A PDE-based Method for Shape Registration

Esten Nicolai Wøien[*][†]   Markus Grasmair[*][‡]

March, 2021

Abstract

In the square root velocity framework, the computation of shape space distances and the registration of curves requires solution of a non-convex variational problem. In this paper, we present a new PDE-based method for solving this problem numerically. The method is constructed from numerical approximation of the Hamilton-Jacobi-Bellman equation for the variational problem, and has quadratic complexity and global convergence for the distance estimate. In conjunction, we propose a backtracking scheme for approximating solutions of the registration problem, which additionally can be used to compute shape space geodesics. The methods have linear numerical convergence, and improved efficiency compared previous global solvers.

1   Introduction

A large number of applications require the manipulation and mathematical or statistical analysis of geometric objects in general and curves in particular. Examples from mathematical image processing are segmentation, where one wants to find and classify different objects within an image based on their outlines (see e.g. [14] for a classical model), or object tracking (see [19]), where one wants to follow the same object over a sequence of consecutive frames. Other examples include the analysis of shapes of proteins [15], modelling and analysis of computer animations [3, 9], or also inverse problems concerning the detection of shapes from indirect measurements [12].

In order to perform these tasks, it is necessary to have a well-defined and easily computable notion of distance between curves at hand. One important example is the *Square Root Velocity* (SRV) distance originally introduced in [17, 16] (see Section 2 below for a precise definition), which can be interpreted as a measure for the bending and stretching energy that is required for transforming one curve into another. For parametrised curves, this distance is defined by applying first a non-linear transformation—the *Square Root Velocity transform* (SRVT)—to the involved curves, which maps them onto the unit sphere in L^2. Then, the distance of the curves is defined as the unit sphere distance of their

---

[*]Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

[†]`esten.n.woien@ntnu.no`

[‡]`markus.grasmair@ntnu.no`

SRV transformations. Even more, this setting makes it possible to regard the space of all parametrised curves as a manifold with a Riemannian structure that is inherited from the unit sphere in L^2. In particular, one can define geodesics between parametrised curves, that is, optimal deformations of one curve into another.

However, in many applications we are only interested in the image of a curve, but not the concrete parametrisation. We thus rather require a distance between *shapes*, that is, equivalence classes of curves modulo reparametrisations. Within the SRV framework, this can be achieved by defining the distance between two shapes as the infimum of the distance between all curves within their equivalence class. Or, given two parametrised curves c_1 and c_2, we define the distance between their shapes as the infimum of the distance between all reparametrisations of c_1 and c_2. It can be shown that this infimum is positive for all distinct shapes and thus defines a distance on the set of all shapes. Moreover, it is again possible to view the space of all shapes as a Riemannian manifold and thus to define geodesic between shapes. We refer to [18] for a detailed introduction into shape analysis within the SRV framework; a short overview can also be found in [2].

The actual computation of the shape distance and of geodesics, though, requires the solution of an optimisation problem over the space of all reparametrisations. This problem has the form

$$\inf_{\varphi_1, \varphi_2} \int_I F(\varphi_1(t), \varphi_2(t), \varphi_1'(t), \varphi_2'(t)) \, dt, \tag{1}$$

where the infimum is computed over all orientation preserving reparametrisations of the unit interval $I = [0, 1]$. Here the integrand F depends on the SRVTs of the curves c_1 and c_2 one wants to compare. Due to invariance properties of the SRV distance, it is sufficient to compute the minimum in (1) only with respect to one of the diffeomorphisms, e.g. w.r.t. φ_1 while leaving φ_2 constant equal to Id. This reduces the dimensionality of the problem, but one is still left with an optimisation problem over a function space.

For the numerical solution, there are two main approaches: gradient based methods and dynamical programming. In the dynamic programming approach introduced in [16] (see also [21] for a similar numerical approach for a different shape distance), one approximates the diffeomorphism φ_1 by a piecewise linear approximation with nodal points and nodal values within a fixed partition $0 = t_0 < t_1 < \ldots < t_N = 1$ of the unit interval. The resulting discrete optimisation problem is then solved by a dynamic programming algorithm. Without further modifications, this algorithm has a time complexity of $O(N^4)$ and thus is not useful for practical applications. A significant speed-up is possible, though, by limiting the set of possible slopes for the linear approximations of φ_1. In fact, a method with complexity $O(N^3)$ has already been proposed in [16]. Even more, a variant with complexity $O(N)$ has been presented in [4] (see also [11]), which is an iterative based on an adaptive, local refinement of the search grid for the dynamic program.

Gradient based methods usually work on a finite dimensional approximation of the space of all reparametrisations, e.g. using B-splines or trigonometrical functions. The optimisation problem (1) is then rephrased as a problem for the basis coefficients. This is a finite dimensional optimisation problem, which can,

in principle, be solved with standard methods like gradient descent or quasi-Newton methods. One difficulty is the constraint that the functions φ we are optimising over are orientation preserving diffeomorphisms and thus monotonically increasing. Thus one has a positivity constraint for φ', which is difficult to handle numerically. Thus [13] rephrase the problem in terms of $\gamma^2 := \varphi'$, which yields an optimisation problem with the single equality constraint that $\|\gamma\|_{L^2}^2 = 1$.

We note here, though, that the optimisation problem (1) is typically highly non-convex, as the reparametrisations appear as arguments of the curves or their SRVTs. Thus it is highly likely that there is a large number of local minimisers and other critical point. Indeed, an example of such a situation is shown in our numerical examples in Section 7.1. Since gradient based methods are local, it is therefore necessary to initialise the iteration with a sufficiently good initial guess of the solution. The same holds for the adaptively refined dynamic program suggested in [4]. If the initialisation of that method is too coarse, it can happen that the refinement strategy is never able to find the true global minimum.

In this paper, we want to present, and analyse, an alternative approach to the solution of (1) which is based on a formulation as a continuous dynamic program. This formulation allows us to define a continuous *value function* $u: I \times I \to \mathbb{R}$, where $u(x_1, x_2)$ measures the minimal (partial) cost of a reparametrisation satisfying $\varphi(x_1) = x_2$. In particular, $u(1, 1)$ is precisely the value of the optimisation problem (1). A precise definition of u is given in (11) below. This value function u has been shown in [8] to satisfy, in the viscosity sense, the associated Hamilton–Jacobi–Bellman equation, which is a hyperbolic PDE with boundary values given for $x_1 = 0$ and $x_2 = 0$. Moreover, convergent numerical schemes for the solution of that PDE have been proposed in [7, 8, 20].

The main contribution of this paper is a generalisation of these schemes, which is closer in spirit to the definition of u by means of a dynamical program. In particular, our approach allows it to recover the optimal reparametrisation in a natural way by solving an ODE. Since the value function is defined by means of the dynamic programming principle, the resulting numerical method will be globally convergent. At the same time, the formulation as a PDE allows the numerical schemes to have a time complexity of only $O(N^2)$. In Section 2, we will formulate the necessary mathematical background in shape analysis and the SRVT. The value function and the Hamilton–Jacobi–Bellman equation are introduced in Section 3.1. In Sections 4 and 5, we will first introduce the general numerical framework together with a convergence analysis, and then propose concrete numerical schemes. Then we will discuss the recovery of the optimal reparametrisation from the value function and the construction of geodesics in shape space in Section 6. Finally, we will present numerical experiments in Section 7.

2 Preliminaries

In the following, we will provide a brief introduction into shape analysis using the square-root-velocity-transform. To that end, let $I = [0, 1]$ be the unit interval

and $d \in \mathbb{N}$, and denote by

$$\mathrm{Imm}(I; \mathbb{R}^d) := \left\{ c \in C^1(I; \mathbb{R}^d) : |\dot{c}(t)| > 0 \text{ for all } t \in I \right\}$$

the space of all C^1 immersions of I in $\mathbb{R}^d$. We define the (scaled) *Square-Root-Velocity-Transform* (SRVT)

$$Q \colon \mathrm{Imm}(I; \mathbb{R}^d) \to C(I; \mathbb{R}^d \setminus \{0\})$$

as

$$q(t) = Q(c)(t) := \frac{1}{\sqrt{\mathrm{Length}(c)}} \frac{\dot{c}(t)}{\sqrt{|\dot{c}(t)|}}, \tag{2}$$

where

$$\mathrm{Length}(c) := \int_I |\dot{c}(t)| \, dt$$

denotes the length of the curve c. Noting that

$$\int_I |q_i(t)|^2 \, dt = \int_I \frac{|\dot{c}_i(t)|}{\mathrm{Length}(c_i)} \, dt = 1,$$

we see that, actually, the SRVT maps a curve to an element of the unit sphere in $L^2(I; \mathbb{R}^d)$. The (scaled) *Square-Root-Velocity* (SRV) distance between c_1 and c_2 is now defined as the geodesic distance between q_1 and q_2, that is,

$$\mathrm{dist}(c_1, c_2) := \arccos\left(\int_I \langle q_1(t), q_2(t) \rangle \, dt \right). \tag{3}$$

Obviously, the SRV distance is translation invariant. Moreover, due to the scaling by the square root of the length of the curves, it is easy to see that it is scale invariant, that is,

$$\mathrm{dist}(\lambda_1 c_1, \lambda_2 c_2) = \mathrm{dist}(c_1, c_2)$$

for all λ_1, $\lambda_2 > 0$. Since dist is defined by means of a geodesic distance, it also satisfies the triangle inequality. As a consequence, it follows that dist is a metric on the *pre-shape space* $\mathrm{Imm}(I; \mathbb{R}^d)/G$, where G denotes the group of translations and scalings in $\mathbb{R}^d$.

In fact, one can show that it is possible to regard $\mathrm{Imm}(I; \mathbb{R}^d)/G$ as a Riemannian manifold for which dist is the geodesic distance. This also makes it possible to construct geodesics between curves: Consider c_1, $c_2 \in \mathrm{Imm}(I; \mathbb{R}^d)/G$ with SRVTs $q_1 = Q(c_1)$ and $q_2 = Q(c_2)$. If $q_1(t) \neq -q_2(t)$ for all $t \in I$ then the geodesic between c_1 and c_2 is given as

$$\tau \mapsto c_\tau := Q^{-1}\big(w(1 - \tau)q_1 + w(\tau)q_2 \big) \in \mathrm{Imm}(I; \mathbb{R}^d)/G. \tag{4}$$

Here Q^{-1} is the inverse SRVT, which can be explicitly computed as

$$Q^{-1}(q)(t) = \int_0^t |q(t')| q(t') \, dt',$$

and

$$w(\tau) = \frac{\sin\big(\tau \, \mathrm{dist}(c_1, c_2)\big)}{\sin\big(\mathrm{dist}(c_1, c_2)\big)}.$$

We note that (4) still makes sense if $q_1(t) = -q_2(t)$ for some (though not all) $t \in I$. In that case, however, the resulting curves c_τ will not all be immersions.

Next we define the *shape space*

$$\mathcal{S}(I; \mathbb{R}^d) := \mathrm{Imm}(I; \mathbb{R}^d)/(\mathrm{Diff}_+(I) \times G),$$

where

$$\mathrm{Diff}_+(I) = \left\{ \varphi \in C^\infty(I) : \varphi(0) = 0, \ \varphi(1) = 1, \ \varphi'(t) > 0 \text{ for all } t \in I \right\}$$

is the group of orientation preserving diffeomorphisms of I. Given two shapes $[c_1]$, $[c_2] \in \mathcal{S}(I; \mathbb{R}^d)$, we then define their distance as

$$\mathrm{dist}^{\mathcal{S}}([c_1], [c_2]) := \inf_{\varphi_1, \varphi_2 \in \mathrm{Diff}_+(I)} \mathrm{dist}(c_1 \circ \varphi_1, c_2 \circ \varphi_2). \tag{5}$$

In fact, it is possible to simplify this expression, as the SRV distance is invariant under simultaneous reparametrisations in the sense that

$$\mathrm{dist}(c_1, c_2) = \mathrm{dist}(c_1 \circ \varphi, c_2 \circ \varphi) \qquad \text{for all } \varphi \in \mathrm{Diff}_+(I).$$

Thus we have that

$$\mathrm{dist}^{\mathcal{S}}([c_1], [c_2]) = \inf_{\varphi \in \mathrm{Diff}_+(I)} \mathrm{dist}(c_1 \circ \varphi, c_2). \tag{6}$$

Again, one can show that this distance is induced by a Riemannian metric on the shape space $\mathcal{S}(I; \mathbb{R}^d)$, which in turn makes it possible to define geodesics between certain shapes. If the infimum in (6) is attained at a diffeomorphism φ_{opt}, then the geodesic between the shapes $[c_1]$ and $[c_2]$ is the equivalence class of the geodesic between $c_1 \circ \varphi_{\mathrm{opt}}$ and c_2. Explicitly, this is given as

$$\tau \mapsto \left[Q^{-1} \left(w^{\mathcal{S}}(1-\tau)(q_1 \circ \varphi_{\mathrm{opt}}) \sqrt{\varphi'_{\mathrm{opt}}} + w^{\mathcal{S}}(\tau) q_2 \right) \right]$$

with

$$w^{\mathcal{S}}(\tau) = \frac{\sin\left(\tau \, \mathrm{dist}^{\mathcal{S}}([c_1], [c_2])\right)}{\sin\left(\mathrm{dist}^{\mathcal{S}}([c_1], [c_2])\right)}.$$

In general, though, the infimum in (6) is not attained in $\mathrm{Diff}_+(I)$. However, it was shown in [6] that a relaxation of the optimisation problem (5) to a larger space of reparametrisations attains its minimum. Denote to that end

$$\Phi := \left\{ \varphi \in AC(I) : \varphi(0) = 0, \ \varphi(1) = 1, \ \varphi'(t) \geq 0 \text{ for a.e. } t \in I \right\}.$$

Lemma 1 (Bruveris 2016)**.** *Assume that c_1, $c_2 \in \mathrm{Imm}(I; \mathbb{R}^d)$. Then*

$$\mathrm{dist}^{\mathcal{S}}([c_1], [c_2]) = \inf_{\varphi_1, \varphi_2 \in \Phi} \mathrm{dist}(c_1 \circ \varphi_1, c_2 \circ \varphi_2). \tag{7}$$

Moreover, the optimisation problem in (7) admits a solution $(\varphi_1, \varphi_2) \in \Phi^2$.

The main topic of this paper is the efficient numerical solution of the optimisation problem (7). In order to do so, we use a more explicit formulation of (7). By applying the chain rule in (2), one obtains that

$$Q(c \circ \varphi)(t) = Q(c)(\varphi(t)) \sqrt{\varphi'(t)}.$$

Thus, the optimisation problem (7) reads explicitly as

$$\inf_{\varphi_1,\,\varphi_2\in\Phi}\arccos\Big(\int_I\Big\langle q_1\big(\varphi_1(t)\big)\sqrt{\varphi_1'(t)},q_2\big(\varphi_2(t)\big)\sqrt{\varphi_2'(t)}\Big\rangle\,dt\Big).$$

Since arccos is monotonically increasing, we can alternatively define

$$J(\varphi_1,\varphi_2):=\int_I\Big\langle q_1\big(\varphi_1(t)\big),q_2\big(\varphi_2(t)\big)\Big\rangle\sqrt{\varphi_1'(t)\varphi_2'(t)}\,dt \tag{8}$$

and compute

$$\mathrm{dist}^{\mathcal{S}}([c_1],[c_2])=\arccos\Big(\sup_{\varphi_1,\,\varphi_2\in\Phi}J(\varphi_1,\varphi_2)\Big). \tag{9}$$

Moreover, it is shown in [6, Proof of Prop. 15] that one can replace the functional J in (9) with its convex relaxation

$$J_c(\varphi_1,\varphi_2):=\int_I\max\big\{\big\langle q_1\big(\varphi_1(t)\big),q_2\big(\varphi_2(t)\big)\big\rangle,0\big\}\sqrt{\varphi_1'(t)\varphi_2'(t)}\,dt$$

without affecting the minimisers.

3 A General Variational Problem

The problem (9) can be seen as a special case of the variational problem

$$\sup_{\boldsymbol{\varphi}\in\mathcal{A}}\left(J(\boldsymbol{\varphi}):=\int_I f(\varphi_1(t),\varphi_2(t))\sqrt{\varphi_1'(t)\varphi_2'(t)}\,dt\right). \tag{10}$$

Here, we denote $\boldsymbol{\varphi}=(\varphi_1,\varphi_2)$ and $\mathcal{A}=\Phi\times\Phi$. Moreover, $f\colon I\times I\to\mathbb{R}_{\geq0}$ is a continuous, non-negative function. Although with a different motivation, this problem has been studied in [8, 7, 20, 10]. In [8], a Hamilton-Jacobi-Bellman (HJB) formulation of this problem was derived, and HJB-based solvers were constructed in [8, 7, 20]. In the following, we recall the main HJB-related results from [8], and provide some useful generalisations of the results.

3.1 Dynamic Programming and the Value Function

For variational problems of the type (10), the solution can be described using dynamic programming. The starting point is the introduction of a value function $u\colon[0,1]^2\to\mathbb{R}$ defined as

$$u(t,\boldsymbol{x}):=\sup_{\boldsymbol{\varphi}\in\mathcal{A}(t,\boldsymbol{x})}\int_0^t f(\varphi_1,\varphi_2)\sqrt{\varphi_1'\varphi_2'}\,dt \tag{11}$$

with

$$\mathcal{A}(t,\boldsymbol{x}):=\big\{\boldsymbol{\varphi}\in AC(I;\mathbb{R}^2):\boldsymbol{\varphi}(0)=\boldsymbol{0},\ \boldsymbol{\varphi}(t)=\boldsymbol{x},\ \varphi'(s)\geq0\text{ for a.e. }s\in I\big\}.$$

Due to the reparametrisation invariance of the integral, we have that $u(t,\boldsymbol{x})$ is independent of t. We will therefore omit the time variable in the definition of u and $\mathcal{A}$ and simply write $u(\boldsymbol{x})=u(1,\boldsymbol{x})$.

In the case when either $x_1 = 0$ or $x_2 = 0$, we have that $\varphi_1'\varphi_2' = 0$ a.e. for all admissible paths. Consequently, the integrand is zero almost everywhere, meaning that we obtain the boundary values $u(0, x_2) = u(x_1, 0) = 0$. Furthermore, the value function satisfies the dynamic programming principle,

$$u(\boldsymbol{\varphi}(t)) \geq u(\boldsymbol{\varphi}(t - h)) + \int_{t-h}^{t} f(\varphi_1, \varphi_2) \sqrt{\varphi_1'\varphi_2'} \, dt, \qquad (12)$$

for all $\boldsymbol{\varphi} \in \mathcal{A}$. Moreover, $\boldsymbol{\varphi}$ is a solution of (10) if and only if we have equality for all t and h. Dividing by h, and taking the limit as $h \to 0$, this means that a solution $\boldsymbol{\varphi}$ formally satisfies the differential equation

$$-\frac{d}{dt}u(\boldsymbol{\varphi}) + f(\boldsymbol{\varphi})\sqrt{\varphi_1'\varphi_2'} = 0,$$

which for smooth u reads

$$-Du(\boldsymbol{\varphi}) \cdot \boldsymbol{\varphi}' + f(\boldsymbol{\varphi})\sqrt{\varphi_1'\varphi_2'} = 0.$$

This means that it is possible to reconstruct $\boldsymbol{\varphi}$ from the value function.

We will now discuss some properties of the value function that will be needed later in the paper. First of all, the dynamic programming principle (12) implies immediately that u is monotone non-decreasing in the sense that $u(\boldsymbol{x}) \geq u(\boldsymbol{y})$ whenever $\boldsymbol{x} \geq \boldsymbol{y}$. Additionally, wherever $f(\boldsymbol{x}) > 0$, the value function is locally strictly increasing: if $x_i > y_i$ element-wise then $u(\boldsymbol{x}) > u(\boldsymbol{y})$. Finally it has been shown that $u(\boldsymbol{x})$ is Hölder continuous with exponent $\frac{1}{2}$ [8, Lemma 1] while $v(\boldsymbol{x}) := u(\boldsymbol{x})^2$ is Lipschitz continuous [7, Lemma 9].

3.2 The HJB equation

For variational problems such as (10), the value function can often be described as the unique solution of the associated Hamilton-Jacobi-Bellman equation. For a general problem of the form

$$\sup_{\boldsymbol{\varphi}} \int_0^1 \ell(\boldsymbol{\varphi}(t), \boldsymbol{\varphi}'(t))dt$$

with associated time dependent value function $u(t, \boldsymbol{x})$, this reads

$$-u_t(t, \boldsymbol{x}) + \sup_{\boldsymbol{\alpha} \in A}\left(-Du(\boldsymbol{x}) \cdot \boldsymbol{\alpha} + \ell(\boldsymbol{x}, \boldsymbol{\alpha})\right) = 0.$$

Here, $\boldsymbol{x}$ and $\boldsymbol{\alpha}$ correspond to $\boldsymbol{\varphi}(t)$ and $\boldsymbol{\varphi}'(t)$, respectively.

As discussed above, the value function is in our case time independent. Thus, we would expect a stationary Hamilton-Jacobi-Bellman equation of the form

$$\begin{cases} H(\boldsymbol{x}, Du) = 0, & \text{in } (0, 1]^2, \\ u(0, x_2) = u(x_1, 0) = 0, \end{cases}$$

with the Hamiltonian

$$H(\boldsymbol{x}, \boldsymbol{p}) = \sup_{\boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^2} -\boldsymbol{p} \cdot \boldsymbol{\alpha} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}.$$

The restriction $\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0}$ follows from the fact that $\boldsymbol{\varphi}'(t)$ takes values in $\mathbb{R}^2_{\geq 0}$. However, since the functional is positively homogeneous in $\boldsymbol{\alpha}$, this leads to a degenerate Hamiltonian which only takes values $H(\boldsymbol{x}, \boldsymbol{p}) \in \{0, +\infty\}$. This property is a consequence of the reparametrisation invariance of the problem, and will be a problem for uniqueness of viscosity solutions of the HJB equation. On the other hand, due to the reparametrisation invariance, we are able to impose restrictions to the admissible space. Therefore, for some well chosen set A representing the admissible derivatives of the paths $\boldsymbol{\varphi}$, we define the Hamiltonian as

$$H(\boldsymbol{x}, \boldsymbol{p}) = \sup_{\boldsymbol{\alpha} \in A} -\boldsymbol{p} \cdot \boldsymbol{\alpha} + f(\boldsymbol{x})\sqrt{\alpha_1 \alpha_2}.$$

We require A to have certain properties.

- First of all, A should reflect the admissible directions of the path $\boldsymbol{\varphi}$. In particular, we must allow for all monotone increasing directions, which means that we must have that $\overline{\mathrm{cone}}\, A = \mathbb{R}^2_{\geq 0}$.
- Secondly, we want the admissible set to permit both negative and positive values for the Hamiltonian (to avoid redundancy of viscosity sub- and supersolutions). This requires that the set A is bounded away from the origin.
- Lastly, we want the admissible set to be compact to allow for the Hamiltonian to have a maximiser $\boldsymbol{\alpha}$. This is not needed for the viscosity characterisation of the value function, but will be a necessary assumption in the construction of numerical solvers.

Together, this can be summarised in the following two assumptions:

Assumption 1. *A satisfies the following:*
(a) $A \subset \mathbb{R}^2_{\geq 0}$ such that $\overline{\mathrm{cone}}\, A = \mathbb{R}^2_{\geq 0}$ and $\inf_A |\boldsymbol{\alpha}| > 0$.
(b) A compact.

There are a few examples of admissible sets which satisfy these assumptions. The natural choices are

$$A_r = \{\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0} \mid |\boldsymbol{\alpha}|_r = \mathrm{const.}\},$$

with $r \in \{1, 2, +\infty\}$. More general, for $1 \leq r \leq +\infty$, A_r satisfies both assumptions 1a and 1b. Additionally, there are options satisfying only assumption 1a including

$$A = \{\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0} \mid \alpha_1 \alpha_2 = \mathrm{const.}\},$$
$$A = \{\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0} \mid \alpha_1 = \mathrm{const.}\}.$$

The first choice is (implicitly) used in [8], while the second choice corresponds to the restriction $\varphi_1' = 1$, which is common in the literature of shape analysis. However, both these cases lead to special situations where the maximum of H is not necessarily attained by any $\boldsymbol{\alpha}$.

We have the following result:

Theorem 2. *Assume that A is such that assumption 1a holds. Then, the value function u is the unique viscosity solution of the hyperbolic PDE*

$$\begin{cases} H(\boldsymbol{x}, Du) = 0, & \text{in } (0, 1]^2, \\ u(0, x_2) = u(x_1, 0) = 0. \end{cases} \tag{13}$$

8

That u is a viscosity solution of the PDE is proved in [8, Theorem 2] while uniqueness follows from [8, Theorem 3]. The actual results in [8] are formulated using a Hamiltonian obtained from the choice $A = \{\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0} \mid \alpha_1\alpha_2 = \text{const.}\}$ after some equivalent reformulations.[1] However, as used in the proof, the positive homogeneity of the functional of H with respect to $\boldsymbol{\alpha}$ implies equivalence of viscosity solutions for all choices of A satisfying assumption 1.

4 Monotone Schemes for the HJB Equation

We will now construct a new family of schemes for solving the HJB equation. The schemes have can be interpreted both as finite difference approximations to the HJB equation similar to the schemes of [7, 8, 20], but also as approximations to the dynamic programming principle (12).

4.1 Schemes based on Du

We start by constructing numerical schemes for approximating solutions to the HJB equation. As in [7, 8, 20], we discretise the unit square into a square grid $[0,1]^2_h := \{0, h, 2h, \ldots, 1\}^2$. Here, we assume that $h = 1/N$, N being the number of discretisation points. For each grid node $\boldsymbol{x}$, we solve a finite difference approximation to the HJB equation, which takes the form

$$\max_{\boldsymbol{\alpha} \in A} -D^- u(\boldsymbol{x})\boldsymbol{\alpha} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} = 0. \tag{14}$$

Here we use the backward difference approximation

$$-D^- u(\boldsymbol{x})\boldsymbol{\alpha} := \frac{u(\boldsymbol{x} - h\boldsymbol{\alpha}) - u(\boldsymbol{x})}{h}. \tag{15}$$

For smooth u, this is a first order approximation to $-Du(\boldsymbol{x})\boldsymbol{\alpha}$. However, the term $u(\boldsymbol{x} - h\boldsymbol{\alpha})$ needs to be approximated as $\boldsymbol{x} - h\boldsymbol{\alpha}$ will not coincide with a grid point for all values of $\boldsymbol{\alpha}$. We will denote this approximation as $g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u)$. Inserting the approximation (15) into (14), this then gives the general scheme

$$\max_{\boldsymbol{\alpha} \in A} \frac{g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u) - u(\boldsymbol{x})}{h} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} = 0. \tag{16}$$

After rearranging the terms, this results in the expression

$$u(\boldsymbol{x}) = \max_{\boldsymbol{\alpha} \in A} g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u) + hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}. \tag{17}$$

The above idea is in contrast to the typical approach as in [7, 8, 20], where $Du(\boldsymbol{x})$, interpreted as a gradient, is approximated numerically. The directional derivative is then computed as the inner product of $\boldsymbol{\alpha}$ with the approximation to $Du(\boldsymbol{x})$. In fact, the approximation (15) can be seen as a generalisation of the typical approach, as we can always approximate $u(\boldsymbol{x} - h\boldsymbol{\alpha})$ using

$$g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u) = u(\boldsymbol{x}) - hD^- u(\boldsymbol{x}) \cdot \boldsymbol{\alpha}.$$

---

[1]This choice of admissible set gives $H(\boldsymbol{x}, \boldsymbol{p}) = -\sqrt{\max\{p_1, 0\}\max\{p_2, 0\}} + \frac{1}{2}f(\boldsymbol{x})$ up to some constant, while [8] uses $H(\boldsymbol{x}, \boldsymbol{p}) = -\max\{p_1, 0\}\max\{p_2, 0\} + \frac{1}{4}f(\boldsymbol{x})^2$. These Hamiltonians will always have the same sign, meaning that they are equivalent in the viscosity sense.

for some approximate gradient $D^- u(\boldsymbol{x})$.

In order to prove convergence of the schemes, we will use the classical proof of Barles & Souganidis for so-called monotone schemes (see [1]) To start, we denote the schemes as $S_h(\boldsymbol{x}, u_h(\boldsymbol{x}), u_h) = 0$ with

$$S_h(\boldsymbol{x}, t, u) = \max_{\boldsymbol{\alpha} \in A} \frac{g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u) - t}{h} + f(\boldsymbol{x}) \sqrt{\alpha_1 \alpha_2}.$$

[1, Theorem 2.1] states that if a scheme is *monotone*, *stable* and *consistent* it is also convergent. Here, we define

- *Monotonicity*: S_h is non-decreasing in u.
- *Stability*: The scheme $S_h(\boldsymbol{x}, u_h(\boldsymbol{x}), u_h) = 0$ has a solution u_h for which $\|u_h\|_\infty \leq$ const. independent of h.
- *Consistency*: For every $\psi \in C^\infty$, we have that

$$\lim_{h \to 0, \boldsymbol{y} \to \boldsymbol{x}, \xi \to 0} S_h(\boldsymbol{y}, \psi(\boldsymbol{y}) + \xi, \psi + \xi) = H(\boldsymbol{x}, D\psi(\boldsymbol{x})).$$

Given A, the scheme (17) is completely determined by g_h. Accordingly, monotonicity, stability and consistency of the scheme can be inferred from the properties of g_h.

Assumption 2. *The approximation* $g_h : [0,1]_h^2 \times A \times C_{\frac{1}{2}}[0,1]_h^2 \to \mathbb{R}$ *satisfies:*

(a) g_h *is monotone non-decreasing in* u.

(b) g_h *is localised: there exists* $C > 0$ *such that, for all functions* $\psi, \xi : [0,1]^2 \to \mathbb{R}$ *that satisfy* $\psi = \xi$ *on the ball* $B_{Ch}(\boldsymbol{x})$ *of radius* Ch *centered at* $\boldsymbol{x}$, *we have that*

$$g_h(\boldsymbol{x}, \boldsymbol{\alpha}, \psi) = g_h(\boldsymbol{x}, \boldsymbol{\alpha}, \xi).$$

(c) *For constant* $\psi = \psi_0$, *we have that* $g_h(\boldsymbol{x}, \boldsymbol{\alpha}, \psi) = \psi_0$.

(d) g_h *is a superlinear approximation: for every* $\psi \in C^\infty([0,1]^2)$ *and all* $L > 0$, *there exists a modulus of continuity* $\omega_{\psi,L}$ *such that*

$$\left| \frac{g_h(\boldsymbol{x}, \boldsymbol{\alpha}, \psi + \xi) - \psi(\boldsymbol{x} - h\boldsymbol{\alpha}) - \xi}{h} \right| \leq \omega_{\psi,L}(h)$$

for every $|\xi| \leq L$.

Theorem 3. *Under assumptions 1 and 2, the scheme* (17) *is convergent.*

Proof. The scheme satisfies the following properties:

- *Monotonicity*: S_h is increasing in g_h and g_h is non-decreasing in u. Hence S_h is non-decreasing in u.
- *Stability*: The solution u_h of $S_h = 0$ is explicitly given in (17). Denote in the following

$$B_{Ch}^-(\boldsymbol{x}) := \{ \boldsymbol{y} \in I \cap B_{Ch}(\boldsymbol{x}) : \boldsymbol{y} \leq \boldsymbol{x} \}.$$

Due to assumptions 2a to 2c, we have that

$$\min_{\boldsymbol{y} \in B_{Ch}^-(\boldsymbol{x})} u_h(\boldsymbol{y}) \leq g_h\left(\boldsymbol{x}, \boldsymbol{\alpha}, \min_{\boldsymbol{y} \in B_{Ch}^-(\boldsymbol{x})} u_h(\boldsymbol{y})\right)$$

$$\leq g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u_h) \leq g_h\left(\boldsymbol{x}, \boldsymbol{\alpha}, \max_{\boldsymbol{y} \in B_{Ch}^-(\boldsymbol{x})} u_h(\boldsymbol{y})\right) = \max_{\boldsymbol{y} \in B_{Ch}^-(\boldsymbol{x})} u_h(\boldsymbol{y})$$

for every $h > 0$. Inserting these estimates into (17), we obtain that

$$\min_{\boldsymbol{y} \in B^-_{Ch}(x)} u_h(\boldsymbol{y}) \leq u_h(\boldsymbol{x}) \leq \max_{\boldsymbol{y} \in B^-_{Ch}(x)} u_h(\boldsymbol{y}) + h\|f\|_\infty A_{max}$$

with $A_{max} = \max_{\boldsymbol{\alpha} \in A} \sqrt{\alpha_1 \alpha_2}$. If we now consider $[0,1]^2_h$ as an directed acyclic graph, each path from $\to \boldsymbol{0}$ to $\boldsymbol{x} \in [0,1]^2_h$ has length of at most $2N$. This gives that $0 \leq u_h(\boldsymbol{x}) \leq 2\|f\|_\infty A_{max}$.

- *Consistency*: Since A is compact, we have that

$$\lim_{\substack{h \to 0 \\ \boldsymbol{y} \to \boldsymbol{x} \\ \xi \to 0}} S_h\big(\boldsymbol{y}, \psi(\boldsymbol{y}) + \xi, \psi + \xi\big)$$

$$= \lim_{\substack{h \to 0 \\ \boldsymbol{y} \to \boldsymbol{x} \\ \xi \to 0}} \max_{\boldsymbol{\alpha} \in A} \frac{g_h(\boldsymbol{y}, \boldsymbol{\alpha}, \psi + \xi) - \psi(\boldsymbol{y}) - \xi}{h} + f(\boldsymbol{y})\sqrt{\alpha_1 \alpha_2}$$

$$= \lim_{\substack{h \to 0 \\ \boldsymbol{y} \to \boldsymbol{x} \\ \xi \to 0}} \max_{\boldsymbol{\alpha} \in A} \frac{\psi(\boldsymbol{y} - h\boldsymbol{\alpha}) - \psi(\boldsymbol{y})}{h} + \omega_{\psi,L}(h) + f(\boldsymbol{y})\sqrt{\alpha_1 \alpha_2}$$

$$\overset{(*)}{=} \max_{\boldsymbol{\alpha} \in A} \lim_{\substack{h \to 0 \\ \boldsymbol{y} \to \boldsymbol{x} \\ \xi \to 0}} \frac{\psi(\boldsymbol{y} - h\boldsymbol{\alpha}) - \psi(\boldsymbol{y})}{h} + \omega_{\psi,L}(h) + f(\boldsymbol{y})\sqrt{\alpha_1 \alpha_2}$$

$$= \max_{\boldsymbol{\alpha} \in A} D\psi(\boldsymbol{x})(\boldsymbol{\alpha}) + f(\boldsymbol{x})\sqrt{\alpha_1 \alpha_2}$$

$$= H(\boldsymbol{x}, D\psi(\boldsymbol{x})).$$

In $(*)$, we used that the functional is uniformly continuous in $\boldsymbol{y}, \boldsymbol{\alpha}, h, \xi$ to exchange the limit and maximisation.

Due to [1, Theorem 2.1], this proves convergence. $\qquad\square$

4.2 Schemes based on $D(u^2)$

Recall that u is only Hölder continuous with exponent $\frac{1}{2}$ while u^2 is Lipschitz continuous. This means that one might expect more accurate schemes based on an approximation of u^2 rather than u. This is done in [7], where schemes are constructed for $v := u^2$.[2]

The idea is to utilise that $D(u^2) = 2uDu$, meaning that $Du = D(u^2)/2u$ wherever $u \geq 0$. In such, one would expect v to be a viscosity solution of

$$\max_{\boldsymbol{\alpha} \in A} \frac{D(u(\boldsymbol{x})^2)(\boldsymbol{\alpha})}{2u(\boldsymbol{x})} + f(\boldsymbol{x})\sqrt{\alpha_1 \alpha_2} = 0.$$

Already, this equation has problems with the singularity at $u(\boldsymbol{x}) = 0$. However, we proceed by assuming for now that $u(\boldsymbol{x}) > 0$. We can then follow the above idea and construct schemes for v based on the approximation

$$\max_{\boldsymbol{\alpha} \in A} \frac{g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u)^2 - u(\boldsymbol{x})^2}{2u(\boldsymbol{x})h} + f(\boldsymbol{x})\sqrt{\alpha_1 \alpha_2} = 0. \tag{18}$$

---

[2]Additionally, a scheme for the Lipschitz continuous term $w := u/\sqrt{x_1 x_2}$ is also constructed in [7]. However, we deem this idea ill-suited for our approach due to the lack of simple closed-form expressions.

For fixed $\boldsymbol{\alpha}$, this can be modified into a quadratic equation in $u(\boldsymbol{x})$, meaning that schemes of this form will have multiple solutions. However, similar to the schemes in [7], we are only interested in the largest of the solutions.

Immediately, this means that $u(\boldsymbol{x}) \geq hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}$. Moreover, we can multiply (18) with $2u(\boldsymbol{x})h$ to find that

$$\max_{\boldsymbol{\alpha} \in A}\Big[g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u)^2 - u(\boldsymbol{x})^2 + 2u(\boldsymbol{x})hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}\Big]$$

$$= \max_{\boldsymbol{\alpha} \in A}\Big[-\underbrace{\Big(u(\boldsymbol{x})^2 - hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}\Big)^2}_{=:F(\boldsymbol{\alpha})} + \underbrace{h^2 f(\boldsymbol{x})^2\alpha_1\alpha_2 + g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u)^2}_{=:G^2(\boldsymbol{\alpha})}\Big]$$

$$= \max_{\boldsymbol{\alpha} \in A}\Big[-F(\boldsymbol{\alpha})^2 + G(\boldsymbol{\alpha})^2\Big] = 0$$

with $F, G \geq 0$. Observe that we have that

$$-F^2(\boldsymbol{\alpha}) + G^2(\boldsymbol{\alpha}) = (F(\boldsymbol{\alpha}) + G(\boldsymbol{\alpha}))(-F(\boldsymbol{\alpha}) + G(\boldsymbol{\alpha})) \leq 0$$

for all $\boldsymbol{\alpha} \in A$ with equality if and only if $\boldsymbol{\alpha}$ is optimal. Moreover, equality can only be achieved if $-F(\boldsymbol{\alpha}) + G(\boldsymbol{\alpha}) = 0$. Accordingly, the above scheme is identical to

$$\max_{\boldsymbol{\alpha} \in A} -F(\boldsymbol{\alpha}) + G(\boldsymbol{\alpha})$$

$$= \max_{\boldsymbol{\alpha} \in A} -u(\boldsymbol{x}) + hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} + \sqrt{h^2 f(\boldsymbol{x})^2\alpha_1\alpha_2 + g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u)^2} = 0,$$

which gives the closed form expression

$$u(\boldsymbol{x}) = \max_{\boldsymbol{\alpha} \in A} hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} + \sqrt{h^2 f(\boldsymbol{x})^2\alpha_1\alpha_2 + g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u)^2}. \qquad (19)$$

Theorem 4. *Under assumptions 1 and 2, the scheme (19) is convergent.*

Proof. Consider

$$\overline{u}(\boldsymbol{x}) = \limsup_{\substack{\boldsymbol{x} \to \boldsymbol{y} \\ h \to 0}} u_h(\boldsymbol{y}), \quad \underline{u}(\boldsymbol{x}) = \liminf_{\substack{\boldsymbol{x} \to \boldsymbol{y} \\ h \to 0}} u_h(\boldsymbol{y}).$$

The proof of [1, Theorem 2.1] still holds for all $\boldsymbol{x}$ for which $\underline{u}(\boldsymbol{x}) > 0$, and proving the properties of monotonicity, stability and consistency is similar to that of Theorem 3. Wherever $\underline{u}(\boldsymbol{x}) = 0$, the singularity of the scheme breaks the proof of convergence. However, we have that $u(\boldsymbol{x}) = 0$ if and only if $f(\boldsymbol{y}) = 0$ for all $\boldsymbol{y} \leq \boldsymbol{x}$. Accordingly, it is sufficient to prove that this property holds for $\underline{u}$ and $\overline{u}$ as well.

Observe that

$$u_h(\boldsymbol{x}) = \max_{\boldsymbol{\alpha} \in A} hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} + \sqrt{h^2 f(\boldsymbol{x})^2\alpha_1\alpha_2 + g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u_h)^2}$$

$$\geq \max_{\boldsymbol{\alpha} \in A} hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} + g_h(\boldsymbol{x}, \boldsymbol{\alpha}, u_h).$$

Inductively, this gives that $u_h \geq \tilde{u}_h$ where $\tilde{u}_h$ is the solution of (17) for the same choices of A and g_h. Since $\tilde{u}_h$ is convergent, this implies that $u_h(\boldsymbol{x}) \geq \tilde{u}_h(\boldsymbol{x}) > 0$ wherever $f(\boldsymbol{x}) > 0$.

Now assume that $f(\boldsymbol{y}) = 0$ for all $\boldsymbol{y} \leq \boldsymbol{x}$. Then, since $g_h(\cdot, \cdot, 0) = 0$, it is clear that $u_h(\boldsymbol{x}) = \tilde{u}_h(\boldsymbol{x}) = 0$, concluding the proof.

$\square$

5 Proposed Schemes

Now, it remains to choose A and g_h such that we obtain efficient schemes. In particular, we desire closed form expressions for both u_h and the optimal $\boldsymbol{\alpha}$ used in each step.

For the closed form expressions for the schemes, it is useful to denote $\boldsymbol{x}_1^1$ as the current grid point for which we are solving the schemes. In addition, we denote $\boldsymbol{x}_0^1$, $\boldsymbol{x}_1^0$ and $\boldsymbol{x}_0^0$ as the other three points of the grid cell. Moreover, denote $u_i^j = u(\boldsymbol{x}_i^j)$ and $f_i^j = f(\boldsymbol{x}_i^j)$. Until now, we have assumed that f_i^j is evaluated exactly. For approximations of this term, see appendix A.

5.1 Schemes based on $\boldsymbol{Du}$

For the scheme (17), we start by letting g_h be the linear interpolation of u through the points $\boldsymbol{x}_0^1$, $\boldsymbol{x}_1^0$ and $\boldsymbol{x}_0^0$, which reads

$$
\begin{aligned}
g_h &= u_0^0 + (1 - \alpha_1)(u_1^0 - u_0^0) + (1 - \alpha_2)(u_0^1 - u_0^0) \\
&= (\alpha_1 + \alpha_2 - 1)u_0^0 + (1 - \alpha_1)u_1^0 + (1 - \alpha_2)u_0^1.
\end{aligned} \tag{20}
$$

It is easy to check that this satisfies all properties of assumption 2 apart from monotonicity. To ensure monotonicity, it is required that $0 \leq \alpha_1, \alpha_2 \leq 1$ and that $\alpha_1 + \alpha_2 \geq 1$. This holds for every $\alpha \in A_r := \{\boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^2 \mid |\boldsymbol{\alpha}|_r = 1\}$ for $1 \leq r \leq +\infty$. We find that for the choices $A = A_1$ and $A = A_\infty$, we can solve the schemes analytically.

Choosing $A = A_1$, combined with the first approximation (20), we obtain the following solution for the scheme, and the optimal $\boldsymbol{\alpha}^*$:

$$
hS_h = u_1^1 - \frac{1}{2}\left(u_0^1 + u_1^0 + \sqrt{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\right),
$$

$$
u_1^1 = \frac{1}{2}\left(u_0^1 + u_1^0 + \sqrt{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\right), \tag{U_1}
$$

$$
\boldsymbol{\alpha}^* = \left(\frac{1}{2}\Big(1 + \frac{u_0^1 - u_1^0}{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\Big), \frac{1}{2}\Big(1 - \frac{u_0^1 - u_1^0}{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\Big)\right).
$$

Interestingly, this is exactly the original scheme proposed in [8]. Using $A = A_\infty$, we obtain with the abbreviation

$$
u_*^* := \max\{u_0^1, u_1^0\}
$$

that

$$
hS_h = u_1^1 - \begin{cases} u_*^* + \dfrac{(hf_1^1)^2}{4(u_*^* - u_0^0)}, & 2(u_*^* - u_0^0) > \sqrt{(hf_1^1)^2}, \\ u_0^0 + hf_1^1, & \text{otherwise,} \end{cases}
$$

$$
u_1^1 = \begin{cases} u_*^* + \dfrac{(hf_1^1)^2}{4(u_*^* - u_0^0)}, & 2(u_*^* - u_0^0) > \sqrt{(hf_1^1)^2}, \\ u_0^0 + hf_1^1, & \text{otherwise,} \end{cases} \tag{U_∞}
$$

$$
\boldsymbol{\alpha}^* = \begin{cases} \left(1, \dfrac{hf_1^1}{2(u_0^1 - u_0^0)}\right), & u_0^1 \geq u_1^0, 2(u_0^1 - u_0^0) > \sqrt{(hf_1^1)^2}, \\ \left(\dfrac{hf_1^1}{2(u_1^0 - u_0^0)}, 1\right), & u_1^0 > u_0^1, 2(u_1^0 - u_0^0) > \sqrt{(hf_1^1)^2}, \\ (1, 1), & \text{otherwise.} \end{cases}
$$

5.2 Schemes based on $D(u^2)$

For the scheme (17), it is useful to express the schemes for $v_i^j := (u_i^j)^2$. Instead of linearly interpolating u, we linearly interpolate v, meaning that

$$
\begin{aligned}
g_h^2 &= v_0^0 + (1 - \alpha_1)(v_1^0 - v_0^0) + (1 - \alpha_2)(v_0^1 - v_0^0) \\
&= (\alpha_1 + \alpha_2 - 1)v_0^0 + (1 - \alpha_1)v_1^0 + (1 - \alpha_2)v_0^1.
\end{aligned}
\tag{21}
$$

Here, we have the same conditions for monotonicity as for the schemes based on Du, and we have analytical solutions for $A = A_1$:

$$
2h\sqrt{v_1^1}S_h = v_1^1 - \frac{1}{2}\left(v_0^1 + v_1^0 + \sqrt{(v_0^1 - v_1^0)^2 + (hf_1^1)^2} \right),
$$

$$
v_1^1 = \frac{1}{2}\left(v_0^1 + v_1^0 + h^2 f^2 + \sqrt{(v_0^1 - v_1^0)^2 + 2(v_0^1 - v_1^0)(hf_1^1)^2 + (hf_1^1)^4} \right),
$$

$$
\boldsymbol{\alpha}^* = \left(\frac{1}{2}\Big(1 + \frac{v_0^1 - u_1^0}{(v_0^1 - v_1^0)^2 + 4v_1^1(hf_1^1)^2}\Big), \frac{1}{2}\Big(1 - \frac{v_0^1 - u_1^0}{(v_0^1 - v_1^0)^2 + 4v_1^1(hf_1^1)^2}\Big) \right).
\tag{V_1}
$$

Again, we have that the scheme using A_1 is identical to that of [7]. For $A = A_\infty$, we have with

$$
v_*^* := \max\{v_0^1, v_1^0\}
$$

that

$$
2h\sqrt{v_1^1}S_h = v_1^1 -
\begin{cases}
v_*^* + \dfrac{v_1^1(hf_1^1)^2}{v_*^* - v_0^0}, & v_*^* - v_0^0 > \sqrt{v_1^1}hf_1^1, \\
v_0^0 + hf_1^1, & \text{otherwise,}
\end{cases}
$$

$$
v_1^1 =
\begin{cases}
\dfrac{v_*^*(v_*^* - v_0^0)}{v_*^* - v_0^0 - (hf_1^1)^2} & (v_*^* - v_0^0)(v_*^* - v_0^0 - (hf_1^1)^2) > v_*^* hf_1^1, \\
u_0^0 + hf_1^1, & \text{otherwise,}
\end{cases}
$$

$$
\boldsymbol{\alpha}^* =
\begin{cases}
\left(1, \dfrac{\sqrt{v_1^1}hf_1^1}{v_0^1 - v_0^0} \right), & v_0^1 \geq v_1^0, \, v_0^1 - v_0^0 > \sqrt{v_1^1}hf_1^1, \\
\left(\dfrac{\sqrt{v_1^1}hf_1^1}{v_1^0 - v_0^0}, 1 \right), & v_1^0 > v_0^1, \, v_1^0 - v_0^0 > \sqrt{v_1^1}hf_1^1, \\
(1, 1), & \text{otherwise.}
\end{cases}
\tag{V_∞}
$$

Due to the singularity at $v_1^1 = 0$, we have that S_h is not defined in these cases. However, the solution of the schemes still exist.

5.3 Higher Order Filtered Schemes

It is known that one cannot construct higher order schemes for solving HJB equations, as one requires monotone schemes to obtain convergence. Still, it is common to construct so-called *filtered* schemes. These schemes are based on a high-order (possibly non-monotone) scheme S_h^a, and a monotone scheme S_h^m. The idea is to choose the higher order scheme only if its approximation to the Hamiltonian is sufficiently close to that of the monotone scheme. The selection criterion is typically chosen as $|S_h^a - S_h^m| \leq k\sqrt{h}$ for some constant k to preserve

the theoretical $\sqrt{h}$ convergence which is typical for schemes for HJB equations. Therefore, we define the filtered scheme as

$$S_h^f := \begin{cases} S_h^a, & |S_h^a - S_h^m| \le k\sqrt{h}, \\ S_h^m, & |S_h^a - S_h^m| > k\sqrt{h}. \end{cases}$$

This can be implemented by first solving $S_h^a = 0$. If this solutions satisfies $|S_h^m| \le k\sqrt{h}$, we keep the solution. Otherwise, we use the solution of $S_h^m = 0$.

To construct the high order scheme, we use the same idea as in section 4, except that we use central differences, rather than backward differences. In practice, this means that we approximate

$$-Du(\boldsymbol{x})\boldsymbol{\alpha} = \frac{u(\boldsymbol{x} - \frac{h}{2}\boldsymbol{\alpha}) - u(\boldsymbol{x} + \frac{h}{2}\boldsymbol{\alpha})}{h}$$

with a similar approximation to $D(u^2)$. Approximating $u(\boldsymbol{x} - \frac{h}{2}\boldsymbol{\alpha})$ and $u(\boldsymbol{x} + \frac{h}{2}\boldsymbol{\alpha})$ using $g_{\frac{h}{2}}$, we obtain the two general schemes:

$$S_h^a = \max_{\alpha \in A} \frac{g_{\frac{h}{2}}(\boldsymbol{x}, \boldsymbol{\alpha}, u) - g_{\frac{h}{2}}(\boldsymbol{x}, -\boldsymbol{\alpha}, u)}{h} + f(x)\sqrt{\alpha_1 \alpha_2} = 0 \qquad (22)$$

$$S_h^a = \max_{\alpha \in A} \frac{g_{\frac{h}{2}}(\boldsymbol{x}, \boldsymbol{\alpha}, u)^2 - g_{\frac{h}{2}}(\boldsymbol{x}, -\boldsymbol{\alpha}, u)^2}{2u(\boldsymbol{x})h} + f(x)\sqrt{\alpha_1 \alpha_2} = 0. \qquad (23)$$

These schemes will be solved with $\boldsymbol{x}$ being the centre of a grid cell.

In (22), we approximate $g_{\frac{h}{2}}(\boldsymbol{x}, \boldsymbol{\alpha}, u)$ using a linear interpolation of u_0^1, u_1^0 and u_0^0 for positive $\boldsymbol{\alpha}$ and a linear interpolation of u_0^1, u_1^0 and u_1^1 for negative $\boldsymbol{\alpha}$. Interestingly, this gives a scheme which is independent of A, reading

$$u_1^1 = u_0^0 + \sqrt{(u_0^1 - u_1^0)^2 + (hf_1^1)^2},$$

$$\boldsymbol{\alpha}^* = \left(\frac{1}{2}\left(1 + \frac{u_0^1 - u_1^0}{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\right), \frac{1}{2}\left(1 - \frac{u_0^1 - u_1^0}{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\right) \right).$$

With similar linear approximations in (23), we need an approximation of $u(\boldsymbol{x})$, present in the denominator, since $\boldsymbol{x}$ does not coincide with a grid cell. Using $2u(\boldsymbol{x}) \approx u_0^1 + u_1^0$, we obtain

$$v_1^1 = v_0^0 + \frac{1}{2}h^2 f^2 + \sqrt{(v_0^1 - v_1^0)^2 + (2v_0^0 + v_0^1 + v_1^0)(hf_1^1)^2 + \frac{1}{4}(hf_1^1)^4},$$

$$\boldsymbol{\alpha}^* = \left(\frac{1}{2}\left(1 + \frac{v_0^1 - u_1^0}{(v_0^1 - v_1^0)^2 + 4v_1^1(hf_1^1)^2}\right), \frac{1}{2}\left(1 - \frac{v_0^1 - u_1^0}{(v_0^1 - v_1^0)^2 + 4v_1^1(hf_1^1)^2}\right) \right).$$

5.4 Fully Discretised Schemes

The scheme (16) can in fact be used to formulate fully discretised schemes with some modification. We start by replacing A with a variable admissible space $A = A_h \subset \mathbb{N}_0^2 \setminus \boldsymbol{0}$, i.e. the set of pairs of non-negative integers. With this choice, $\boldsymbol{x} - h\boldsymbol{\alpha}$ coincides with other grid points of the square grid (as long as $\boldsymbol{x}$ lies on a grid point). In such, the scheme

$$u(\boldsymbol{x}) = \max_{\boldsymbol{\alpha} \in A_h} u(\boldsymbol{x} - h\boldsymbol{\alpha}) + f(\boldsymbol{x})\sqrt{\alpha_1 \alpha_2} \qquad (24)$$

can be solved without the need for an approximation to $u(\boldsymbol{x} - h\boldsymbol{\alpha})$. By setting $\boldsymbol{x} = \boldsymbol{x}_i^j \in [0,1]_h^2$, this scheme reads

$$u_i^j = \max_{(k,l) \in A_h} u_{i-k}^{j-l} + h f_i^j \sqrt{kl}. \tag{DDP}$$

This is exactly the discretised dynamic programming method commonly used in the literature. Under certain assumptions on A_h, we can still use the HJB based approach to prove convergence of this scheme. See assumption 3 and Theorem 5 in Appendix B for details.

Choosing A_h requires a compromise between accuracy and complexity. We want A_h to include as many directions as possible for optimal accuracy. However, the larger the set A_h, the higher the computational cost. One example of at set satisfying assumption 3 is

$$A_h = \{\boldsymbol{\alpha} \in \mathbb{N}_0^2 \mid |\boldsymbol{\alpha}| \leq kh^{-r}\}$$

for constants $k > 0$ and $0 < r < 1$. Here, $r = \frac{1}{2}$ will typically give a good compromise between accuracy and efficiency.

6   Numerical Computation of Geodesics

The numerical solution of the value function gives an estimate to $u(\boldsymbol{1})$, which in turn can be used to approximate the geodesic distance through $\mathrm{dist}^{\mathcal{S}}([c_1], [c_2]) = \arccos(u(\boldsymbol{1}))$. Additionally, through a backtracking method, we can use the value function to obtain an estimate of the solution $\boldsymbol{\varphi}$ of the variational problem (10). This can then be used to estimate the shape space geodesic between c_1 and c_2.

6.1   Backtracking

To retrieve the optimal reparametrisation path $\boldsymbol{\varphi}$, we propose a piecewise constant interpolation of the maximiser $\boldsymbol{\alpha}^*$ of the approximated HJB equation, where $\boldsymbol{\alpha}^*$ is constant on each grid cell $(x_i, x_{i+1}] \times (x^j, x^{j+1}]$. With $\boldsymbol{\varphi}'(t) = \boldsymbol{\alpha}^*$, this gives a first order piecewise constant differential equation for $\boldsymbol{\varphi}'$, which therefore can be computed explicitly.

In practice, the path $\boldsymbol{\varphi}$ will be piecewise linear, only changing direction when intersecting a grid line, meaning that the path can be represented by a sequence $\{\boldsymbol{\varphi}_k\}$ with length at most $2N$. Assume that the backtracking procedure has reached the point $\boldsymbol{\varphi}_k \in (x_i, x_{i+1}] \times (x^j, x^{j+1}]$. In order to obtain the next point in the sequence, we construct the line $\boldsymbol{\psi}(t) = \boldsymbol{\varphi}_k - t\boldsymbol{\alpha}^*$, defined for $t \geq 0$ where $\boldsymbol{\alpha}^*$ is optimal for the given grid cell. Then, we find the intersection point between $\boldsymbol{\psi}$ and the vertical line $(x_i, \cdot)$ and the intersection point between $\boldsymbol{\psi}$ and the horizontal line $(\cdot, x^j)$. The next point in the sequence will then be the maximum of these points. This reads

$$\boldsymbol{\varphi}_{k-1} = \max\big\{(x_i, \varphi_{2,k} - (\varphi_{1,k} - x_i)\alpha_2^*/\alpha_1^*), (\varphi_{1,k} - (\varphi_{2,k} - x^j)\alpha_1^*/\alpha_2^*, x^j)\big\}.$$

Note that since the path $\boldsymbol{\varphi}$ is monotone increasing, one of the intersection points will actually be maximal with respect to the standard partial ordering of $\mathcal{R}^2$.

The terminal condition for the path $\boldsymbol{\varphi}$ is $\boldsymbol{\varphi}(1) = \mathbf{1}$, which also acts as the starting point for the backtracking procedure. With the convention that $\boldsymbol{\alpha}^* = (1, 0)$ wherever $\boldsymbol{x} = (x_1, 0)$ and $\boldsymbol{\alpha}^* = (0, 1)$ wherever $\boldsymbol{x} = (0, x_2)$, we ensure that the (inferred) initial condition $\boldsymbol{\varphi}(0) = \mathbf{0}$ is met.

6.2 Computing Geodesics and Geodesic Distances

Now that we have an estimate of $\boldsymbol{\varphi}$, we can estimate the SRVTs after reparametrisation. Similar to the reparametrisation path, we construct a sequence of points of the form

$$q_{i,k} := q_i(\varphi_{i,k})\sqrt{\frac{\varphi_{i,k} - \varphi_{i,k-1}}{\Delta t_k}},$$

for $i = 1, 2$. Note that this expression requires Δt_k, representing the joint parametrisation of φ_1 and φ_2. Since the problem is reparametrisation invariant, this can be chosen based on the application. One natural option is to choose $\Delta t_k = \frac{1}{2}(\varphi_{1,k} - \varphi_{1,k-1} + \varphi_{2,k} - \varphi_{2,k-1})$, motivated from the assumption that $\|\boldsymbol{\varphi}'\|_1 = 1$. This constraint is especially useful since $\boldsymbol{\varphi}'$ is bounded and the domain I remains unchanged.

Using the point estimates of the SRVTs, we can approximate the objective function and the geodesics. First of all, for the objective function, we have the following estimate:

$$
\begin{aligned}
J_h(\boldsymbol{\varphi}_h) &= \sum_k \langle q_{1,k},\, q_{2,k} \rangle \Delta t_k \\
&= \sum_k \langle q_1(\varphi_{1,k}),\, q_2(\varphi_{2,k}) \rangle \sqrt{(\varphi_{k,1} - \varphi_{k-1,1})(\varphi_{k,2} - \varphi_{k-1,2})}.
\end{aligned}
$$

Observe in particular that this expression is independent of Δt_k, as desired.

Similarly, we can pointwise approximate the geodesic using

$$
\begin{aligned}
\gamma_k(\tau) &= w_h^{\mathcal{S}}(1 - \tau)q_{1,k} + w_h^{\mathcal{S}}(\tau)q_{2,k} \\
&= w_h^{\mathcal{S}}(1 - \tau)q_1(\varphi_{1,k})\sqrt{\frac{\varphi_{1,k} - \varphi_{1,k-1}}{\Delta t_k}} + w_h^{\mathcal{S}}(\tau)q_2(\varphi_{2,k})\sqrt{\frac{\varphi_{2,k} - \varphi_{2,k-1}}{\Delta t_k}},
\end{aligned}
$$

where $w_h^{\mathcal{S}}(\tau) = \sin(\tau \arccos J_h(\boldsymbol{\varphi}_h))/\sin(\arccos J_h(\varphi_h))$. In the pre-shape space, the geodesic can be approximated using

$$Q^{-1}(\gamma(\tau))(t_k) \approx \sum_{l=1}^{k} \gamma_l(\tau)|\gamma_l(\tau)|\Delta t_l = \sum_{l=1}^{k} \gamma_l(\tau)\sqrt{\Delta t_l}\left|\gamma_l(\tau)\sqrt{\Delta t_l}\right|.$$

Similarly to the objective function, this estimate is independent of Δt, as desired.

7 Numerical Experiments

To test the numerical schemes, we use three test problems labeled A, B, C for which the curves and shape space geodesics are illustrated in Figure 1. For each of the test problems, the schemes were ran with grid sizes $N = h^{-1} = 5 \cdot 2^2, \ldots, 5 \cdot 2^{10}$ (for the discretised dynamic programming, the smallest two step sizes were omitted due to computational complexity).
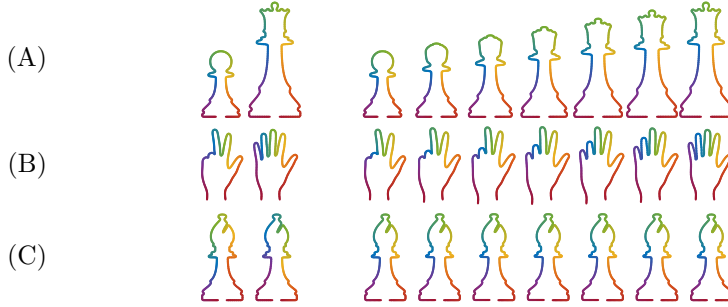
Figure 1: Left: curves coloured by initial parametrisation. Right: shape space geodesics.

For test problems (A) and (B), we use arc length parametrisation as the initial parametrisation of the curves. For these problems, we do not have analytic solutions for any of the variables of interest. The analytic solutions were therefore approximated using the filtered scheme with a fine grid size $h^{-1} = \epsilon^{-1} = 5 \cdot 2^{11}$. For test problem (C), we compare two curves with equal shape but different initial parametrisations. In particular, we let $c_1 = c_0 \circ \psi_1$ and $c_2 = c_0 \circ \psi_2$ with c_0 being the arc length parametrisation of the curve. Here, we use the Möbius transformations

$$\psi_1(t) = 3t/(1 + 2t), \quad \psi_2(t) = t/(3 - 2t),$$

which are each other's inverses. Hence, one solution of the reparametrisation problem is given by $\varphi_1 = \psi_1^{-1} = \psi_2$ and $\varphi_2 = \psi_2^{-1} = \psi_1$. For this problem, we have the exact geodesic distance $d(c_1, c_2) = 1$ and exact expressions for the geodesics (which are constant in τ).

It has been demonstrated in [20] that filtered schemes can give an improvement for simple problems. However, for our experiments, we found that there was no significant improvement of the filtered schemes compared to the best performing of the monotone schemes.[3] Therefore, we will only compare the four monotone schemes presented together with the fully discretised dynamic programming. For the discretised dynamic programming, we tested $A_h = \{\boldsymbol{\alpha} \in \mathbb{N}_0^2 \mid |\boldsymbol{\alpha}| \leq kh^{-r}\}$ for different values of k, r. We found that $k = \frac{3}{4}$ and $r = \frac{1}{2}$ gave the best performance when measuring accuracy vs computation time. For all schemes, we found that the approximation of f as described in appendix A gives better results. These approximations were therefore used in the following experiments.

7.1 Presence of Local Solutions

Dynamic programming based methods typically converge slower than gradient based method. Therefore, it is important to assess whether local solutions are present or not. In order to do this, we consider the total value function

$$u_{tot}(\boldsymbol{x}) := \sup_{\boldsymbol{\varphi} \in \mathcal{A}} \int_0^1 f(\varphi_1, \varphi_2)\sqrt{\varphi_1' \varphi_2'} dt \quad \text{s.t.} \quad \boldsymbol{\varphi}(\tfrac{1}{2}) = \boldsymbol{x}.$$

---

[3]There is in some cases a small improvement, but this is outweighed by the added computational time.
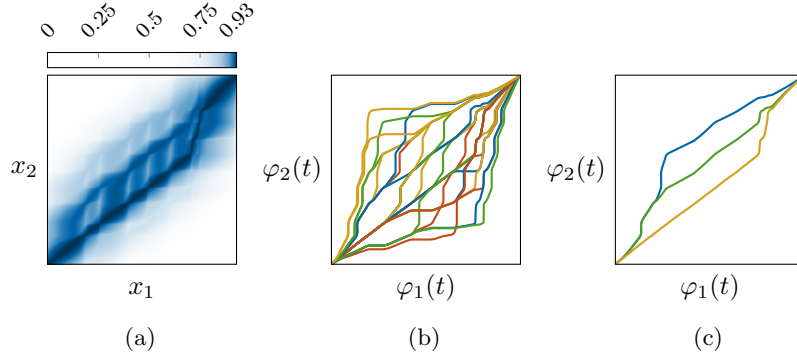
Figure 2: The total value function (a), some local maxima of (10) (b), and three local maxima near the diagonal (c).
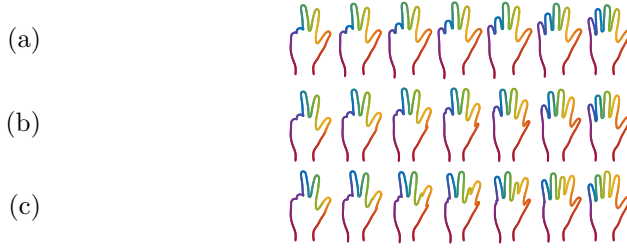

Figure 3: Pre-shape geodesics with parametrisation corresponding to the three local maxima visualised in Figure 2c.

This variant of the value function measures the similarity between the curves c_1, c_2 given a landmark constraint at $\boldsymbol{x}$, that is, requiring that the point $c_1(x_1)$ is registered to $c_2(x_2)$. If u_{tot} has a local maximum at $\boldsymbol{x}$, there is a local solution of (10) passing through $\boldsymbol{x}$. This means that u_{tot} can be used to find local solutions. The total value function will not characterise all local solutions, but the number of local maxima of u_{tot} is an indication of the number of local solutions of (10). Note that all maxima of u_{tot} are inherently flat, meaning that there are in practice paths of local maxima.

The total value function is easy to compute. The "standard" value function (11) was defined by maximising over all paths from $\boldsymbol{0}$ to $\boldsymbol{x}$. Alternatively, we can define a reversed value function where we optimise over all paths from $\boldsymbol{x}$ to $\boldsymbol{1}$. Since the problem is fundamentally invariant to reparametrisations, these are identical problems up to replacing $f(x_1, x_2)$ with $f(1 - x_1, 1 - x_2)$. Then, the sum of the standard and reversed value functions together gives the total value function. For each local maximum of u_{tot}, one can run the backtracking algorithm in both directions to obtain a local solution of (10).

For test problem (B), the total value function was estimated using $h = 5 \cdot 10^{-4}$. The estimate is visualised in Figure 2a and the local maxima of (10) are visualised in Figure 2a. Note that a highly nonlinear colormap has been used in Figure 2a to accentuate the local maxima. Using this method, 27 local maxima were found. However, this method of finding local maxima is conservative, and there are likely a lot more. We chose three local maxima close to the diagonal,

19

and computed the resulting curve space geodesics. The result is visualised in Figure 3. As one can see, the resulting geodesics are very different. A priori, it is hard to tell which one of these solutions a local, gradient based methods will find. This accentuates the importance of global solvers.

7.2 Convergence of the Value Function

For the value function, we have theoretical point-wise uniform convergence. Therefore, the natural metric for evaluating convergence is the L^∞-error. We approximate this error by a point-wise maximum between u_h and u_ϵ through $\|u_h - u_\epsilon\|_{L^\infty} \approx \max_{\boldsymbol{x} \in [0,1]_h^2} |u_h(\boldsymbol{x}) - u_\epsilon(\boldsymbol{x})|$. Since we only consider h as integer multiples of ϵ, we have hat $[0,1]_h^2 \subset [0,1]_\epsilon^2$, meaning that this can be evaluated exactly.

The convergence plots can be seen in Figure 4. We seem to have numerical convergence for all variables. Among the semi-discretised schemes, (V_∞) performs the best for all test problems. Apart from test problem (C), which is to some extent less interesting anyways, the scheme (V_∞) also performs better than the discretised dynamic programming.

In [7], it was demonstrated that the schemes based on $D(u^2)$ have a higher numerical convergence rate than the schemes based on Du. At first glance, we do not seem to have this property. However, the difference between the schemes becomes apparent in test problem (B), where the convergence rate of the scheme (U_∞) flattens out for $h^{-1} \geq 10^3$. There are multiple factors contributing to the error of the schemes: the regularity of u (not being Lipschitz), the local variation of f and the number of shocks apparent in the value function. In [7], the problems considered were very regular, with little to no variation in f and at most one shock solution. The test problems (A) to (C) are substantially more complex, meaning that the error contributed from the lack of Lipschitz continuity of u is in most cases irrelevant.
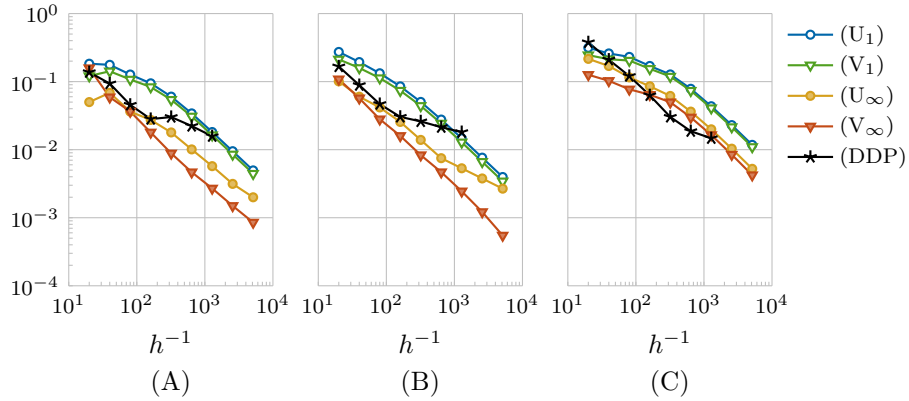

Figure 4: Convergence of u_h for test problems (A) to (C).

7.3 Convergence of the Geodesic Distance

By construction, we have that $J(\boldsymbol{\varphi}) = u(\mathbf{1})$ whenever $\boldsymbol{\varphi}$ is optimal. This gives us two ways to approximate the shape space distance:

$$\text{dist}^{\mathcal{S}}([c_1], [c_2]) \approx \arccos u_h(\mathbf{1}),$$
$$\text{dist}^{\mathcal{S}}([c_1], [c_2]) \approx \arccos J_h(\boldsymbol{\varphi}_h).$$

For the fully discretised schemes, these quantities are the same by construction of the scheme. For the semi-discretised schemes, however, these are different quantities and might have different convergence properties. The approximations were computed for each scheme and step size h. For test problems (A) and (B), we measured the error by comparison with $\arccos u_h(\mathbf{1})$. For test problem (C), we have the exact solution $\arccos u(\mathbf{1}) = \arccos J(\boldsymbol{\varphi}) = 0$. Convergence plots can be found in figures 5 and 6, respectively.

We seem to have numerical convergence for all methods considered. We observe some cancellation effects, particularly for schemes U_∞ and DDP for test problem (A) and V_∞ for test problem (B). The scheme V_∞ performs the best among the semi-discretised schemes while the fully discretised scheme has quite variable convergence properties. Generally, it is hard to determine the exact convergence properties as we are essentially solving a PDE, but only measure convergence of the solution at a single point. For test problem (C), we only seem to have an $\mathcal{O}(\sqrt{h})$ convergence rate for the semi-discretised schemes. This is due to the non-differentiability of $\arccos J$ at $J = 1$, which only occurs when the shape space distance is zero.[4]

For $J_h(\boldsymbol{\varphi}_h)$, all semi-discretised schemes perform almost identically. This might be due to the simple backtracking scheme we have proposed. Higher order backtracking schemes were tested without any significant improvement. For test problem (A), the convergence is too non-regular for a convergence rate to be estimated, for test problem (B), we seem to have a superlinear numerical convergence rate, and for test problem (C), we have a linear numerical convergence rate. Note also that apart from V_∞, the distance estimates based on $J_h(\boldsymbol{\varphi}_h)$ are more accurate than those based on $u_h(\mathbf{1})$. Finally, also for the distance estimate based on $u_h(\mathbf{1})$, we have worse convergence properties for test problem (C) compared to test problems (A) and (B). Again, this is explained by the non-differentiability of arccos. Consequently, we expect the schemes to perform worse for curves with equal shapes than for curves with non-zero shape space distance.

7.4 Convergence of the Geodesics

Although we have numerical convergence of the geodesic distance estimate, this need not imply numerical convergence of the geodesics. Therefore, we consider numerical convergence of the geodesics as well. Consider the two approximate geodesics

$$\gamma_h(\tau) = w_h(1 - \tau)q_{1,h} + w_h(\tau)q_{2,h},$$
$$\gamma_\epsilon(\tau) = w_\epsilon(1 - \tau)q_{1,\epsilon} + w_\epsilon(\tau)q_{2,\epsilon}.$$

---

[4]The function $\arccos J$ is also non-differentiable at $J = -1$. This value, however, can never occur as the solution of the optimisation problem.

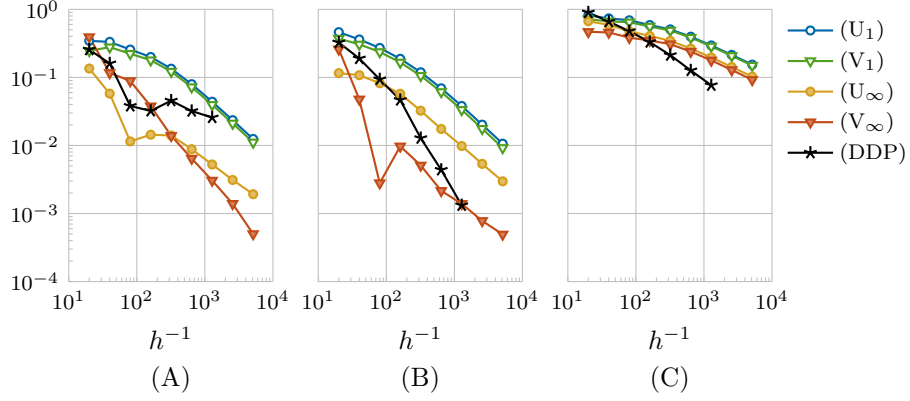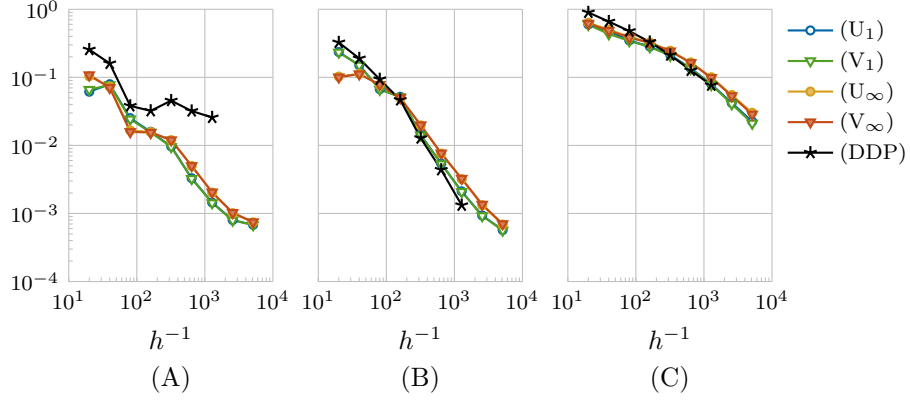Figure 5: Convergence of $\arccos u_h(\mathbf{1})$ for test problems (A) to (C).

Figure 6: Convergence of $\arccos J_h(\varphi_h)$ for test problems (A) to (C).

To measure the difference between these geodesics, we use the maximal pre-shape distance over τ. Since the unit sphere distance is at most π times larger than the L^2 distance, we have that

$$
\begin{aligned}
\max_{\tau \in [0,1]} \arccos \langle \gamma_h(\tau), \gamma_\epsilon(\tau) \rangle_{L^2} &\leq \max_{\tau \in [0,1]} \pi \| \gamma_h(\tau) - \gamma_\epsilon(\tau) \|_{L^2} \\
&= \pi \max \left\{ \| \gamma_h(0) - \gamma_\epsilon(0) \|_{L^2}, \ \| \gamma_h(1) - \gamma_\epsilon(1) \|_{L^2} \right\} \\
&= \pi \max \left\{ \| q_{1,h} - q_{1,\epsilon} \|_{L^2}, \ \| q_{2,h} - q_{2,\epsilon} \|_{L^2} \right\}.
\end{aligned}
$$

In other words, we can easily compute an upper bound to the maximal unit sphere distance between the geodesics. Note that it would be even better to use the maximal *shape space* distance between the geodesics. However, since the shape space distance requires the minimisation of the pre-shape distance, the upper bound is also an upper bound for the shape space distance.

From the convergence plots in Figure 7, we observe numerical convergence. Again, we have no observable difference between the semi-discretised schemes. However, in this case, the semi-discretised schemes perform better than the discretised dynamic programming.
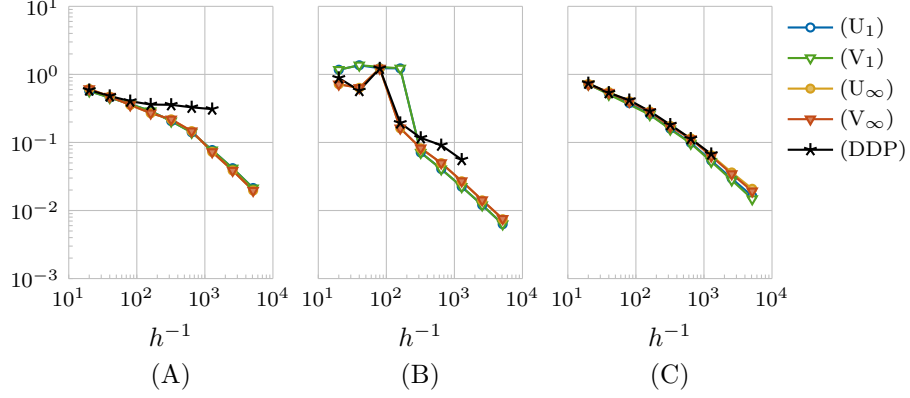
22

Figure 7: Convergence of γ_h for test problems (A) to (C).

7.5 Computational Complexity

Until now, we have evaluated performance in terms of error vs step size. However, there is a significant difference in computational complexity between the semi-discretised and the fully discretised methods. For the fully discretised dynamic programming scheme, the computational complexity is $\mathcal{O}(|A_h|N^2)$, while for the semi-discretised schemes, the computational complexity is $\mathcal{O}(N^2)$.
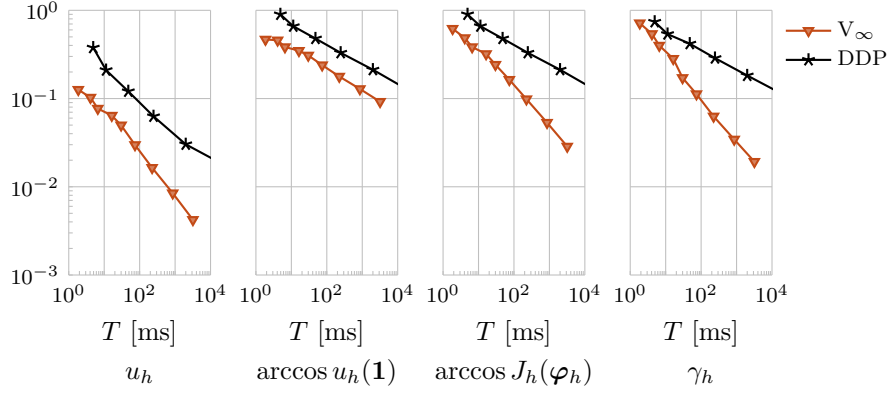

Figure 8: Work-precision diagrams for test problem (C). T denotes the computation time.

To implement the schemes, we used Python using NumPy with vectorised updates. Even though the implementation is to some extent naïve, it is useful to evaluate the accuracy of the schemes vs the computation times. From the three test problems, it is test problem (C) where the fully discretised scheme performs best compared to the semi-discretised schemes. Work-precision diagrams for this problem are visualised in Figure 8. As one can see, the semi-discretised scheme (V_∞) performs significantly better than the fully discretised method.

8 Conclusion

In this article, we have shown how PDE based method can be applied to the computation of shape space distances of open shapes. The method has global convergence and runs in $\mathcal{O}(N^2)$ time, which is strictly better than existing global solvers. Additionally, the numerical experiments indicate a linear convergence in practice, although we expect a lower theoretical convergence rate.

First, we presented a family of schemes which generalises the schemes of [8, 7]. These are based on the Hamilton-Jacobi-Bellman equation for the value function of the problem. However, whereas the schemes of [8, 7] approximate the gradient of the value function using finite difference approximations, we approximate its directional derivatives. This allows for greater flexibility in the construction of the schemes. The resulting family of schemes has theoretical convergence, and we show that two instances of the scheme are more accurate than previous approaches.

In conjunction with the schemes for the value function, we presented a backtracking scheme to obtain the solution of the reparametrisation problem. This is then used to estimate the shape space geodesics numerically. For different problems, the scheme seems to converge numerically, and the work-precision efficiency is better than that of previous global solvers.

From here, there is a number of interesting topics for future work, including the following:
- Assessment of the typical $\mathcal{O}(\sqrt{h})$ convergence rate for the HJB schemes, similar to [7, schemes S2, S3].
- Assessing theoretical convergence of the backtracking method.
- Constructing schemes such as (16) for general HJB equations.
- Construction of iterative solvers with adaptive grid refinement, where the HJB equation is solved on smaller and smaller strips around the solution of the reparametrisation problem, as has been done with great success for the fully discretised schemes [11, 4, 5].

References

[1] Guy Barles and Panagiotis E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis*, 4(4):271–283, 1991.

[2] Martin Bauer, Nicolas Charon, Eric Klassen, and Alice Le Brigant. Intrinsic Riemannian metrics on spaces of curves: theory and computation, 2020.

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

[4] Javier Bernal, Gunay Dogan, 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.

[5] Javier Bernal, James Lawrence, Gunay Dogan, and Robert Hagwood. On computing elastic shape distances between curves in d-dimensional space, 2021-02-20 2021.

[6] Martins Bruveris. Optimal reparametrizations in the square root velocity framework. *SIAM J. Math. Anal.*, 48(6):4335–4354, 2016.

[7] Jeff Calder. Numerical schemes and rates of convergence for the Hamilton-Jacobi equation continuum limit of nondominated sorting. *Numer. Math.*, 137(4):819–856, 2017.

[8] Jeff Calder, Selim Esedoglu, and Alfred Hero. A Hamilton–Jacobi equation for the continuum limit of nondominated sorting. *SIAM Journal on Mathematical Analysis*, 46, 02 2013.

[9] Elena Celledoni, Markus Eslitzbichler, and Alexander Schmeding. Shape analysis on Lie groups with applications in computer animation. *J. Geom. Mech.*, 8(3):273–304, 2016.

[10] Jean-Dominique Deuschel and Ofer Zeitouni. Limiting curves for i.i.d. records. *Ann. Probab.*, 23(2):852–878, 1995.

[11] Gunay Dogan, Javier Bernal, and Charles R. Hagwood. A fast algorithm for elastic shape distances between closed planar curves. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[12] J. Eckhardt, R. Hiptmair, T. Hohage, H. Schumacher, and M. Wardetzky. Elastic energy regularization for inverse obstacle scattering problems. *Inverse Problems*, 35(10):104009, 20, 2019.

[13] Wen Huang, Kyle A. Gallivan, Anuj Srivastava, and Pierre-Antoine Absil. Riemannian optimization for registration of curves in elastic shape analysis. *J. Math. Imaging Vision*, 54(3):320–343, 2016.

[14] Michael Kass Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[15] Wei Liu, Anuj Srivastava, and Jinfeng Zhang. A mathematical framework for protein structure comparison. *PLoS computational biology*, 7:e1001075, 02 2011.

[16] Washington Mio, Anuj Srivastava, and Shantanu Joshi. On Shape of Plane Elastic Curves. *International Journal of Computer Vision*, 73(3):307–324, 2007.

[17] 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*, 10 2010.

[18] Anuj Srivastava and Eric P. Klassen. *Functional and shape data analysis.* Springer Series in Statistics. Springer-Verlag, New York, 2016.

[19] Ganesh Sundaramoorthi, Andrea Mennucci, Stefano Soatto, and Anthony Yezzi. A new geometric metric in the space of curves, and applications to tracking deforming objects by prediction and filtering. *SIAM J. Imaging Sci.*, 4(1):109–145, 2011.

[20] Warut Thawinrak and Jeff Calder. High-order Filtered Schemes for the Hamilton-Jacobi Continuum Limit of Nondominated Sorting. *Journal of Mathematics Research*, 10, December 2017.

[21] Laurent Younes. Computable elastic distances between shapes. *SIAM J. Appl. Math.*, 58(2):565–586, 1998.

A   Approximating the SRVTs

The schemes presented in this article are based on exact computation of the forcing term $f(\boldsymbol{x})$, which in turns requires access to the SRVTs q_1 and q_2. Whenever these are not available, we can use finite difference approximations of the curves c_1 and c_2. Here, we suggest using backward differences of the form

$$q_i(t) \approx \frac{c_i(t) - c_i(t-h)}{\sqrt{h|c_i(t) - c_i(t-h)|}},$$

leading to the approximation

$$hf(\boldsymbol{x}) \approx \max\left\{ \left\langle \frac{c_1(x_1) - c_1(x_1-h)}{\sqrt{|c_1(x_1) - c_1(x_1-h)|}}, \frac{c_2(x_2) - c_2(x_2-h)}{\sqrt{|c_2(x_2) - c_2(x_2-h)|}} \right\rangle, \, 0 \right\}.$$

For the fully discretised schemes, we suggest using backwards differences of the form

$$q_i(t)\sqrt{k} \approx \frac{c_i(t) - c_i(t-kh)}{\sqrt{h|c_i(t) - c_i(t-kh)|}},$$

leading to the approximation

$$hf(\boldsymbol{x})\sqrt{kl} \approx \max\left\{ \left\langle \frac{c_1(x_1) - c_1(x_1-kh)}{\sqrt{|c_1(x_1) - c_1(x_1-kh)|}}, \frac{c_2(x_2) - c_2(x_2-lh)}{\sqrt{|c_2(x_2) - c_2(x_2-lh)|}} \right\rangle, \, 0 \right\}.$$

As long as the curves are immersions, i.e., that $|c_i'| > 0$ everywhere, these are consistent approximations, meaning that the proofs for convergence still hold. Moreover, we find that these approximations actually give better convergence properties for all implementations of the schemes.

B   Convergence for Fully Discretised Schemes

We express the scheme (24) as the solution of $S_h = 0$ with

$$S_h = \max_{\boldsymbol{\alpha} \in A_h} \frac{u(\boldsymbol{x} - h\boldsymbol{\alpha}) - u(\boldsymbol{x})}{h|\boldsymbol{\alpha}|} + f(\boldsymbol{x}) \frac{\sqrt{\alpha_1 \alpha_2}}{|\boldsymbol{\alpha}|}. \tag{25}$$

Assumption 3. A_h *satisfies the following:*
  (a) $A_h \subset \mathbb{N}_0^2 \setminus \{\boldsymbol{0}\}$.

(b) A_h *is finite for all* $h > 0$.

(c) $\lim_{h \to 0} \max_{\boldsymbol{\alpha} \in A_h} h|\boldsymbol{\alpha}| = 0$.

(d) *For every* $\boldsymbol{\beta} \in A_2$ *and* $\epsilon > 0$, *there exists* $h_0 > 0$ *such that for every* $0 < h \le h_0$, *there is* $\boldsymbol{\alpha} \in A_h$ *with* $|\boldsymbol{\alpha}/|\boldsymbol{\alpha}| - \boldsymbol{\beta}| < \epsilon$.

Theorem 5. *Under assumption 3, the scheme* (25) *is convergent.*

Proof. The scheme satisfies the following properties:

- *Monotonicity*: S_h is clearly non-decreasing in $u(\boldsymbol{y})$.
- *Stability*: For all grid points $\boldsymbol{x}$, there exists a grid point $\boldsymbol{y} < \boldsymbol{x}$, such that

$$u_h(\boldsymbol{y}) \le u_h(\boldsymbol{x}) = u_h(\boldsymbol{y}) + f(\boldsymbol{x})\sqrt{(x_1 - y_1)(x_2 - y_2)}$$
$$\le u_h(\boldsymbol{y}) + \|f\|_\infty \frac{1}{2}(x_1 - y_1 + x_2 - y_2)$$

using the Cauchy-Schwarz inequality. Inductively, this gives that $0 \le u(\boldsymbol{x}) \le \|f\|_\infty$.

- *Consistency*: We have that

$$S_h\big(\boldsymbol{y}, \psi(\boldsymbol{y}) + \xi, \psi + \xi\big) = \max_{\boldsymbol{\alpha} \in A_h} \frac{\psi(\boldsymbol{y} - h\boldsymbol{\alpha}) - \psi(\boldsymbol{y})}{h|\boldsymbol{\alpha}|} + f(\boldsymbol{y})\frac{\sqrt{\alpha_1 \alpha_2}}{|\boldsymbol{\alpha}|}$$
$$= \max_{\boldsymbol{\alpha} \in A_h} -D\psi(\boldsymbol{y})\frac{\boldsymbol{\alpha}}{|\boldsymbol{\alpha}|} + O(h|\boldsymbol{\alpha}|) + f(\boldsymbol{y})\frac{\sqrt{\alpha_1 \alpha_2}}{|\boldsymbol{\alpha}|}.$$

Hence, due to assumption 3c, we have that $\max_{\boldsymbol{\alpha} \in A_h} O(h|\boldsymbol{\alpha}|) = o(1)$. Moreover, we have that $D\psi$ and f is uniformly continuous in y. This, combined with assumption 3d, gives that

$$\lim_{\substack{h \to 0 \\ \boldsymbol{y} \to \boldsymbol{x} \\ \xi \to 0}} S_h\big(\boldsymbol{y}, \psi(\boldsymbol{y}) + \xi, \psi + \xi\big) = \max_{\boldsymbol{\alpha} \in A_2} -D\psi(\boldsymbol{x})\boldsymbol{\alpha} + f(\boldsymbol{x})\sqrt{\alpha_1 \alpha_2}$$

Due to [1, Theorem 2.1], this proves convergence. $\square$