus gives us

$$\tfrac{d}{dt}S(x(t),y(t)) = Q(x(t),y(t))\sqrt{\dot{x}(t)\dot{y}(t)},$$

or $\dot{S} = Q\sqrt{\dot{x}\dot{y}}$, for short. Further, assuming that x, y are differentiable at t and S is differentiable at x, y, we have that $\dot{S} = S_x\dot{x} + S_y\dot{y}$. In other words, we have two expressions for $\dot{S}$ which together becomes

$$S_x\dot{x} + S_y\dot{y} = Q\sqrt{\dot{x}\dot{y}}. \tag{2.12}$$

This is a central equation in the proof of the following results.

Proposition 2.6.2. *Assume that S is differentiable. Then, we have that (x,y) solves (2.8) if and only if*

$$\begin{aligned} S_x\dot{x} - S_y\dot{y} &= 0, \qquad \textit{if } S_x + S_y > 0, \\ \dot{x}\dot{y} &= 0, \qquad \textit{if } S_x + S_y = 0. \end{aligned} \tag{2.13}$$

almost everywhere.

Proof. Note that if $\dot{x} = 0$, $\dot{y} > 0$, the property (2.12) implies that $S_y = 0$, which means that the differential equation (2.13) is trivially satisfied. This also holds for $\dot{x} > 0$, $\dot{y} = 0$. We can therefore assume $\dot{x}, \dot{y} > 0$.

In general, the derivatives of a function varies a lot faster than the function value itself. In other words, $\dot{x}$ and $\dot{y}$ varies a lot faster than x and y. This means that the optimization problem (2.5) can locally be seen as an optimization problem over $\dot{x}$ and $\dot{y}$. The objective function will then be the temporal derivative of the objective function as given in (2.10). Still, we must have that (2.12) is satisfied. In other words, using the abbreviations $u = \dot{x}(t)$ and $v = \dot{y}(t)$, we get that u and v must solve

$$\sup_{u,v>0} Q\sqrt{uv} \quad \text{s.t.} \quad S_x u + S_y v = Q\sqrt{uv},$$

where $Q = Q(x(t),y(t))$, $S_x = S_x(x(t),y(t))$ and $S_y = S_y(x(t),y(t))$. The first order optimality system for this optimization problem is given by

$$\begin{aligned} S_x + \tfrac{1}{2}(1-\lambda)Q\sqrt{v/u} &= 0, \\ S_y + \tfrac{1}{2}(1-\lambda)Q\sqrt{u/v} &= 0, \\ S_x u + S_y v &= Q\sqrt{uv}, \\ u, v &> 0. \end{aligned}$$

where λ is the Lagrange multiplier for the constraint $S_x u + S_y v = Q\sqrt{uv}$. We do not need a Lagrange multiplier for the constraint $u, v > 0$ since this is already assumed to hold. Multiplying the first two equations by u and v, respectively, gives

$$\begin{aligned} S_x u + (1-\lambda)\tfrac{1}{2}Q\sqrt{uv} &= 0, \\ S_y v + (1-\lambda)\tfrac{1}{2}Q\sqrt{uv} &= 0. \end{aligned}$$

By subtracting the two equations, we obtain $S_x u - S_y v = 0$, as desired. □

This proposition gives a foundation for computing the optimal reparmetrizations, given that S is known. Later on, we will see that if S is piecewise bilinear, then the solution of (2.13) is a pair of piecewise linear functions. It should be noted that (2.13) portrays the invariance under reparametrization property for the problem. The equation is invariant under simultaneous scaling of $\dot{x}$ and $\dot{y}$, and hence also under simultaneous reparametrization.

We can also use Proposition 2.6.2 to formulate a partial differential equation for S.

Proposition 2.6.3. *If S is differentiable, then S solves the nonlinear partial differential equation*

$$S_x S_y = \frac{1}{4}Q^2 \tag{2.14}$$

on the unit square, with boundary conditions $S(x,0) = S(0,y) = 0$ and the monotonicity constraints $S_x, S_y \geq 0$.

Proof. Note that the boundary conditions and the monotonicity has already been established. We start by assuming either $\dot{x} = 0$ or $\dot{y} = 0$. Via theorem 2.4.4 we have that $Q = 0$, and via proposition 2.6.2 we have that either $S_x = 0$ or $S_y = 0$. In other words, (2.14) is satisfied. Now assume $\dot{x}, \dot{y} > 0$.

The squares of the equations (2.12) and (2.13) gives the system of equations given by

$$\begin{aligned} S_x^2(\dot{x})^2 + 2S_xS_y\dot{x}\dot{y} + S_y^2(\dot{y})^2 &= Q^2\dot{x}\dot{y}, \\ S_x^2(\dot{x})^2 - 2S_xS_y\dot{x}\dot{y} + S_y^2(\dot{y})^2 &= 0. \end{aligned}$$

Subtracting the two equations and dividing by $\dot{x}\dot{y}$ on both sides yields $4S_xS_y = Q^2$, concluding the proof. □

Remark 3. Propositions 2.6.2 and 2.6.3 are great examples as to why we obtain nicer results by maximizing the inner product rather than minimizing the norm of the SRVTs. If we define D as the cumulative distance in the same way as we defined S as the cumulative similarity, the equations equivalent to (2.13) and (2.14) would read

$$\begin{aligned} (D_x - |\dot{c}_1(x)|)\dot{x} - (D_y - |\dot{c}_2(y)|)\dot{y} &= 0, \\ (D_x - |\dot{c}_1(x)|)(D_y - |\dot{c}_2(y)|) &= \frac{1}{4}Q^2. \end{aligned}$$

Although the properties of the differential equations remains the same, the equations derived from the similarity are much more compact, and arguably much easier to comprehend.

The value of S is defined through optimal reparametrization paths. Since the reparametrizations are monotone, they can be considered as characteristics in the context of hyperbolic partial differential equations. In other words, the nonlinear PDE (2.14) is hyperbolic. Further, we can write the PDE of the form $G(x, y, S, S_x, S_y) = 0$, where

$$G(x, y, S, S_x, S_y) = S_x S_y - \frac{1}{4} Q(x, y)^2.$$

For PDEs of this form, the method of characteristics is given by

$$\begin{aligned} \dot{x} &= \lambda S_y, \\ \dot{y} &= \lambda S_x, \\ \dot{S} &= 2\lambda S_x S_y, \\ \dot{S}_x &= \tfrac{1}{2}\lambda Q Q_x, \\ \dot{S}_y &= \tfrac{1}{2}\lambda Q Q_y, \end{aligned}$$

where λ is some scaling factor. We can also obtain the first two equations directly from (2.13). The third equation is easily derived from the first two by the following steps:

$$\dot{S} = S_x \dot{x} + S_y \dot{y} = \lambda S_x S_y + \lambda S_y S_x = 2\lambda S_x S_y.$$

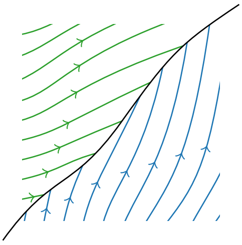


Figure 2.4: Behaviour of optimal paths around a shock path. Optimal paths are drawn with arrows, and the shock path is drawn black.

A common phenomenon for solutions of PDEs are shock. These occur when the characteristics collide, meaning that the system of ODEs is overdetermined. In our context, a characteristic is actually a reparemtrization path. Accordingly, we expect shock solutions to appear whenever we have two linearly independent directions which are optimal and that S decreases in both of these directions. Since the reparametrization paths are monotone, we expect the shock paths to be monotone as well. Additionally, from numerical experiments, there seems to only be a finite number of shock paths regardless of the curves in question. One example of how the optimal paths behave around a shock solution is visualized in Figure 2.4.

To help us analyse the behavior of the shocks, consider the following notation for the left and right derivatives of S:

$$\begin{aligned} S_{x^+}(x,y) &= \lim_{h\to 0} \frac{1}{h}\big[S(x+h,y) - S(x,y)\big], \\ S_{x^-}(x,y) &= \lim_{h\to 0} \frac{1}{h}\big[S(x,y) - S(x-h,y)\big], \\ S_{y^+}(x,y) &= \lim_{h\to 0} \frac{1}{h}\big[S(x,y+h) - S(x,y)\big], \\ S_{y^-}(x,y) &= \lim_{h\to 0} \frac{1}{h}\big[S(x,y) - S(x,y-h)\big]. \end{aligned}$$

With this notation, the derivatives S_{x^+} and S_{y^-} corresponds to the partial derivatives of S in the south-east region relative to the shock path. Similarly, the derivatives S_{x^-} and S_{y^+} corresponds to the derivatives of S in the north-west region relative to the shock path. Further, using the assumption that the shock paths are monotone increasing, the differential equation (2.14) can be written as either of

$$S_{x^+}S_{y^-} = \frac{1}{4}Q^2, \qquad S_{x^-}S_{y^+} = \frac{1}{4}Q^2,$$

which holds even if we are on a shock path.

Proposition 2.6.4. *Shock solutions appear when either of the following inequalities are satisfied*

$$S_{x^+}S_{y^+} > \frac{1}{4}Q^2, \qquad S_{x^-}S_{y^-} > \frac{1}{4}Q^2. \tag{2.15}$$

Proof. Either applying $S_{x^-}S_{y^+} = \frac{1}{4}Q^2$ to the first inequality or applying $S_{x^+}S_{y^-} = \frac{1}{4}Q^2$ to the second inequality, we obtain

$$S_{x^+} > S_{x^-}, \quad \text{or} \quad S_{y^+} > S_{y^-},$$

respectively. This implies that we have a shock solution. □

Now that we know when shock paths arises, we need to address the evolution of the shock paths. This is described in the following proposition.

Proposition 2.6.5. *The evolution of shock paths are governed by either of the following differential equations*

$$\begin{aligned} S_{x^+}\dot{x} &= S_{y^+}\dot{y}, \\ S_{x^-}\dot{x} &= S_{y^-}\dot{y}. \end{aligned}$$

Proof. The value of S in the south-east and north-west regions on each side of the shock path, can be seen as two independent surfaces. With this interpretation, the shock path can be seen as the intersecting path between the two surfaces. In other words, it is the path for which the value of S is equal for the south-east and north-west regions. Furthermore, this implies that $\dot{S}$ must be the same for both regions along this path. In other words, we have that

$$S_{x^+}\dot{x} + S_{y^-}\dot{y} = S_{x^-}\dot{x} + S_{y^+}\dot{y}. \tag{2.16}$$

We start by assuming $Q = 0$. First of all, not left and right partial derivatives of S can be zero (in this case we would not have a shock). However, (2.15) must be satisfied, meaning that we must have either $S_{x^+} = 0$ or $S_{y^-} = 0$ and either $S_{x^-} = 0$ or $S_{y^+} = 0$. We have two cases:

(i) $S_{x^-} = S_{y^-} = 0$. In this case, (2.16) reads $S_{x^+}\dot{x} = S_{y^+}\dot{y}$, as desired. The other differential equation we wanted to prove trivially holds as it in this case reads $0 = 0$. This argument also holds if we assume $S_{x^+} = S_{y^+} = 0$.

(ii) $S_{x^+} = S_{x^-} = 0$. In this case, all optimal paths are horizontal, i.e. they satisfy $\dot{y} = 0$. This also holds for the shock path. Hence the differential equations are satisfied. A similar argument holds for $S_{y^+} = S_{y^-} = 0$.

Now assume $Q > 0$. This also means that all left and right directional derivatives of S are positive. Rearranging the terms of (2.16) gives

$$(S_{x^+} - S_{x^-})\dot{x} = (S_{y^+} - S_{y^-})\dot{y}.$$

Inserting the identity $S_{y^+}S_{x^-} = S_{y^-}S_{x^+} = \frac{1}{4}Q^2$ on the right hand side, we obtain

$$(S_{x^+} - S_{x^-})\dot{x} = \frac{1}{4}Q^2\left(\frac{1}{S_{x^-}} - \frac{1}{S_{x^+}}\right)\dot{y}.$$

Multiply by $S_{x^+}S_{x^-}$ on both sides to obtain

$$S_{x^+}S_{x^+}(S_{x^+} - S_{x^-})\dot{x} = \frac{1}{4}Q^2(S_{x^+} - S_{x^-})\dot{y}.$$

Finally, dividing by $(S_{x^+} - S_{x^-})$, and applying either of the equalities $S_{y^+}S_{x^-} = \frac{1}{4}Q^2$ or $S_{y^-}S_{x^+} = \frac{1}{4}Q^2$, we obtain either of $S_{x^+}\dot{x} = S_{y^+}\dot{y}$ or $S_{x^-}\dot{x} = S_{y^-}\dot{y}$, concluding the proof. □

We have in this section presented new differential equations which gives new insight into the behaviour of the optimal reparametrization paths. By assuming that S is differentiable, we derived a hyperbolic differential equation for S, as given in (2.14). Additionally, we derived a differential equation for the optimal reparametrization path, as given in (2.13), and showed how reparametrization paths can be considered characteristics for the hyperbolic partial differential equation.

3 | Dynamic Programming

In the problem of finding optimal reparametrizations, we are only concerned with monotone reparametrizations. In other words, the curves in question are assumed to have a defined orientation which should be preserved through the reparametrization. This is especially useful if we know some feature point of the curves. Then, the problems of finding optimal reparametrizations *before* and *after* this feature point are separate, independent problems. This property can be translated to the concept of optimal substructure within computer science, and can be used to construct efficient solvers for the problem in question, if used correctly.

In this chapter, we will review previous dynamic programming based methods, and present a new framework for computing solutions to the optimization problem. The framework can in general be applied to any (orienting preserving) optimal reparametrization problem for curves on bounded domains. However, we will see how the optimization problem (2.5) is particularly suitable for the framework.

3.1 A Fully Discretized Method

One idea is to consider a full discretization of the problem by constructing the solutions as piecewise linear functions by connecting grid nodes on a rectilinear grid. An example of such a path can be seen in Figure 3.1. This method approximates the variational optimization problem by an alternative optimization problem that is both discrete and finite. Additionally, this alternative problem has overlapping subproblems and optimal substructure, meaning that a dynamic programming approach is suitable. To keep the monotonicity of the solution, we need to constrain which nodes are allowed to connect to each other. Therefore, to allow a connection $(k, l) \to (i, j)$, we require $k < i$ and $l < j$. This ensures that no loops are allowed, a property which is essential to the optimal substructure property.

The dynamic programming method is common for problems concerned with curve alignment. We refer to [5] for implementation details for the reparametrization problem within the square root velocity framework. The method is not, however, limited to this framework and has been constructed or similar reparametrization problems [12]. Although the method is simplistic and easily implemented,

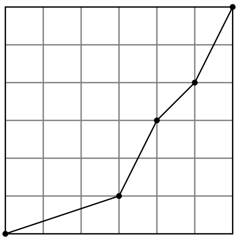


Figure 3.1: Example of a piecewise linear path defined by connecting grid points in a regular grid.

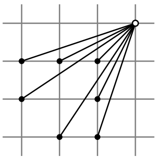


Figure 3.2: Example of restrictions to node connections.

the method has some drawbacks, especially related to computational complexity. Assuming a grid with $n \times n$ nodes, there are $\mathcal{O}(n^4)$ ways of constructing monotone paths between the grid nodes, starting at $(0,0)$ and ending at $(1,1)$. This computational complexity is not compatible with practical applications, and it is common to apply constraints to the node connections. One way to do this, is by only allow nodes in a neighborhood of (i,j), to connect to (i,j). An example of this is illustrated in Figure 3.2. To ensure convergence, however, the size neighborhood must depend on n. This is because in the limit, the path should be able to attain all possible slopes at all points. This requires that the set of slopes possible should in the limit be dense in the positive real numbers. If the neighborhood is chosen as in Figure 3.2 for all n, we would not have convergence since only a finite possible slopes are available. To ensure convergence, the asymptotic complexity of this method will therefore be $\mathcal{O}(n^2|\mathcal{N}(n)|)$, where $|\mathcal{N}(n)|$ is the size of the neighbourhoods.

To combat the substantial computational costs, approximative improvements of the dynamic programming methods has been studied in [13, 14]. These methods are based on a key property of the optimization problem: Assume that (φ_1, φ_2) is a good estimate of a true solution, and consider the reparametrized optimization problem

$$\sup_{\psi_1, \psi_2 \in \Phi(I)} F(\varphi_1 \circ \psi_1, \varphi_2 \circ \psi_2). \tag{3.1}$$

If we assume that the curves c_1, c_2 are C^1 as in [9], the above optimization problem is equivalent to (2.5). Further, since (φ_1, φ_2) is assumed to be a good approximation to the true solution, we would expect the solution (ψ_1^*, ψ_2^*) of (3.1) to be close to the pair identity functions. At least, we expect the graph of the functions to be inside a strip along the diagonal of the unit square, as visualized in Figure 3.3.

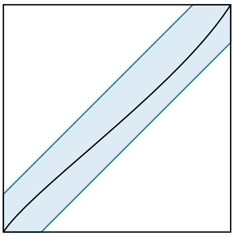


Figure 3.3: Expected domain for solutions to (3.1).

Therefore, one can improve the similarity (or distance) estimate by obtaining a rough approximation (φ_1, φ_2) and using (3.1) to improve this estimate. This has proved to be more efficient than the basic dynamic programming approach. However, the method requires an efficiency-optimality tradeoff. The narrower the the strip along the diagonal, the more efficient the method. However, the method is no longer guaranteed to solve the optimization problem, and with a narrower strip, it is more likely that the true solution is outside the strip.

3.2 A Semi-Discretized Method

Piecewise linear functions, constructed by connecting grid nodes, are not very flexible, as they only allow the direction to change at the grid nodes. In this thesis, we will will loosen this restriction, and allow the solution to change direction at intersections of grid lines. One example of such a path is visualized in Figure 3.4.

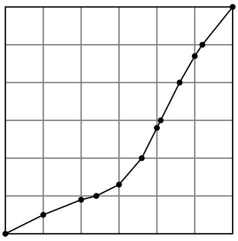


Figure 3.4: Example of a piecewise linear path only allowed to change slope when intersecting grid lines on a regular grid.

To approximate solutions to the optimization problem (2.8), we will do a grid search, similarly to the previous method. Where the two method differs, is in the update equation at each grid point. The previous method approximates solutions by discretizing both x, y and the directions $\dot{x}, \dot{y}$. While we will still discretize x, y, we will construct a method which is continuous in $\dot{x}, \dot{y}$. In other words, we will construct a semi-discretized method. In the update equation for each grid point, this can be done by optimizing over the south and west boundaries of the belonging grid cell, rather than a neighboring set of grid nodes. This idea can be

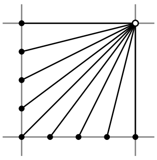


Figure 3.5: Linear paths connected to the north east point on a grid cell.

seen in Figure 3.5. We will see that using consistent approximations to c_1, c_2, we obtain a consistent approximation to S. Additionally, we will see how the method is analogous to a first order finite difference scheme for the nonlinear differential equation for S, derived in Proposition 2.6.3.

The fully discretized dynamic programming approach has one great benefit. Once we have iterated through the whole grid, we can easily use backtracking to retrieve the optimal reparametrization path. For all grid nodes, we just have to store the previous grid node, i.e. the optimal predecessor. For the new semi-discretized approach, we have to retrieve the optimal reparametrizations in a slightly different way. We will see that if we assume a bilinear approximation to S, a solution path can be constructed explicitly using Proposition 2.6.2. The new method will ignore shock solutions and have an $\mathcal{O}(n^2)$ computational complexity.

In the following derivations, we will apply the optional constraint $\dot{\varphi}_1 + \dot{\varphi}_2 = 2$ to avoid the redundancy of the problem. In that context, we will introduce a new notation for the search space to keep the derivations compact. Consider

$$\begin{aligned}\Psi(x_0, y_0, x_1, y_1) = \{(\varphi_1, \varphi_2) \mid\, & \varphi_1 \in \Phi([t_0, t_1], [x_0, x_1]),\\ & \varphi_2 \in \Phi([t_0, t_1], [y_0, y_1]),\\ & \dot{\varphi}_1 + \dot{\varphi}_2 = 2,\\ & t_0 = (x_0 + y_0)/2,\\ & t_1 = (x_1 + y_1)/2\}.\end{aligned}$$

Then, we can define our optimization problem as

$$S(x, y) = \sup_{(\varphi_1, \varphi_2) \in \Psi(0,0,x,y)} F_{[0,(x+y)/2]}(\varphi_1, \varphi_2).$$

3.3 Linear Curves

The building blocks of a successful dynamic programming method are the base cases, where we do not divide the problem into smaller subproblems. It is important to have a well defined base case to ensure that the method as a whole is both efficient and correct. For our method, the base case will be each grid cell on the rectilinear grid. For sufficiently small grid cells, the curves are approximately linear, which means that $\langle q_1(x), q_2(y)\rangle$ is approximately constant. To motivate the derivation of the dynamic programming method, we will therefore consider a special case of the problem where we assume that $\langle q_1(x), q_2(y)\rangle = Q_0$ is constant, for which we for now will assume to be positive.

To start, we will consider the most trivial case where $Q_0 = 1$ on the entire unit square. In this case, the optimization problem reads

$$s(c_1, c_1) = \sup_{\varphi_1,\varphi_2 \in \Psi(0,0,1,1)} \int_I \sqrt{\dot\varphi_1 \dot\varphi_2} dt.$$

The supremum can easily be found via the AM-GM inequality, which gives

$$\int_I \sqrt{\dot\varphi_1 \dot\varphi_2} dt \leq \int_I \frac{1}{2}(\dot\varphi_1 + \dot\varphi_2) dt = \frac{1}{2}(\varphi_1(1) + \varphi_2(1)) = 1.$$

Further, we have equality if and only if $\dot\varphi_1 = \dot\varphi_2$ for all t. This, together with $\dot\varphi_1 + \dot\varphi_2 = 2$ gives $\dot\varphi_1 = \dot\varphi_2 = 1$. In other words, the problem is uniquely solved by the pair of identity functions. Note that without the constraint $\dot\varphi_1 + \dot\varphi_2 = 2$, we will still have the same supremum. But in this case, the solutions are only identified by $\dot\varphi_1 = \dot\varphi_2$. Still, this is quite intuitive since the problem is (without any constraints) invariant under reparametrizations. The result can be generalized for positive correlations $Q_0 > 0$ on general rectangles $[x_0, x_1] \times [y_0, y_1]$. In this case, the problem reads

$$P(x_0, y_0, x_1, y_1) = \sup_{\varphi_1,\varphi_2 \in \Psi(x_0,y_0,x_1,y_1)} \int_I Q_0 \sqrt{\dot\varphi_1 \dot\varphi_2} dt,$$

which has supremum $P(x_0, y_0, x_1, y_1) = Q_0\sqrt{(x_1 - x_0)(y_1 - y_0)}$ and solution given by the pair of linear functions given by

$$\begin{aligned} \varphi_1(t) &= x_0 + \frac{x_1 - x_0}{t_1 - t_0}(t - t_0), \\ \varphi_2(t) &= y_0 + \frac{y_1 - y_0}{t_1 - t_0}(t - t_0). \end{aligned}$$

on the interval $[t_0, t_1] = [\frac{1}{2}(x_0+y_0), \frac{1}{2}(x_1+y_1)]$. This generalization follows directly from applying change of domain, and that the problem is invariant under (positive) rescaling. With these simple expressions, we can also find analytic solutions for problems with linear boundaries and linear boundary conditions.

Proposition 3.3.1. *Consider solution paths starting at the boundary $x \geq 0$, $y = 0$, and assume that $S(x, 0) = ax + b$ and $Q(x, y) = Q_0 \geq 0$ for $x, y \geq 0$ with $a > 0$. Then, we have that*

$$S(x, y) = \begin{cases} b + ax + \frac{Q_0^2}{4a} y, & 4a^2 x \geq Q_0^2 y, \\ b + Q_0\sqrt{xy}, & 4a^2 x < Q_0^2 y, \end{cases}$$

for $x, y > 0$.

Proof. Consider any point (x, y) such that $x, y > 0$. Due to the monotonicity of reparametrization paths, we can find the value of S by optimizing over the part of the x-axis which is allowed to reach (x, y). This is the the interval $[0, x]$ which means that our optimization problem becomes

$$\begin{aligned} S(x,y) &= \sup_{0 \le x^* \le x} S(x^*, 0) + \sup_{(\varphi_1,\varphi_2)\in\Psi(x^*,0,x,y)} \int_I Q_0 \sqrt{\dot\varphi_1 \dot\varphi_2} dt \\ &= \sup_{0 \le x^* \le x} ax^* + b + Q_0 \sqrt{(x - x^*)y}. \end{aligned}$$

Defining $\xi^2 = (x - x^*)$, we can rewrite this equation into

$$S(x,y) = \sup_{0 \le \xi \le \sqrt{x}} -a\xi^2 + Q_0 \sqrt{y}\xi + ax + b, \tag{3.2}$$

which is a quadratic optimization problem with a unique solution since $a > 0$. The unconstrained optimization has solution $\xi^* = Q_0\sqrt{y}/(2a)$. Therefore, since the optimization problem is concave, the constrained optimization problem has solution

$$\xi^* = \mathcal{P}_{[0,\sqrt{x}]}\left(\frac{Q_0\sqrt{y}}{2a}\right),$$

where $\mathcal{P}_{[a,b]}(x) = \max\{\min\{x, b\}, a\}$ is a projection operator. If the maximum is attained at $\xi^* = \sqrt{x}$, the maximum is $b + Q_0\sqrt{xy}$, and if $\xi^* < \sqrt{x}$, the maximum is given by

$$S(x,y) = ax + \frac{Q_0^2}{4a}y + b,$$

concluding the proof. □

A typical solution as constructed in Proposition 3.3.1 is visualized in Figure 3.6. In Figure 3.6a, the blue lines denote the solutions where the optimal value was attained at $\xi^* < \sqrt{x}$, while the black lines denote the rarefaction part where $\xi^* = \sqrt{x}$. A similar result to Proposition 3.3.1 holds when we assume the paths to start from a vertical boundary, and this situation is visualized in Figure 3.7. In general, the proposition can be modified to work for any linear boundary defined from the equation $cx + dy = 0$ as long as $c, d \ge 0$ and not both $c = 0$ and $d = 0$. However, we will for now consider either horizontal or vertical boundaries since we want to approximate the solution on rectangular grid cells.

For the horizontal boundary in Proposition 3.3.1, the optimal paths are unique for $a > 0$, and the direction of the path is given by $(\dot x, \dot y) \propto ((\xi^*)^2, y)$. For $4a^2x \ge Q_0^2 y$, i.e. in the linear part, we have that

$$S_x = a, \quad S_y = \frac{Q_0^2}{4a}, \quad \dot x \propto \frac{Q_0^2}{4a^2}y, \quad \dot y \propto y.$$

Observe that the solutions satisfy both $S_x\dot x - S_y\dot y = 0$ and $S_xS_y = \frac{1}{4}Q_0^2$. In other words, both differential equations derived in Section 2.6 are satisfied. For the region where $4a^2x < Q_0^2y$, i.e. in the rarefaction part, we have that

$$S_x = \frac{Q_0}{2}\sqrt{\frac{y}{x}}, \quad S_y = \frac{Q_0}{2}\sqrt{\frac{x}{y}}, \quad \dot x \propto x, \quad \dot y \propto y.$$

Again we have that both $S_x\dot x - S_y\dot y = 0$ and $S_xS_y = \frac{1}{4}Q_0^2$ are satisfied.

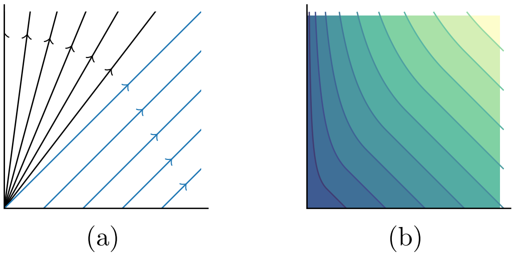


Figure 3.6: Visualization of optimal paths (a) and values of S (b) for the problem discussed in Proposition 3.3.1.

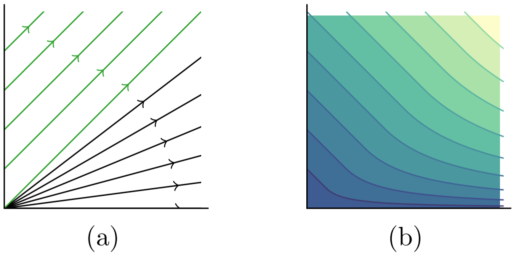


Figure 3.7: Visualization of optimal paths (a) and values of S (b) for the problem discussed in Proposition 3.3.1, but now for paths starting from $(x, y) = (0, \cdot)$.

3.3.1 Shock and Rarefaction Waves

Until now, we constrained the solutions such that they were only allowed to start from one of the boundaries. However, we clearly need to allow the solutions to start from either of the south and west the boundaries. Again, this can be formulated as a maximization problem over the boundaries. To do this, we consider each boundary separately, which means that we for each point (x, y) have two possible values of S and two "optimal" paths, one starting at each boundary. Then, by comparing the values of S, the true optimal path can be determined. In other words, if S^S denotes the value of S when paths must start from the south boundary, and S^W denotes the value of S when the paths must start from the west boundary, we obtain S through

$$S(x, y) = \max\{S^S(x, y), S^W(x, y)\}.$$

The optimal direction at this point is given by

$$(\dot{x}, \dot{y}) = \begin{cases} (\dot{x}, \dot{y})^S, & S(x, y) = S^S(x, y), \\ (\dot{x}, \dot{y})^W, & S(x, y) = S^W(x, y), \end{cases}$$

where $(\dot{x}, \dot{y})^S$ and $(\dot{x}, \dot{y})^W$ denotes the optimal directions when the paths start at the south and west boundaries, respectively.

The above construction of the optimal solution paths is unambiguous wherever $S^S(x, y) \neq S^W(x, y)$. If the two values are equal, however, we have two choices for the optimal direction $(\dot{x}, \dot{y})$. Note that since the optimal paths are straight lines, we will always have that the slopes satiefies $(\dot{y}/\dot{x})^S \geq (\dot{y}/\dot{x})^W$. This means that we have two cases. Either, the slopes are equal, given by $\dot{y}/\dot{x} \propto y/x$. In this case, the optimal path originates in the origin, and we get a rarefaction wave. If the slopes are not equal on the other hand, we get a shock wave. These cases are visualized in Figure 3.8 and 3.9, respectively.

To see which cases gives shock and rarefaction waves, consider a corner boundary, with piecewise linear initial condition

$$S(x, y) = \begin{cases} a^S(x - x_0) + b, & x > 0, y = 0, \\ a^W(y - y_0) + b, & x = 0, y > 0, \end{cases}$$

for some $a^S, a^W \geq 0$. For the non-rarefaction part of the solution, the optimal slopes are given by

$$\left(\frac{dt}{dx}\right)^S = \frac{Q_0^2}{4(a^S)^2}, \quad \left(\frac{dt}{dx}\right)^W = \frac{4(a^W)^2}{Q_0^2}.$$

We get shock solutions if the optimal paths from each of the boundaries are colliding, and rarefaction waves if the optimal paths are diverging. Further, we will observe colliding optimal paths whenever $(dy/dx)^S > (dy/dx)^W$. In other words, we get shock solutions if $a^S a^W > \frac{1}{4}Q_0^2$. Similarly, we get diverging optimal paths and rarefaction waves if $a^S a^W < \frac{1}{4}Q_0^2$. At the origin, a^S is equivalent to S_x and a^W is equivalent to S_y. Hence, the condition for a shock solution can be written as $S_x S_y > \frac{1}{4}Q^2$, which is exactly the condition we obtained in Proposition 2.6.4.

3.3.2 Shock Paths

Given that we have a shock path, this path can be found from the intersection between the surfaces $(x, y) \mapsto S^S(x, y)$ and $(x, y) \mapsto S^W(x, y)$. In other words, the path is defined from

$$\sup_{0 \leq x^* \leq x} a^S x^* + b + Q_0\sqrt{(x - x^*)y} = \sup_{0 \leq y^* \leq y} a^W y^* + b + Q_0\sqrt{x(y - y^*)}$$

As before, we introduce $\xi^2 = (x - x^*)$, and now also $\zeta^2 = (y - y^*)$ which gives

$$\sup_{0 \leq \xi \leq \sqrt{x}} a^S(x - \xi^2) + Q_0\sqrt{(y - y_0)}\xi = \sup_{0 \leq \zeta \leq \sqrt{y}} a^W(y - \zeta^2) + Q_0\sqrt{(x - x_0)}\zeta.$$

Since the optimal paths are straight lines and we assume that there is a shock, the solutions of these optimziation problems will be attained at $\xi^* < \sqrt{x}$ and $\zeta^* < \sqrt{y}$, which means that the supremums are

$$a^S x + \frac{Q_0^2}{4a^S}y = a^W y + \frac{Q_0^2}{4a^W}x.$$

After simplification, this gives $a^S x = a^W y$, which is equivalent to the shock paths $S_{x^+}\dot{x} = S_{y^+}\dot{y}$, derived in Proposition 2.6.5.

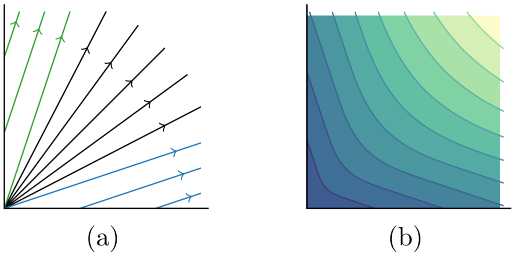


Figure 3.8: Diverging optimal paths (a) resulting in a rarefaction wave for S, shaded in (b).

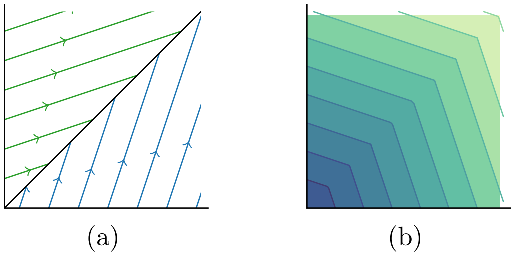


Figure 3.9: Colliding optimal paths (a) resulting in a shock wave for S, shaded in (b).

3.4 A General Dynamic Programming Framework

We will now define the general framework for the dynamic programming algorithm. Consider a region $\mathcal{N}(x, y) \subset \mathbb{R}^2$ which separates the rectangle $[0, x] \times [0, y]$ into two regions, one containing the origin $(0, 0)$ and one containing (x, y). An example of such a region is visualized in Figure 3.10. We will in general assume that $\mathcal{N}(x, y)$ is unit dimensional, and denote such regions as *separating paths*. Since the path separates the origin from the point (x, y), we know that any monotone increasing path from $(0, 0)$ to (x, y) must pass through $\mathcal{N}(x, y)$.

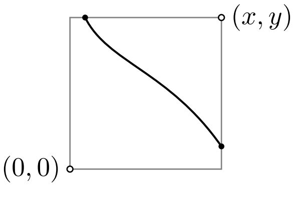


Figure 3.10: A separating path $\mathcal{N}(x, y)$.

We want to show that using separating paths, we can decompose the search space by function concatenation. To do so, we need to generalize the concept of function concatenation to sets of functions. Specifically, for two sets A and B of absolutely continuous functions, we define

$$A \oplus B = \{a \oplus b \mid a \in A, b \in B\}.$$

This requires that the sets A and B are compatible in the sense that the "last" point of all functions $a \in A$ equals the "first" point of all functions $b \in B$. In particular, $\Psi(0, 0, x^*, y^*)$ and $\Psi(x^*, y^*, x, y)$ are by construction compatible.

Lemma 3.4.1. *For any separating path $\mathcal{N}(x, y)$, we can decompose our search space by the relation*

$$\Psi(0, 0, x, y) = \bigcup_{(x^*,y^*)\in\mathcal{N}(x,y)} \Psi(0, 0, x^*, y^*) \oplus \Psi(x^*, y^*, x, y).$$

Proof. Since $\mathcal{N}(x, y)$ is a separating path, we have the decomposition

$$\Psi(0, 0, x, y) = \bigcup_{(x^*,y^*)\in\mathcal{N}(x,y)} \Psi(0, 0, x, y)|_{x^*,y^*},$$

where

$$\begin{aligned}\Psi(0, 0, x, y)|_{x^*,y^*} = \{(\varphi_1, \varphi_2) \in \Psi(0, 0, x, y) \,|&\varphi_1(t^*) = x^*,\\ &\varphi_2(t^*) = y^*,\\ &t^* = \tfrac{1}{2}(x^* + y^*)\}.\end{aligned}$$

Now, the restriction of any absolutely continuous function can be done by the concatenation of two absolutely continuous functions. For all $\varphi \in \Phi([0, t], [0, x])$, there exists $\psi \in \Phi([0, t^*], [0, x^*])$ and $\vartheta \in \Phi([t^*, t], [x^*, x])$ such that $\varphi = \psi \oplus \vartheta$. The reverse also holds in the sense that for all $\psi \in \Phi([0, t^*], [0, x^*])$ and $\vartheta \in \Phi([t^*, t], [x^*, x])$, then $\psi \oplus \vartheta \in \Phi([0, t], [0, x])$. In other words, we can identify

$$\Psi(0, 0, x, x)|_{x^*,y^*} = \Psi(0, 0, x^*, y^*) \oplus \Psi(x^*, y^*, x, y),$$

concluding the proof. □

Lemma 3.4.1 can be used to reformulate the optimization problem (2.8). Firstly, recall that our functional F is additive under concatenation, as seen in Section 2.3. In other words, if $(\psi_1, \psi_2) \in \Psi(0, 0, x^*, y^*)$ and $(\vartheta_1, \vartheta_2) \in \Psi(x^*, y^*, x, y)$, then

$$F_{[0,t]}(\psi_1 \oplus \vartheta_1, \psi_2 \oplus \vartheta_2) = F_{[0,t^*]}(\psi_1, \psi_2) + F_{[t^*,t]}(\vartheta_1, \vartheta_2).$$

Using this property, together with Lemma 3.4.1, we obtain the following theorem.

Theorem 3.4.2. *Let $\mathcal{N}(x, y)$ be a separating path. Then, we have that*

$$S(x, y) = \sup_{(x^*,y^*)\in\mathcal{N}(x,y)} S(x^*, y^*) + P(x^*, y^*, x, y). \tag{3.3}$$

Proof. We have that

$$
\begin{aligned}
S(x,y) &= \sup_{(\varphi_1,\varphi_2)\in\Psi(0,0,x,y)} F_{[0,(x+y)/2]}(\varphi_1,\varphi_2) \\
&= \sup_{(x^*,y^*)\in\mathcal{N}(x,y)} \sup_{(\varphi_1,\varphi_2)\in\Psi(0,0,x,y)|_{x^*,y^*}} F_{[0,(x+y)/2]}(\varphi_1,\varphi_2) \\
&= \sup_{(x^*,y^*)\in\mathcal{N}(x,y)} \Bigg(\sup_{(\psi_1,\psi_2)\in\Psi(0,0,x^*,y^*)} F_{[0,(x^*+y^*)/2]}(\psi_1,\psi_2) \\
&\qquad\qquad + \sup_{(\vartheta_1,\vartheta_2)\in\Psi(x^*,y^*,x,y)} F_{[(x^*+y^*)/2,(x+y)/2]}(\vartheta_1,\vartheta_2) \Bigg) \\
&= \sup_{(x^*,y^*)\in\mathcal{N}(x,y)} S(x^*,y^*) + P(x^*,y^*,x,y). \qquad \square
\end{aligned}
$$

Theorem 3.4.2 gives a foundation for constructing dynamic programming based solvers for our optimization problem (2.8). But it is not limited to being a solver for this optimization problem in particular. In order to apply the theorem, we only need an optimization problem which has

(i) a search space of monotone, absolutely continuous / Lipchitz functions with a fixed start end end point,

(ii) an integral functional (or any other functional which is additive under concatenation).

If these requirements are fulfilled, we can apply Theorem 3.4.2. However, this dynamic programming framework might be more suitable for certain problems. Specifically, we are interested in problems where we can find a sufficiently nice separating path $\mathcal{N}(x,y)$ such that with sufficiently nice approximations of $S(x^*,y^*)$ and $P(x^*,y^*,x,y)$, the remaining optimization problem (3.3) is easy to solve.

3.5 Local Approximations

We will now see how the base cases discussed in Section 3.3 together with the framework defined in Section 3.4 can be used to construct a consistent approximation to S. To apply the analytical solutions from Section 3.3, we will consider piecewise linear approximations to the curves, or equivalently piecewise constant approximations to the SRVTs. This is equivalent to approximating $Q(x,y)$ as piecewise constant on some rectilinear grid. Since q_1 and q_2 are continuously differentiable, we can approximate Q by

$$Q(x,y) = Q_0 + \mathcal{O}(h)$$

for all x,y in some grid cell $[x_0,x_1]\times[y_0,y_1]$. Here, we define $h=\max\{h_x,h_y\}$ where $h_x = x_1 - x_0$ and $h_y = y_1 - y_0$ denotes the width and height of the grid cell, respectively. We will now focus on using this approximation to get an estimate of S at the north-east point (x_1,y_1), but the following derivations will work for any point in the grid cell. For any point $(x^*, y^*) \in [x_0, x_1] \times [y_0, y_1]$, we have that

$$\begin{aligned} P(x^*, y^*, x_1, y_1) &= \sup_{\varphi_1, \varphi_2 \in \Psi(x^*, y^*, x_1, y_1)} \int_I (Q_0 + \mathcal{O}(h)) \sqrt{\dot{\varphi}_1 \dot{\varphi}_2} dt \\ &= (Q_0 + \mathcal{O}(h)) \sqrt{(x_1 - x^*)(y_1 - y^*)} \\ &= Q_0 \sqrt{(x_1 - x^*)(y_1 - y^*)} + \mathcal{O}(h^2). \end{aligned}$$

As before, we have that any path which passes through the rectangle $[x_0, x_1] \times [y_0, y_1]$, must enter this rectangle through its south or west boundary. In other words, for the problem of finding optimal paths from $(0, 0)$ to (x_1, y_1), the union of the south and west boundaries is a separating path. Now, the only remaining part we need is an approximation of $S(x^*, y^*)$ for any boundary point $(x^*, y^*) \in \mathcal{N}(x_1, y_1)$. To do so, we will use linear approximations between the corner points on the rectangle. In other words, for the south boundary, we approximate

$$S(x^*, y_0) \approx \frac{x_1 - x^*}{h_x} S(x_0, y_0) + \frac{x^* - x_0}{h_x} S(x_1, y_0),$$

and for the west boundary, we approximate

$$S(x_0, y^*) \approx \frac{y_1 - y^*}{h_y} S(x_0, y_0) + \frac{y^* - y_0}{h_y} S(x_0, y_1).$$

To obtain an error estimate for S, we need to assume certain regularity properties for the variable. From experiments, S seems to be C^2-continuous everywhere except the boundaries $(0, \cdot)$ and $(\cdot, 0)$ and certain shock paths. These shock paths seem to occur on a negligible subset of the unit square. Therefore, we will assume S to be piecewise C^2, meaning that we assume that the above approximation is $\mathcal{O}(h^2)$ almost everywhere.

We can now apply the results from Section 3.3 directly. To do so, we need the directional derivatives of S at the boundaries, which are given by

$$a^S = \frac{S(x_1, y_0) - S(x_0, y_0)}{h_x}, \qquad a^W = \frac{S(x_0, y_1) - S(x_0, y_0)}{h_y}.$$

Inserting these expressions into (3.2) and shifting the origin to (x_0, y_0), we obtain

$$S^S(x_1, y_1) = \max_{0 \leq \xi \leq \sqrt{h_x}} S(x_0, y_0) + (x_1 - \xi^2) \frac{S(x_1, y_0) - S(x_0, y_0)}{h_x} + Q_0 \sqrt{h_y} \xi.$$

This can also be done for the paths starting from the west boundary, and after rescaling the variable ξ, we end up with the following approximation for $S(x_1, y_1)$:

$$S(x_1, y_1) = \max\{S^S, S^W\},$$

where

$$\begin{aligned} S^S &= \max_{\xi \in [0,1]} S(x_1, y_0) + Q_0 \sqrt{h_x h_y} \xi - (S(x_1, y_0) - S(x_0, y_0)) \xi^2, \\ S^W &= \max_{\zeta \in [0,1]} S(x_0, y_1) + Q_0 \sqrt{h_x h_y} \zeta - (S(x_0, y_1) - S(x_0, y_0)) \zeta^2. \end{aligned}$$

If the C^2 assumption holds for S, this is an $\mathcal{O}(h^2)$ approximation. Further, this is a quadratic optimization problem on a closed interval, meaning that a solution can be efficiently found in constant time.

The above notation can be adapted to a grid with grid nodes (x_i, y_j) and abbreviations $S_{i,j} = S(x_i, y_j)$ and $Q_{i,j} = \langle q_1(x_i), q_2(x_j)\rangle$. Using this notation, the above update equation can be reformulated as

$$S_{i,j} = \max\{S^S, S^W\}, \tag{3.4}$$

where

$$\begin{aligned} S^S &= \max_{\xi\in[0,1]} S_{i,j-1} + Q_{i,j}\sqrt{h_i h_j}\xi - (S_{i,j-1} - S_{i-1,j-1})\xi^2, \\ S^W &= \max_{\zeta\in[0,1]} S_{i-1,j} + Q_{i,j}\sqrt{h_i h_j}\zeta - (S_{i-1,j} - S_{i-1,j-1})\zeta^2. \end{aligned} \tag{3.5}$$

This is the update equation we will use in the grid search algorithm. The error of these approximations will be $\mathcal{O}(h^2)$ if we can assume that S is piecewise C^2. If S is C^1- but not C^2-continuous, we will have an $\mathcal{O}(h)$ error.

3.5.1 An Alternative Update Equation

It should be noted that there is another choice of the separating path $\mathcal{N}(x, y)$ which also gives an update equation we can compute in constant time. Consider the diagonal between the north west and the south east corner of the rectangle. Any point (x^*, y^*) on this diagonal can be parametrized by

$$\begin{aligned} x^* &= \tfrac{1}{2}(1+\xi)x_0 + \tfrac{1}{2}(1-\xi)x_1, \\ y^* &= \tfrac{1}{2}(1-\xi)y_0 + \tfrac{1}{2}(1+\xi)y_1. \end{aligned}$$

for $-1 \le \xi \le 1$. Note that this implies that

$$\begin{aligned} \sqrt{(x_1 - x^*)(y_1 - y^*)} &= \sqrt{(x_1 - x_0)(y_1 - y_0)}\sqrt{1-\xi^2} \\ &= \sqrt{h_x h_y}\sqrt{1-\xi^2}. \end{aligned}$$

If we now assume a linear approximation to S on the diagonal, and insert the above expressions, we get the following optimization problem

$$S(x_1, y_1) = \sup_{-1\le\xi\le1} \frac{1}{2}(1+\xi)S(x_0, y_1) + \frac{1}{2}(1-\xi)S(x_1, y_0) + Q_0\sqrt{h_x h_y}\sqrt{1-\xi^2},$$

for which the maximum is given by

$$\begin{aligned} S(x_1, y_1) = \frac{1}{2}\bigg(&S(x_0, y_1) + S(x_1, y_0) \\ &+ \sqrt{(S(x_0, y_1) - S(x_1, y_0))^2 + Q_0^2 h_x h_y}\bigg). \end{aligned}$$

Note that in this case, the requirement $Q_0 \geq 0$ is crucial for correctness. On a rectilinear grid, this update equation reads

$$S_{i,j} = \frac{1}{2}\left(S_{i-1,j} + S_{i,j-1} + \sqrt{(S_{i-1,j} - S_{i,j-1})^2 + Q_{i,j}^2 h_i h_j} \right). \tag{3.6}$$

Observe how both update equations relies on the coupled variable $Q_{i,j}\sqrt{h_i h_j}$. One interpretation of the update equations is that we rescale the grid cells to be uint squares in size. Since $Q(x,y) \propto \sqrt{|\dot{c}_1(x)||\dot{c}_2(y)|}$, linearly rescaling the interval $[x_{i-1}, x_i]$ to $[0,1]$ implies rescaling $Q(x,y)$ by $\sqrt{h_i}$. This also holds for rescaling y which implies rescaling $Q(x,y)$ by $\sqrt{h_j}$.

3.5.2 Relationship to Finite Difference Methods

The proposed methods has nice interpretations as a finite difference scheme for the nonlinear PDE $S_x S_y = \frac{1}{4}Q^2$. One way to formulate a finite difference scheme for this equation is by backwards differences, given by

$$\frac{S(x,y) - S(x-h_x,y)}{h_x} \frac{S(x,y) - S(x,y-h_y)}{h_y} = \frac{1}{4}Q(x,y)^2$$

which on a rectilinear grid is given by

$$\frac{S_{i,j} - S_{i-1,j}}{h_i} \frac{S_{i,j} - S_{i,j-1}}{h_j} = \frac{1}{4}Q_{i,j}^2.$$

Observe that this is a quadratic equation for $S_{i,j}$, for which the solution is given by

$$S_{i,j} = \frac{1}{2}\left(S_{i+1,j} + S_{i,j+1} + \sqrt{(S_{i+1,j} - S_{i,j+1})^2 + Q_{i,j}^2 h_i h_j} \right).$$

And “out of the blue,” we obtain the second update equation (3.6). However, to confidently apply finite difference schemes, numerical stability needs to be established. The following example illustrates that the stability is not trivial: We can also obtain the first derived update equation by considering another form of finite differences. Specifically, we use the following approximation

$$\frac{S(x,y-h_y) - S(x-h_i,y-h_y)}{h_i} \frac{S(x,y) - S(x,y-h_y)}{h_y} = \frac{1}{4}Q(x,y)^2,$$

which on a rectilinear grid is given by

$$\frac{S_{i,j-1} - S_{i-1,j-1}}{h_i} \frac{S_{i,j} - S_{i,j-1}}{h_j} = \frac{1}{4}Q_{i,j}^2.$$

This equation is linear in $S_{i,j}$ with solution

$$S_{i,j} = S_{i,j-1} + \frac{Q_{i,j}^2 h_i h_j}{4(S_{i,j-1} - S_{i-1,j-1})}.$$

This is exactly the solution of the unconstrained version of the optimization problem (3.5). However, the unconstrained problem allows the paths to start on an infinitely long boundary, which clearly would mean that the constant approximation to $Q(x, y)$ will not be $\mathcal{O}(h)$. In other words, we cannot simply apply finite difference methods to the nonlinear differential equation without investigating the numerical stability of the finite differences.

3.6 Grid Search

Now that we have an update equation for each grid cell, we can assemble this into a grid search algorithm. Assume that the grid is constructed with grid nodes at the points

$$x_i = g_1(i/n), \quad y_j = g_2(j/n)$$

for $i, j = 0, \dots n$ and some smooth, bijective, monotone increasing maps $g_1, g_2 : [0, 1] \to [0, 1]$. We require the mappings to be smooth to ensure that the grid cells has width and height $h_x, h_y = \mathcal{O}(h)$, where $h = n^{-1}$ is the average width / height of the grid cells. Recall that for $(x, y) \in I \times I$, we have that

$$F(x\varphi_1, y\varphi_2) = \sqrt{xy} \int_0^1 Q(x\varphi_1, y\varphi_2)\sqrt{\dot{\varphi}_1, \varphi_2}dt.$$

Further, for small x and y, this implies that $S(x, y) \sim \sqrt{xy}$. This imposes a problem since we assumed that S is C^2-continuous in the update equation derived in the previous section. This is, however, why we do not consider regular grid, but some irregular rectilinear grid as described above. Specifically, for $g_1, g_2 \approx 0$ we have that $S(g_1(x), g_2(y)) \sim \sqrt{g_1(x)g_2(y)}$. Therefore, by requiring that $g_1(x) \sim x^2$ and $g_2(y) \sim y^2$ for small x and y, the convergence result is maintained.

Given that S is C^2, we have that the truncation error is $\mathcal{O}(h^2)$. This means that we get a global error of $\mathcal{O}(h)$ as we integrate over the grid nodes. If S is not continuously differentiable, we only have a $\mathcal{O}(h)$ truncation error, which integrates to $\mathcal{O}(1)$. However if the assumption that we only have a finite number of shocks hold, and that S is therefore piecewise C^2, we will still maintain an $\mathcal{O}(h)$ error.

Concerning the computational complexity of this algorithm, it should be clear that it is asymptotically $\mathcal{O}(n^2)$. For small n, however, we can use parallelization to achieve a linear apparent running time. The method can be parallelized in the following way: consider the set of nodes for which $i + j = k$, given by

$$A_k = \{(i, j) \mid i, j \in \{0, \dots, n\}, \;\; i + j = k\}.$$

The value of S at these nodes are only dependent on the value of S at the nodes in A_{k-1} and possibly A_{k-2} (depending on whether we use the first or second update equation). This means that finding the value of S at the nodes in A_k are independent problems, which can be ran in parallell. Using update equation (3.4), an outline of the parallelized algorithm can be seen in Algorithm 1. Here, the innermost for-loop can be run in parallel. In the algorithm, we used the first update equation. However, the second update equation can be used by replacing lines 9, 10 and 11 with equation (3.6).

Algorithm 1 Approximate Similarity

```
Require: $q_1, q_2$: SRVT's of $c_1$, $c_2$,
         $g_1, g_2$: Grid transformations,
         $n$: Number of grid points.
Ensure: $S$: Point estimates $S_{i,j} = \hat{S}(g_1(i/n), g_2(j/n))$.

procedure SIMILARITY($q_1$, $q_2$, $g_1$, $g_2$, $n$)
    $x_i \leftarrow g_1(i/n), \quad i = 0, \dots, n$
    $y_j \leftarrow g_2(j/n), \quad j = 0, \dots, n$
    $Q_{i,j}\sqrt{h_i h_j} \leftarrow \max\{\langle q_1(x_i), q_2(y_j)\rangle, 0\}\sqrt{(x_{i+1} - x_i)(y_{j+1} - y_j)},$
                                                                  $i, j = 0, \dots, n$
    $S_{i,0} \leftarrow 0, \quad i = 0, \dots, n$
    $S_{0,j} \leftarrow 0, \quad j = 0, \dots, n$
    for $k = 0, \dots, 2n - 2$ do
        for $i, j \in \{0, \dots, n - 1\}$ s.t. $i + j = k$ do
            $S^S \leftarrow \max_{\xi \in [0,1]} S_{i+1,j} + Q_{i,j}\sqrt{h_i h_j}\xi - (S_{i+1,j} - S_{i,j})\xi^2$
            $S^W \leftarrow \max_{\zeta \in [0,1]} S_{i,j+1} + Q_{i,j}\sqrt{h_i h_j}\zeta - (S_{i,j+1} - S_{i,j})\zeta^2$
            $S_{i+1,j+1} \leftarrow \max\{S^S, S^W\}$
        end for
    end for
end procedure
```

3.7 Retrieving the Optimal Reparametrizations

Recall that we in Proposition 2.6.2 derived a differential equation for the optimal reparametrizers, given by

$$\begin{aligned} S_x \dot{x} - S_y \dot{y} &= 0, \qquad \text{if } S_x + S_y > 0, \\ \dot{x}\dot{y} &= 0, \qquad \text{if } S_x + S_y = 0. \end{aligned} \tag{2.13}$$

The following proposition gives a foundation for computing explicit solutions to the differential equation, when we assume that S is piecewise bilinear.

Proposition 3.7.1. *If $S(x, y)$ is piecewise bilinear, then the solution to the differential equation*

$$\begin{aligned} S_x \dot{x} - S_y \dot{y} &= 0 \\ \dot{x} + \dot{y} &= 2 \end{aligned}$$

is piecewise linear.

Proof. It is sufficient to prove that the solution is linear in any grid cell $[x_0, x_1] \times [y_0, y_1]$. In this grid cell, S is bilinear, and can hence be expressed as

$$S(x, y) = a_0 + a_x x + a_y y + a_{xy} xy,$$

for some constants a_0, a_x, a_y and a_{xy}. In particular, this implies that

$$\begin{aligned} S_x(x,y) &= a_x + a_{xy}y, \\ S_y(x,y) &= a_y + a_{xy}x. \end{aligned}$$

Inserting this into (2.13), we obtain

$$(a_x + a_{xy}y)\dot{x} - (a_y + a_{xy}x)\dot{y} = 0,$$

which has the solution $(a_y + a_{xy}x) = K(a_x + a_{xy}y)$ for some constant K. This, together with $\dot{x} + \dot{y} = 2$ ensures that we have a linear solution. □

Since any solution path is piecewise linear, the slope of the path will be piecewise constant. This means that we only need to know the slope at a single point of any grid cell to know the solution on the entire grid cell. On the grid cell $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, the bilinear variable S can be expressed as

$$S(x,y) = S_{i,j} + \Delta_i S_{i,j}\frac{x - x_i}{h_i} + \Delta_j S_{i,j}\frac{y - y_j}{h_j} + \Delta_i\Delta_j S_{i,j}\frac{(x - x_i)(y - y_j)}{h_i h_j}.$$

Here, $\Delta_i S_{i,j}$ and $\Delta_j S_{i,j}$ denotes the integer forward differences of S in direction x and y, respectively. In this grid cell, assume that we know one point $(x_k, y_k) = (x_i + h_i\mu, y_j + h_j\eta)$ where $\mu, \eta \in [0,1]$ are the values for x and y, normalized to the current grid cell. The partial derivatives of S at this point are given by

$$\begin{aligned} h_i S_x(x_k, y_k) &= \Delta_i S_{i,j} + \eta\Delta_i\Delta_j S_{i,j}\eta \\ &= (1 - \eta)(S_{i+1,j} - S_{i,j}) + \eta(S_{i+1,j+1} - S_{i,j+1}). \end{aligned}$$

and

$$\begin{aligned} h_i S_y(x_k, y_k) &= \Delta_i S_{i,j} + \mu\Delta_i\Delta_j S_{i,j}\mu \\ &= (1 - \mu)(S_{i,j+1} - S_{i,j}) + \mu(S_{i+1,j+1} - S_{i+1,j}). \end{aligned}$$

The values of $h_i S_x(x_k, y_k)$ and $h_j S_y(x_k, y_k)$ provides a direction for the piecewise linear solution in the current grid cell. Further, since the solution path only changes direction when intersecting a grid line, the solution can be expressed using sequences (x_k) and (y_k), where each point (x_k, y_k) lies on a grid line. Whenever a solution passes through a grid cell, we need two points to determine the solution within that grid cell; the point where the solution enters the grid cell, and the point where the solution exits. Due to the monotonicity of (x, y), there are only two ways the backtracking solution can enter a grid cell: through the north boundary or through the east boundary. Similarly, the solution can only exit the grid cell through the south or west boundary. The four ways a solution can enter and exit a grid cell are visualized in Figure 3.11.

The next point in the backtracking sequence, (x_{k-1}, y_{k-1}) can be found by

$$\begin{aligned} x_{k-1} &= x_i + h_i \max\left\{\mu - \eta\frac{h_j S_y}{h_i S_x}, 0\right\}, \\ y_{k-1} &= y_j + h_j \max\left\{\eta - \mu\frac{h_i S_x}{h_j S_y}, 0\right\}. \end{aligned} \tag{3.7}$$

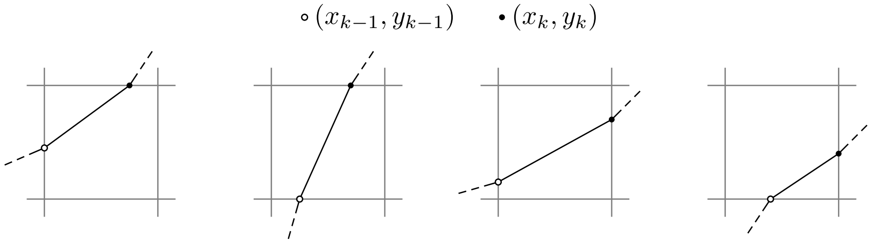


Figure 3.11: The four ways a path can enter and exit a grid cell.

Here, we have used the abbreviations $S_x = S_x(x_k, y_k)$ and $S_y = S_y(x_k, y_k)$. To see why this gives the next point in the backtracking sequence, we refer to the following derivation: since the path is linear in the grid cell, the slope is constantly given by $\dot{x}/\dot{y} = (y_k - y_{k-1})/(x_k - x_{k-1})$. Inserting the above expressions, we obtain

$$\frac{\dot{y}}{\dot{x}} = \frac{y_k - y_{k-1}}{x_k - x_{k-1}} = \frac{h_j\eta - h_j \max\left\{\eta - \mu\frac{h_i S_x}{h_j S_y}, 0\right\}}{h_i\mu - h_i \max\left\{\mu - \eta\frac{h_j S_y}{h_i S_x}, 0\right\}} = \frac{S_x}{S_y}\frac{\min\{\mu h_i S_x, \eta h_j S_y\}}{\min\{\eta h_j S_y, \mu h_i S_x\}} = \frac{S_x}{S_y}.$$

This implies $S_x\dot{x} - S_y\dot{y} = 0$, as desired.

Note that there are a few special cases we need to take care of. The one-step equation (3.7) only holds whenever both $S_x > 0$ and $S_y > 0$. If, on the other hand, $S_x > 0$ but $S_y = 0$, the above equations includes division by zero, which can be ambiguous. In this case, it is not problematic as the differential equation (2.13) gives $\dot{x} = 2$ and $\dot{y} = 0$. This is enforced with the one-step equation

$$y_{k-1} = y_j, \qquad x_{k-1} = x_k.$$

Similarly, if $S_x = 0$ but $S_y > 0$, we use the following one-step equation:

$$y_{k-1} = y_k, \qquad x_{k-1} = x_i.$$

Lastly if both $S_x = 0$ and $S_y = 0$, both horizontal and vertical paths solves the differential equation. In these cases, we will always choose a horizontal path in order to be consistent. Additionally, in the case where we are on the x-axis, given by $x_k = 0$ but $y_k > 0$, the path must be defined from $\dot{x} = 0$ and $\dot{y} = 2$, to ensure that we do not exit the unit square. We have similar constraint for the y-axis. In these cases we also apply one of the special case one-step equations above.

An outline of the backtracking algorithm can be seen in algorithm 2. In the algorithm, we compute a rescaled sequence $(\tilde{x}_k, \tilde{y}_k)_k$ which is scaled such that $x_k = g_1(\tilde{x}_k)$ and $y_k = g_2(\tilde{y}_k)$. This rescaling has several benefits. First of all, the grid cells in the rescaled system are unit squares. Additionally, the current grid cell can easily be found from $(i, j) = (\lceil\tilde{x}_k\rceil - 1, \lceil\tilde{y}_k\rceil - 1)$ which implies that $\mu = \tilde{x}_k - i$ and $\eta = \tilde{y}_k - j$. Here, $\lceil\cdot\rceil$ denotes the ceiling operator.

The estimate for the optimal reparametrization path is found analytically by the linear ordinary differential equation, given a bilinear approximation to S. If the solution is unique, i.e. if $S_x + S_y > 0$ in a region around the optimal path, and that S has an $\mathcal{O}(h)$ error, then both $(\dot{x}, \dot{y})$ and (x, y) will have $\mathcal{O}(h)$ error. However, we expect a slightly better convergence result, which is that $(\sqrt{\dot{x}}, \sqrt{\dot{y}})$ has an $\mathcal{O}(h)$ error. We have two cases:

(i) If both $S_x > 0$ and $S_y > 0$, we have that derivatives satisfy $\dot{x} > 0$ and $\dot{y} > 0$, meaning that the square root of the derivatives will have a linear convergence.

(ii) If in some region we have that $S_x = 0$ but $S_y > 0$, it could seem like we would get $\mathcal{O}(\sqrt{h})$ convergence for $(\sqrt{\dot{x}}, \sqrt{\dot{y}})$. However, in and around this region, we have that $S_x = Q^2/(4S_y)$. Since $\dot{y} \propto S_x$, this means that $\dot{y}$ approach this region at least quadratically, which again implies that $\sqrt{\dot{y}}$ approach the region at least linearly. A similar argument holds for whenever $S_x > 0$ and $S_y = 0$.

In other words, we expect the square root of the derivatives to have linear convergence, given that the solution is unique. This reads

$$\begin{aligned} x(t) &= x_k + \mathcal{O}(h), \\ y(t) &= y_k + \mathcal{O}(h), \\ \sqrt{\dot{x}(t)} &= \sqrt{\Delta x_k/\Delta t_k} + \mathcal{O}(h), \\ \sqrt{\dot{y}(t)} &= \sqrt{\Delta y_k/\Delta t_k} + \mathcal{O}(h), \end{aligned}$$

for all $t \in [t_k, t_{k+1}] = [\frac{1}{2}(x_k + y_k), \frac{1}{2}(x_{k+1} + y_{k+1})]$. In particular, this means that the estimated optimal reparametrizations have linear error. This reads

$$\begin{aligned} q_1(x(t))\sqrt{\dot{x}(t)} &= q_1(x_k)\sqrt{\Delta x_k/\Delta t_k} + \mathcal{O}(h), \\ q_2(y(t))\sqrt{\dot{y}(t)} &= q_2(y_k)\sqrt{\Delta y_k/\Delta t_k} + \mathcal{O}(h) \end{aligned} \tag{3.8}$$

on the same interval. This holds since the sequence produced by the backtracking algorithm satisfies $\Delta t_k = \mathcal{O}(h)$. However, further investigation is needed to get a rigorous proof of this property.

If the path passes through a region where $S_x + S_y = 0$, the solution will not be unique, and evaluation of convergence of the optimal path, might be problematic.

3.7.1 Improving the Similarity Estimate

Interestingly, we can use the approximations to x and y to obtain a better estimate of $s(c_1, c_2)$. For both methods, we can evaluate the fitness of the path (x, y) via the approximation to $S(1, 1)$. And for the previous dynamic programming method, this approximation is equal to the approximated functional

$$F((x_k), (y_k)) = \sum_k \langle q_1(x_k), q_2(y_k)\rangle \sqrt{\Delta x_k \Delta y_k}, \tag{3.9}$$

for some optimal sequence of points $(x_k, y_k)_k$. As for the previous method, the new approach computes a sequence for which we can evaluate the functional F as

Algorithm 2 Backtracking

```
Require: $S$: Point estimates $S_{i,j} = S(g_1(i/n), g_2(j/n))$,
         $g_1, g_2$: Grid transformations.
Ensure: $(x_k, y_k)_k$: Point estimates $(x(t_k), y(t_k)) = (x_k, y_k)$.

procedure OptimalReparametrisations($S$, $g_1$, $g_2$)
    $x_0 \leftarrow n$
    $y_0 \leftarrow n$
    $k \leftarrow 0$
    while $x_k > 0$ and $y_k > 0$ do
        $i \leftarrow \lceil x_k \rceil - 1$
        $j \leftarrow \lceil y_k \rceil - 1$
        $\mu \leftarrow x_k - i$
        $\eta \leftarrow y_k - j$
        $\Delta_x S \leftarrow (1-\eta)(S_{i,j+1} - S_{i,j}) + \eta(S_{i+1,j+1} - S_{i+1,j})$
        $\Delta_y S \leftarrow (1-\mu)(S_{i+1,j} - S_{i,j}) + \mu(S_{i+1,j+1} - S_{i,j+1})$
        if $\Delta_x S = 0$ or $y = 0$ then
            $x_{k-1} \leftarrow i$
            $y_{k-1} \leftarrow y_k$
        else if $\Delta_y S = 0$ or $x = 0$ then
            $x_{k-1} \leftarrow x_k$
            $y_{k-1} \leftarrow j$
        else
            $x_{k-1} \leftarrow i + \max\{\mu - \eta \Delta_x S / \Delta_y S, 0\}$
            $y_{k-1} \leftarrow j + \max\{\eta - \mu \Delta_y S / \Delta_x S, 0\}$
        end if
        $k \leftarrow k - 1$
    end while
    $K \leftarrow -k$                                          ▷ Length of the sequence.
    $x_l \leftarrow g_1(x_{k-K}/n), \quad k = 0, \ldots, K$    ▷ Normalize the sequence.
    $y_l \leftarrow g_2(y_{k-K}/n), \quad k = 0, \ldots, K$
    $t_k \leftarrow (x_k + y_k)/2, \quad k = 0, \ldots, K$
end procedure
```

above. However, this estimate will not be equal to the estimate of $S(1,1)$ since the approximation of S uses a grid search algorithm, while the path x, y need not pass through the grid points. By constructing a similar estimate to (3.9), we can obtain a better estimate of the total similarity. Taking the inner product of the optimal reparametrizers, applying the approximations in (3.8) and multiplying with Δt_k, we obtain

$$\langle q_1(x(t)), q_2(y(t))\rangle \sqrt{\dot{x}(t)\dot{y}(t)}\Delta t_k = \langle q_1(x_k), q_2(y_k)\rangle \sqrt{\Delta x_k \Delta y_k} + \mathcal{O}(h^2).$$

This holds for all $t \in [t_k, t_{k+1}]$. In other words, we expect to have an $\mathcal{O}(h^2)$ estimate of $F(x,y) = s(c_1, c_2)$. Note that this convergence should apply even if the solution is not unique.

3.8 Computation of Geodesics

We are working in the L^2-topology for the SRVTs, which implies that the geodesic between q_1 and q_2 is given as a straight line between the two. In other words, a geodesic takes the form $q(\tau) = (1-\tau)q_1 + \tau q_2$. If $(q_1 \circ x)\sqrt{\dot{x}}$ and $(q_2 \circ y)\sqrt{\dot{y}}$ are optimally reparametrized, we have that

$$q(\tau)(t) = (1-\tau)q_1(x(t))\sqrt{\dot{x}(t)} + \tau q_2(y(t))\sqrt{\dot{y}(t)}.$$

Inserting the approximations in (3.8), we get

$$q(\tau)(t) = (1-\tau)q_1(x_k)\sqrt{\frac{\Delta x_k}{\Delta t_k}} + \tau q_2(y_k)\sqrt{\frac{\Delta y_k}{\Delta t_k}} + \mathcal{O}(h).$$

for all $t \in [t_k, t_{k+1}]$. In other words, we have a consistent approximation to the geodesic in SRVT form. However, we want to retrieve the geodesic curve as well.

Recall that given the SRVT, the original curve can be retrieved through the integral

$$c(\tau)(t) = \int_0^t q(\tau)(s)|q(\tau)(s)|ds.$$

Since $\Delta t_k = \mathcal{O}(h)$, the approximating sequence can be used through a Riemann sum to obtain an estimate of $c(\tau)$. We have that

$$\begin{aligned} c(\tau)(t) &= \sum_{i=0}^{k-1} \big(q(\tau)(t_i)|q(\tau)(t_i)|\Delta t_i + \mathcal{O}(h^2)\big) \\ &= \sum_{i=0}^{k-1} \big(q\sqrt{\Delta t}\big)_i \Big|\big(q\sqrt{\Delta t}\big)_i\Big| + \mathcal{O}(h) \end{aligned}$$

for all $t \in [t_k, t_{k+1}]$, where we define

$$\big(q\sqrt{\Delta t}\big)_i := (1-\tau)q_1(x_i)\sqrt{\Delta x_i} + \tau q_2(y_i)\sqrt{\Delta y_i}.$$

Algorithm 3 Geodesics

```
Require: $q_1, q_2$: SRVT's of $c_1$, $c_2$,
         $(x_k, y_k)_k$: Point estimates $(x(t_k), y(t_k)) = (x_k, y_k)$,
         $\tau$: Geodesic time.
Ensure: $(c_k)_k$: Point estimates $c_k = c(\tau)(t_k)$.

procedure GEODESICS($q_1$, $q_2$, $\tau$, $(x_k)$, $(y_k)$)
    $c_0 \leftarrow 0$
    for $k = 0, \dots, K$ do
        $\Delta x_k \leftarrow x_{k+1} - x_k$
        $\Delta y_k \leftarrow y_{k+1} - y_k$
        $(q\sqrt{\Delta t})_k \leftarrow (1-\tau) q_1(x_{k+1})\sqrt{\Delta x_k} + \tau q_2(y_{k+1})\sqrt{\Delta y_k}$
        $c_{k+1} \leftarrow c_k + (q\sqrt{\Delta t})_k |(q\sqrt{\Delta t})_k|$
    end for
end procedure
```

An outline of the algorithm computing the geodesics is given in Algorithm 3.

If we are working with curves of equal length, however, it might be more suitable to consider length preserving geodesics. Recall that the set of unit length curves corresponds to the unit sphere of SRVTs. Further, geodesics on the unit sphere are given by

$$q(\tau) = \frac{\sin(\theta(1-\tau))}{\sin(\theta)} q_1 + \frac{\sin(\theta\tau)}{\sin(\theta)} q_2$$

where $\theta = \arccos(\langle q_1, q_2 \rangle)$, which for optimally reparametrized curves means that $\theta = \arccos(s(c_1, c_2))$. In general, any geodesics which is linear in q_1 and q_2 can be approximated in the same way. For any geodesic of the form $q(\tau) = f_1(1-\tau)q_1 + f_2(\tau)q_2$, the same method applies, with altered definition of q given by

$$\left(q\sqrt{\Delta t}\right)_i := f_1(1-\tau) q_1(x_i)\sqrt{\Delta x_i} + f_2(\tau) q_2(y_i)\sqrt{\Delta y_i}.$$

3.9 Richardson Extrapolation

We have seen that if the variable S is continuously differentiable, the update equation for S has $\mathcal{O}(h^2)$ truncation error. It is only due to the summation over such errors that we get an $\mathcal{O}(h)$ error in total. However, through extrapolation, we can achieve a quadratic convergence rate. Let $S^{(h)}$ be the approximation of S with average cell size h. We assume that

$$S^{(h)} = S + k_1 h + k_2 h^2 + o(h^2).$$

If this holds, we can use the following extrapolation:

$$2S^{(h/2)} - S^{(h)} = S - \tfrac{1}{2} k_1 h^2 + o(h^2).$$

This technique is known as Richardson Extrapolation. Note that if the extrapolation works, this will affect the convergence rate of x, y and c as well. A bilinear approximation is an $\mathcal{O}(h^2)$ approximation method, which means that the bilinear approximation to the extrapolated S, will be $\mathcal{O}(h^2)$. Therefore $\sqrt{x}$, $\sqrt{y}$ and hence also x, y and c will have the same error.

If we have a greater Taylor series, i.e. if for example $S^{(h)} = S + k_1 h + k_2 h^2 + k_3 h^2 + o(h^3)$, we can apply several steps of extrapolation which could yield a better convergence rate for S. However, we expect this to not work for x, y and c. This is because the quadratic error we obtain when using the bilinear approximation to S dominates the possibly higher convergence rate of S at the grid points.

The estimate of total similarity s, as given in equation (3.9), already has a quadratic error. Therefore, to increase the convergence rate, the extrapolation has to be done slightly differently. If we assume that

$$s^{(h)} = s + k_2 h^2 + k_3 h^3 + o(h^3),$$

we can use the extrapolation

$$\frac{4s^{(h/2)} - s^{(h)}}{3} = s - \tfrac{1}{6} k_1 h^3 + o(h^3).$$

However, we do not have results on the Taylor expansion of s, and we cannot determine whether this extrapolation will work or not. We know that the extrapolation will be $o(h^2)$, but this need not be significantly better than quadratic convergence.

For the extrapolation to work, we need to have an even finer grid at the boundaries. For one step of extrapolation, we need the grid transformations to go as $g_1(x) \sim x^4$ and $g_2(y) \sim y^4$ for small x and y. Further, to use two steps of extrapolation, they must go as x^6 and y^6 etc. To ensure that the grid transformations are not the limiting factor for applying extrapolation, we suggest using smooth grid transforms which satisfy $d^k/dx^k g_1(0) = d^k/dx^k g_2(0) = 0$ for all $0 \leq k < \infty$. One such example is

$$g_1(t) = g_2(t) = \exp\left\{1 - \frac{1}{t^2}\right\}.$$

4 | Numerical Experiments

The method derived in this thesis is consistent, however under assumptions which not necessarily hold. Therefore, numerical experiments are needed to test whether we have the convergence rates we expect. To do so, we will both study simple problems where analytical solutions are available, and more complicated problems to support the theoretical convergence results. To evaluate the convergence, we need a to construct error estimates for the approximations. Notation wise, we will for any variable, say x, use $\hat{x}^{(h)}$ denote the approximation to x using an average grid cell size h.

Since the total similarity is just a real number, we define the error as $e_s(h) := |\hat{s}^{(h)} - s|$. For the approximation to S, it is natural to use the pointwise max-norm over the grid, and define the error as

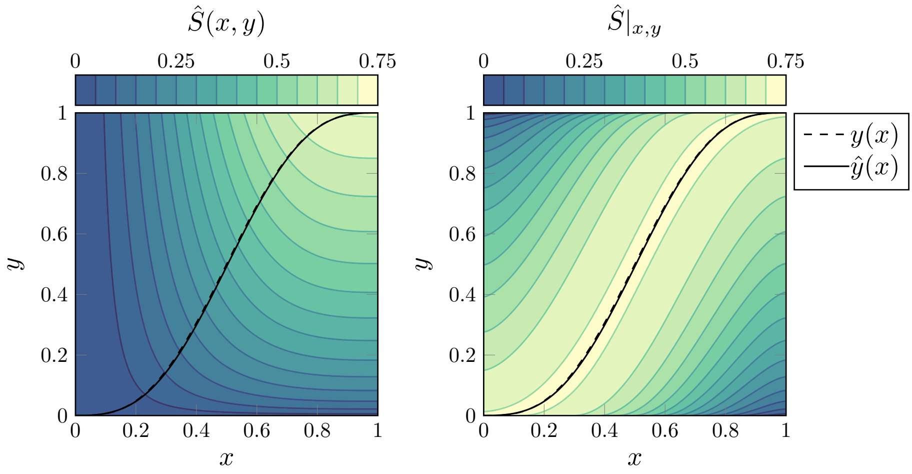
$$e_S(h) := \max_{i,j \in 0,\ldots n} \left|\hat{S}^{(h)}_{i,j} - S(g_1(i/n), g_2(j/n))\right|.$$

If S is smooth, this error estimate is a consistent approximation to the max-norm $\|S^{(h)} - S\|_\infty$ over the unit square. For the error of the optimal path and the geodesics, there are different natural choices for the error based on whether analytic solutions are available or not.

In all the following experiments, we used the second update equation (3.6). From experience, there seems to be next to no difference in accuracy between the update equations. However, the second update equation seems to be slightly less computationally intensive, hence a better choice. We used grid transformations $g_1(t) = g_2(t) = \exp\{1 - 1/t^2\}$ as suggested in Section 3.9. The experiments were implemented in Python, parallelized using NumPy, and ran on a 2.7 GHz Intel Core i5 processor with 8 GB of memory. The processor has two cores, but effectively, four threads can be ran in parallel.

4.1 Line and Circle

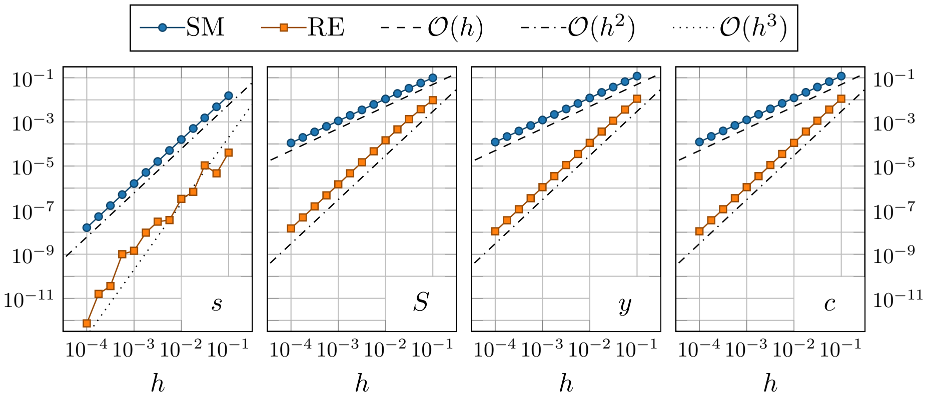
Problems where analytic solutions are available are of especial interest, as global convergence can be determined. One example where we have an analytic solution is when the curves are given by a line and a half circle, oriented such that they



(a)

(b)

Figure 4.1: Shadings of S for experiment 4.1. Approximations to $S(x, y)$ and $S|_{x,y}$ are shaded in subfigure (a) and (b), respectively, and the theoretical and approximated alignment path are drawn. Here, $n = 100$ discretisation points were used.



(a)

(b)

(c)

(d)

Figure 4.2: Convergence plots for experiment 4.1. Convergence for both the standard method (SM) and Richardson extrapolation (RE) are plotted, and the variable of convergence is denoted in the bottom right corner of each subfigure.

are always positively correlated. Here, we can define the SRVTs of these curves by $q_1(t) = [\cos(\pi t), \sin(\pi t)]^T$ and $q_2(t) = [0, 1]^T$. Derivations for the similarity, optimal reparametrization path and geodesics can be found in appendix A.1 and the curves in question are drawn in Figure 4.3.

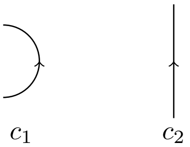


Figure 4.3: Curves analysed in experiment 4.1.

For this problem, analytic solutions are available if we only reparametrize the second curve. Therefore, we measure the error of the reparametrizer $y(x)$ by the point-wise maximum error. Similarly, for the geodesics, we measure the error by the maximum pointwise euclidian distance. These error estimates can be approximated by

$$e_y(h) := \max_{k \in 0, \dots K^{(h)}} \left| \hat{y}_k^{(h)} - y(x_k) \right|,$$
$$e_c(h) := \max_{k \in 0, \dots K^{(h)}, l \in 0, \dots n} \left| \hat{c}_{l,k}^{(h)} - c(\tau_l)(x_k^{(h)}) \right|.$$

Here, we are maximizing over the parametric times x_k generated by the backtracking algorithm, and over a discrete set of geodesic times $\tau_l = l/n$ for $l = 0, \dots, n$.

The algorithms presented in this thesis were ran on c_1 and c_2 with $n = 100$ discretisation points in each direction. The similarity estimate was $S(1, 1) = 0.7045$ which is close to the theoretical value of $S(1, 1) = 1/\sqrt{2} \approx 0.7071$. Further, the approximated and theoretical optimal reparametriser and approximations of the cumulative similarity $S(x, y)$ and restricted similarity $S|_{x,y}$ are visualised in Figure 4.1. As one can see, the estimated reparametriser is almost identical to the theoretical. Additionally, one can see that the reparametriser is located at a maximum of the restricted similarity $S|_{x,y}$ as expected.

Both the standard algorithm, and the algorithm with one step of Richardson extrapolation were ran with different number of discretization points, and the resulting convergence plots are visualized in Figure 4.2. It is clear that the standard method converges linearly for the cumulative similarity S, the optimal reparametrizer $y(x)$ and the geodesics c. For the same variables, one step of Richardson extrapolation gives a quadratic convergence rate. This is in line with what we expect since this problem do not have shocks in S.

For the total similarity $s(c_1, c_2)$ the standard method gives a quadratic convergence. Again, this is in line with what we expect. We also applied Richardson extrapolation to the estimates of $s(c_1, c_2)$. It is clear that the extrapolation yields a much better estimate. The actual convergence rate, is hard to determine, but it is likely close to cubic, and at least superquadratic. The sporadic behavior of the convergence, however, might indicate that the improved convergence rate is more or less coincidental, and just a special case for this problem.

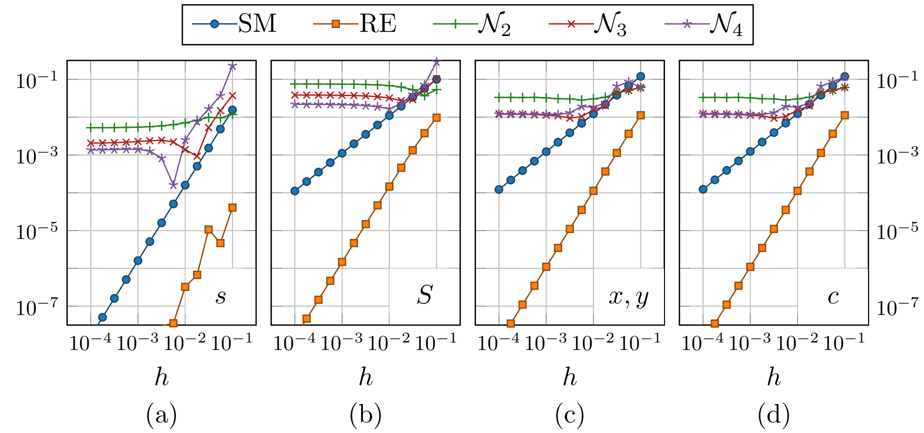


Figure 4.4: Convergence plots for experiment 4.2. Convergence for the standard method (SM), Richardson extrapolation (RE), and the previous method using three different neighbourhoods are plotted, and the variable of convergence is denoted in the bottom right corner of each subfigure.

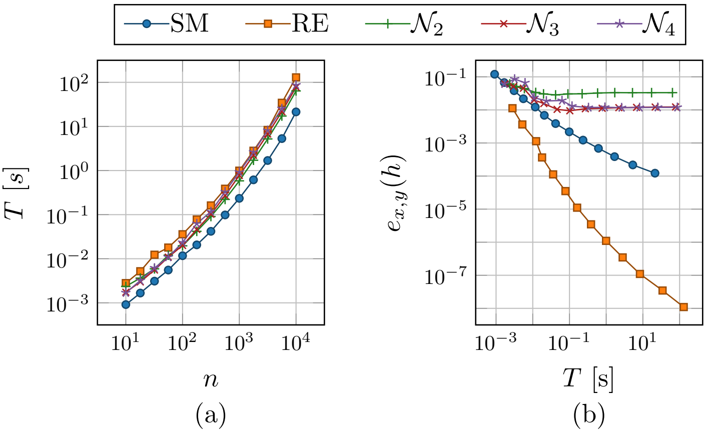


Figure 4.5: Running times for estimating S in experiment 4.2. In (a), the running times is plotted against the number of discretization points n, and in (b), the the error of the optimal path x, y is plotted against the running time.

4.2 Comparison of Neighbourhoods

A major benchmark for our method is comparing the performance of the method with previous dynamic programming based methods. To do so, we will consider fixed sets of neighbouring nodes on the form

$$\begin{aligned}\mathcal{N}_m(i,j) =&\{(k,l) \mid i-m \leq k < i,\;\; j-m \leq l < j,\;\; \gcd(i-k, j-l) = 1\} \\ &\cup \{(i-1,j)\} \cup \{(i,j-1)\}.\end{aligned}$$

The constraint $\gcd(i-k, j-l) = 1$ is included to ensure that a grid connection does not pass over another grid point. For example, we do not want to allow $(0,0)$ to connect with $(2,2)$ since this can be done in two steps: $(0,0) \to (1,1)$ and $(1,1) \to (2,2)$. We will also allow the path to be vertical or horizontal, to be consistent with the new approach. These types of sets for $m = 2,3,4$ are visualized in fig. 4.6. We approximate the update equation for the previous method using a one point right Riemann sum, given by

$$S(x_i, y_j) = \max_{(k,l)\in\mathcal{N}_m(x_i,y_j)} S(x_k, y_l) + \langle q_1(x_i), q_2(y_j)\rangle \sqrt{(x_i - x_k)(y_j - y_l)}.$$

There are more precise methods to approximate the update equation. However, using the above update equation, parallelization is especially efficient and easy to implement. Further, with fixed neighbourhoods, the maximum step size of the path is $\mathcal{O}(h)$, meaning that the error from the one point right Riemann sum will vanish as $h \to 0$.

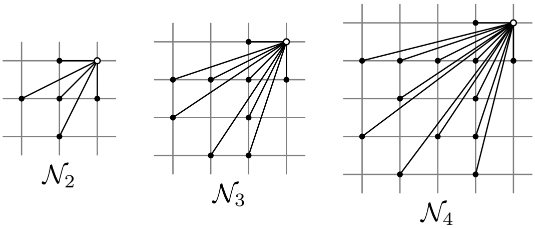
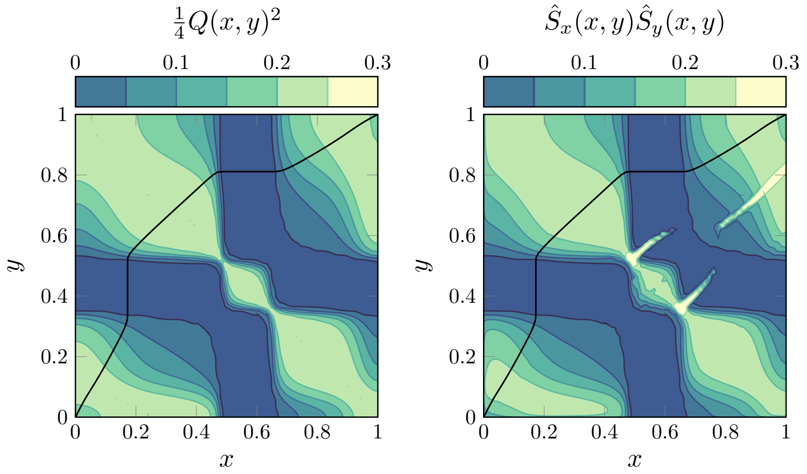


Figure 4.6: Examples of sets of neighbouring nodes.

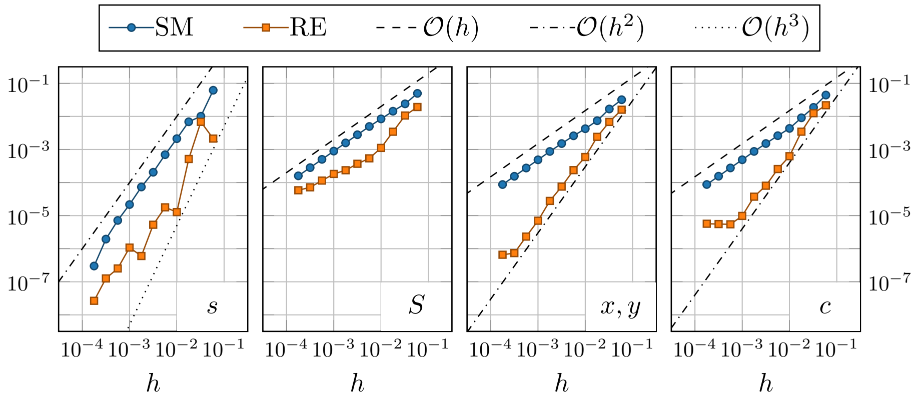
We revisit the line – half circle problem, and the convergence results are visualized in fig. 4.4. As one can see, the new approach outperforms the previous method for all variables. Even though the previous method gives a decent estimate of the total similarity $s(c_1, c_2)$, the optimal path x, y and the geodesics are not well approximated. The estimate of S is even worse. However, the previous method was not constructed particularly to estimate $S(x,y)$ for all x, y. Hence, it might be an unfair comparison. Observe that the error for previous method with fixed sets of neighbouring nodes does not converge. This is as expected since we for fixed sets of neighbouring nodes do not allow the path to attain all possible slopes.

The running times for this experiment are visualized in fig. 4.5a. For small n, the running times seem to grow linearly with n, while for larger n, the running



(a) (b)

Figure 4.7: Shadings of the PDE (2.14) for experiment 4.3. The exact value of $\frac{1}{4}Q^2$ are shaded in (a), and approximations to S_xS_y are shaded in (b). Approximated optimal alignment path is drawn. Here, $n = 100$ discretization points were used.



(a) (b) (c) (d)

Figure 4.8: Convergence plots for experiment 4.3. Convergence plots for both the standard method (SM) and Richardson extrapolation (RE) are plotted, and the variable of convergence is denoted in the bottom right corner of each subfigure.

times grows quadratically. This is exactly what we expect. When n is small, the effect of parallelization is seen as a linear computational complexity, while for large n, the effect is no longer apparent. We emphasise that these results should be seen as a proof of concept, as the implementation is not optimized. In particular, the running times are highly dependent on NumPy's parallelization procedures and memory handling. It is also interesting to plot the error of the method against the running time. Since the approximation of the optimal reparametrization path is the most important method in this thesis, the error of $y(x)$ is plotted against the running time in fig. 4.5b. As one can see, the extrapolation yields a better approximation than all other methods for equal running times.

4.3 Presence of Shock Solutions

In this experiment, we want to study how the appearance of shock solutions affect the convergence rates. To do so, we constructed two natural cubic splines (which are C^2-continuous), as visualized in Figure 4.9. From experience, shock solutions seem to appear whenever the two curves are oppositely curved, which is the case in this experiment.

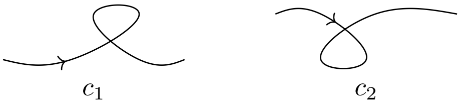


Figure 4.9: Curves analysed in experiment 4.3.

To start, recall that we expect the nonlinear partial differential equation $S_xS_y = \frac{1}{4}Q^2$ to hold. In Figure 4.7, shadings of the left and right hand side of this PDE are visualized together with the optimal reparametrization path for this problem. First of all, observe that the values of the right and left hand size seems be equal almost everywhere, indicating that the PDE holds. There seems to be two cases where the PDE is not satisfied:

(i) At the x- and y-axes. This is as expected. Consider the x-axis where we have that $S_x = 0$ and $S_y = +\infty$ (wherever $\langle q_1(x), q_2(0)\rangle > 0$). The infinite directional derivative cannot be well approximated numerically, which is why we see a discrepancy between the left and right hand side).

(ii) Three distinct paths where $\hat{S}_x\hat{S}_y$ is much larger than $\frac{1}{4}Q^2$. These are the shock paths we expect. At the paths, S is not differentiable, which explains why the numerical estimates of the differential equation does not hold. Note that we expect the theoretical shock paths to be unit dimensional, and it is due to the discretization that they are thicker.

In this experiment, we do not have analytic solutions to the optimization problem, and we must therefore approximate the convergence rates. We will also reformulate the error for the optimal path and the geodesics. For the optimal path, recall that the constraint $\dot{x} + \dot{y} = 2$ ensures that the path is Lipchitz continuous. We are now considering reparametrization paths of the form $x(t), y(t)$ where $t = \frac{1}{2}(x + y)$. This can be utilized to construct a consistent approximations to the optimal path and the geodesics on the form

$$\begin{aligned} e_{x,y}(h) &:= \max_{k \in 0,\ldots K^{(h)}} \left|\hat{y}_k^{(h)} - y(t_k)\right|, \\ e_c(h) &:= \max_{k \in 0,\ldots K^{(h)},\, l \in 0,\ldots n} \left|\hat{c}_{l,k}^{(h)} - c(\tau_l)(t_k^{(h)})\right|. \end{aligned}$$

Note that we here define the error of the optimal path using only y. In fact, the error for x and y will be always the same since

$$\left|\hat{y}_k^{(h)} - y(t_k)\right| = \left|\widehat{(}2t_k - x_k^{(h)}) - (2t_k - x(t_k))\right| = \left|\hat{x}_k^{(h)} - x(t_k)\right|.$$

Still, we do not have analytic solutions to the optimization problems. Therefore, we used a very fine grid of $n = 10^4$ and the extrapolation method to approximate “theoretical” solutions for s, S, x, y and c.

Convergence plots for this experiments are visualised in Figure 4.8. For the standard method, the convergence rates are exactly as expected. Even though we have shocks appearing, they are only on a unit-dimensional subset of the unit square, which means that the linear convergence rates for S, x, y and c are maintained. Additionally, we have a quadratic convergence rate for s. When we apply Richardson extrapolation, we do not gain any additional convergence rate in S. This is as expected since S is now only piecewise C^2, meaning that the max-error for the extrapolation will still have a linear convergence.

For the convergence of x, y and c, the situation is slightly different. For course grids, the convergence seems to be quadratic when using extrapolation. This indicates that that the Richardson extrapolation works for S almost everywhere. However, when the grid gets fine enough, the convergence curve flattens out. This can be understood from looking at the L^1-error of S, which we approximate by

$$e_{S,L^1}(h) := \sum_{i,j=0}^{n-1} |S_{i,j}^{(h)} - S(x_i, y_j)|\Delta x_i \Delta y_j.$$

Convergence of the L^1-norm is visualized in Figure 4.10. One can see that the L^1-error converges quadratically when using extrapolation, at least for course grids. In other words, the shock solutions seems to make extrapolation only locally unavailable. However, the error obtained from the shock solutions does not seem to propagate, which is why the extrapolation is available for almost all points (x, y). Using extrapolation, the convergence rate for $s(c_1, c_2)$ might be slightly better than the standard method. However, it should be clear that the method does not have cubic convergence.

Lastly, observe that any path must pass through a region with negative correlation. And from Figure 4.7a, one can see that the optimal path is either vertical or horizontal when passing through such regions. This supports the result on the characterisation of optimal paths we obtained in Theorem 2.4.4.

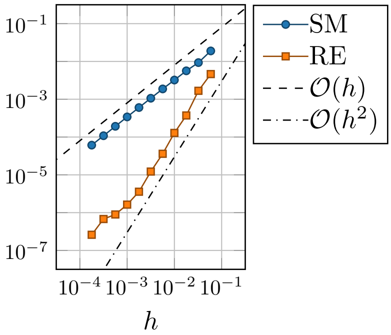


Figure 4.10: Convergence of the L^1 error of S for experiment 4.3.

4.4 Almost C^2-continuous Curves

To stress test our method, we consider curves which are piecewise C^2-continuous with a strict C^1-condition. Specifically, we consider composite Bézier curves representing the chess pieces pawn and queen, as seen in Figure 4.11. The length of the pawn curve is normalized to $L(c_1) = 1$, while the queen has length $L(c_2) = 1.69$. The curves are in this case arc length parametrized. The geodesics for the right halves of the curves are visualized in Figure 4.12. As one can see, the geodesics registers part of the curves which we expect to be registered.

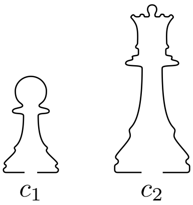


Figure 4.11: Curves analysed in experiment 4.4.

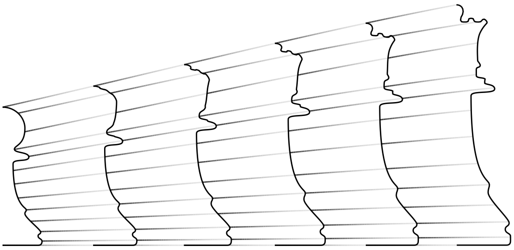
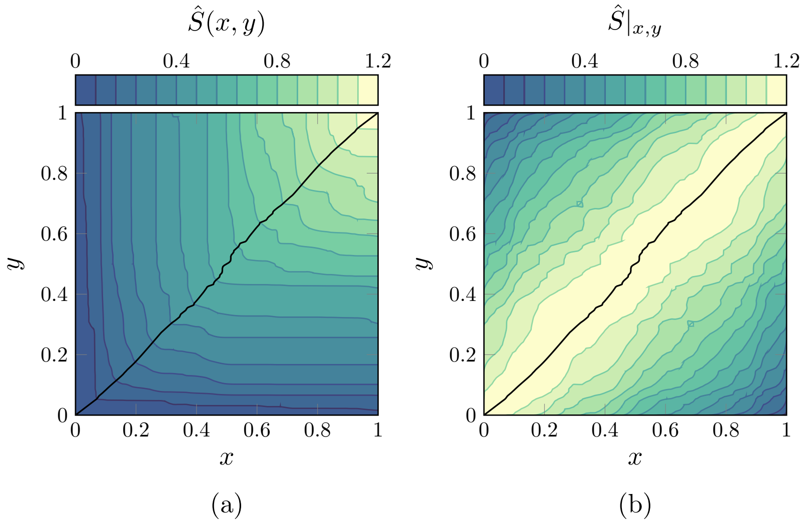
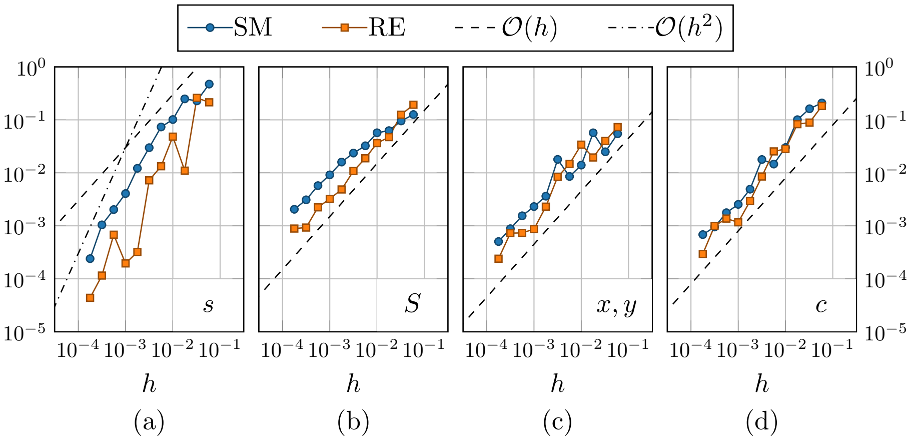


Figure 4.12: Geodesics between halves of the chess pieces pawn and queen. Both $c(\cdot, t_i)$ and $c(\tau_i, \cdot)$ are drawn for selected t_i and τ_i. Due to symmetry, the second halves look identical.



(a) (b)

Figure 4.13: Shadings of S for experiment 4.4. Approximations to $S(x,y)$ and $S|_{x,y}$ are shaded in subfigure (a) and (b), respectively, and the approximated alignment path are drawn. Here, $n = 2000$ discretisation points were used.



(a) (b) (c) (d)

Figure 4.14: Convergence plots for experiment 4.4. Convergence for both the standard method (SM) and Richardson extrapolation (RE) are plotted, and the variable of convergence is denoted in the bottom right corner of each subfigure.

As for experiment 4.1, we shaded the cumulative and restricted similarity which can be seen in Figure 4.13 together with the approximated optimal path. Again, the optimal path is located a the maximum ridge of the restricted similarity $S|_{x,y}$. Observe that the optimal path is almost linear for approximately the first and last third of the unit interval. This fits well with the shapes of the curves in question. The bottom part of the curves (i.e. for $t \lessapprox 1/3$ and $t \gtrapprox 2/3$) are almost identical up to a scaling factor. This explains why the path is almost linear.

As for the previous experiment, we estimate the theoretical values of the variables in question using $n = 10^4$ discretization points. Then, the methods were ran on the curves with different number of discretization points, and the resulting convergence plots are visualized in Figure 4.14. First of all, observe that the method converges for all variables of interest. The standard method seems to converge almost linearly for both S, x, y and c. This indicates that the method can handle problems with less regular curves. It is, however, not surprising that the extrapolation does not work for this problem, as we do not have a uniform first order Taylor expansion of q_1 and q_2. Lastly, observe that the estimate for the total similarity s converges superlinearly but not quite quadratic. This fits well with the convergence of x, y which seems to be almost linear.

5 | Conclusion and Future Work

In this thesis, we have constructed a new method for solving the reparametrization problem within the square root velocity framework. The method builds upon the idea of an existing dynamic programming approach, but while the previous method is fully discretized, the new method is only semi-discretized. This is utilized to give both a better convergence rate and a lower computational complexity.

The method is based on new theoretical insight into the problem. We have expanded the theory characterizing the optimal reparametrization paths, and introduced a new auxiliary variable, S, which allows us to construct a differential equation describing the evolution of the optimal path x, y. A dynamic programming method is constructed to approximate S, and we have shown that consistent approximations to S give consistent approximations to x, y. In addition, we have shown how these approximations can be used to obtain consistent approximations to the reparametrized curves and the geodesics, and that extrapolation can be utilized to increase the order of convergence in certain cases. Numerical experiments have demonstrated that the method converges, and the numerical order of convergence support the expected convergence rates. Experiments also indicate that the method works even if the regularity assumptions of the curves and the auxiliary variable S are not strictly met.

Although the numerical experiments demonstrate convergence, more work is needed on the theoretical aspects of the method in order to rigorously prove theoretical convergence. In addition, we constructed the method as a special case of a more general framework which might be applicable to a greater set of problems. We will therefore conclude this thesis with the following research proposals:

Study of the Differential Properties. The method derived in this thesis relies on the assumption that S is piecewise C^2-continuous. Although numerical experiments indicate that we have sufficient regularity for convergence, theoretical regularity properties of S need to be established.

We believe that the method can be adapted to handle the shock solutions in S. The dynamic programming framework builds upon analytic solutions to the base cases of the optimization problem, where we assume that S is piecewise linear on some boundary and that the curves c_1 and c_2 are linear. Further, we believe that piecewise linear approximations to S can be established in these base cases, which means that the method could handle shock solutions. This is particularly desired, as extrapolation would then work in all situations.

There is also work to be done concerning the differential properties of x, y. We claim that the method gives consistent approximations to $\sqrt{\dot{x}}$ and $\sqrt{\dot{y}}$. This, however, needs a rigorous proof.

Adaptation to Closed Curves. We have derived a method for computing optimal reparametrization of open curves. As many applications consider closed curves, it is natural to ask whether our method can be adapted to work for closed curves as well.

Applications to Other Variational Problems. Although we constructed the dynamic programming method to solve the reparametrization problem using the square root velocity transform, the general framework of the method is not limited to this problem. We only require a problem of finding a monotone function with fixed start and end points. Such problems include any reparametrization problem for curves and the problem concerned with alignment of cumulative distribution functions.

Bibliography

[1] Laurent Younes. Computable elastic distances between shapes. *SIAM Journal of Applied Mathematics*, 58:565–586, 1998.

[2] Washington Mio and Anuj Srivastava. Elastic-string models for representation and analysis of planar shapes. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 10–15, 2004.

[3] Washington Mio, Anuj Srivastava, and Shantanu Joshi. On shape of plane elastic curves. *International Journal of Computer Vision*, 73:307–324, 2007.

[4] Anuj Srivastava, Eric Klassen, Shantanu Joshi, and Ian Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, 2011.

[5] Martin Bauer, Markus Eslitzbichler, and Markus Grasmair. Landmark-guided elastic shape analysis of human character motions. *Inverse Problems and Imaging*, 11(4):601–621, 2017.

[6] Martin Bauer, Martins Bruveris, Philipp Harms, and Peter W. Michor. Vanishing geodesic distance for the riemannian metric with geodesic equation the kdv-equation. *Annals of Global Analysis and Geometry*, 41(4):461–472, 2012.

[7] Peter Michor and David Mumford. Vanishing geodesic distance on spaces of submanifolds and diffeomorphisms. *Documenta Mathematica*, 10:2017–245, 2005.

[8] Peter Michor and David Mumford. Riemannian geometries on spaces of plane curves. *Journal of the European Mathematical Society*, 8(1):1–48, 2006.

[9] Martins Bruveris. Optimal reparametrizations in the square root velocity framework. *SIAM Journal on Mathematical Analysis*, 48(6):4335–4354, 2016.

[10] Martins Bruveris, Peter Michor, and David Mumford. Geodesic completeness for sobolev metrics on the space of immersed plane curves. *Forum of Mathematics, Sigma*, 2:e19, 2014.

[11] Sebastian Kurtek, Anuj Srivastava, Eric Klassen, and Hamid Laga. Landmark-guided elastic shape analysis of spherically-parameterized surfaces. *Computer Graphics Forum*, 32:429–43, 2013.

[12] Thomas Sebastian, Philip N. Klein, and Benjamin B. Kimia. On aligning curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:116–125, 02 2003.

[13] Günay Doğan, Javier Bernal, and Charles R. Hagwood. A fast algorithm for elastic shape distances between closed planar curves. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4222–4230, 2015.

[14] Günay Doğan, Javier Bernal, and Charles R. Hagwood. Fast dynamic programming for elastic registration of curves. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1066–1073, 2016.

A | Analytic Solutions

A.1 Line and Circle

Consider the unit arc-length curves defined as a half circle and a line. The SRVT's of these curves can be expressed as

$$q_1(t) = \begin{bmatrix} \cos(\pi t) \\ \sin(\pi t) \end{bmatrix} \text{ and } q_2(t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

and the restricted similarity is given by

$$S(x,y) = \sup_{\varphi \in \mathrm{Diff}([0,1])} \sqrt{xy} \int_0^1 \sin(\pi x t)\sqrt{\dot{\varphi}(t)}dt.$$

To ensure that $\varphi(1) = 1$, we must have that $\sqrt{\dot{\varphi}} \in S_1$, where S_1 denotes the unit sphere. Further, since the projection of any element onto the unit sphere is given by a scalar multiplication, the supremum is attained at $\sqrt{\dot{\varphi}(t)} = C\sin(\pi x t)$ for some constant C. It is important to note that this is a valid projection since $\sin(\pi x t) \geq 0$ for all $x, t \in [0,1]$. This gives

$$\varphi(t) = \frac{C^2}{2}\left(t + \frac{\sin(2\pi x t)}{2\pi x}\right).$$

To ensure $\varphi(1) = 1$, we must require that $C^2/2 = (1 + \mathrm{sinc}(2x))^{-1}$, where we define $\mathrm{sinc}(x) := \sin(\pi x)/(\pi x)$. In particular, for $x = y = 1$, the constant collapses to $C = \sqrt{2}$ and optimal reparametrisation becomes

$$\varphi(t) = t + \frac{\sin(2\pi t)}{2\pi}.$$

Moreover, the restricted similarity can now be computed as

$$S(x,y) = \sqrt{\frac{2xy}{1 + \mathrm{sinc}(2x)}} \int_0^1 \sin^2(\pi x t)dt = \sqrt{\frac{1}{2}xy(1 + \mathrm{sinc}(2x))}.$$

In particular, we have that $s(c_1, c_2) := S(1,1) = 1/\sqrt{2}$. We consider arc-length preserving geodesics, which means that the geodesic distance is given by the "angle" between the SRVTs, given by $\theta = \arccos(s(c_1, c_2)) = \pi/4$. Therefore, the geodesics are defined from

$$q(\tau)(t) = \sqrt{2}\sin\big(\tfrac{\pi}{4}(1-\tau)\big)\begin{bmatrix}\cos(\pi t)\\ \sin(\pi t)\end{bmatrix} + 2\sin\big(\tfrac{\pi}{4}\tau\big)\begin{bmatrix}0\\ \sin(\pi t)\end{bmatrix} = \begin{bmatrix}a\cos(\pi t)\\ (a+b)\sin(\pi t)\end{bmatrix}$$

where we have introduced $a = \sqrt{2}\sin\big(\frac{\pi}{4}(1-\tau)\big)$ and $b = 2\sin\big(\frac{\pi}{4}\tau\big)$. Using this notation, the squared absolute value of the geodesics can be computed by

$$\begin{aligned}|q(\tau)|^2 &= a^2 + (2ab + b^2)\sin^2(\pi t)\\ &= (a+b)^2 - (2ab+b^2)\cos^2(\pi t).\end{aligned}$$

Recall that given q, the inverse square root velocity transform is given by $R^{-1}(q) = \int q|q|dt$ up to a translation. Writing the interpolated curve on the form $c(\tau)(t) = [x(\tau)(t), y(\tau)(t)]^T$, the components can now be expressed as

$$\begin{aligned}x(\tau)(t) &= a\int \cos(\pi t)\sqrt{a^2 + (2ab+b^2)\sin^2(\pi t)}dt\\ &= -\frac{a}{2\pi}\Bigg(\sin(\pi t)\sqrt{a^2+(2ab+b^2)\sin^2(\pi t)}\\ &\qquad + \frac{a^2}{\sqrt{2ab+b^2}}\operatorname{arcsinh}\left(\frac{\sqrt{2ab+b^2}}{a}\sin(\pi t)\right)\Bigg),\\ y(\tau)(t) &= (a+b)\int \sin(\pi t)\sqrt{(a+b)^2 - (2ab+b^2)\cos^2(\pi t)}dt\\ &= \frac{a+b}{2\pi}\Bigg(\cos(\pi t)\sqrt{(a+b)^2-(2ab+b^2)\cos^2(\pi t)}\\ &\qquad + \frac{(a+b)^2}{\sqrt{2ab+b^2}}\arcsin\left(\frac{\sqrt{2ab+b^2}}{a+b}\cos(\pi t)\right)\Bigg).\end{aligned}$$