A PDE-based Method for Shape Registration

Esten Nicolai Wien*† Markus Grasmair*‡

September, 2021

Abstract

In the square root velocity framework and similar approaches, the computation of shape space distances and the registration of curves requires the 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.

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. [22] for a classical model), or object tracking (see [33]), where one wants to follow the same object over a sequence of consecutive frames. Other examples include the analysis of shapes of proteins [26], modelling and analysis of computer animations [4, 11], or also inverse problems concerning the detection of shapes from indirect measurements [17].

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 [31, 28] (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 [32] for a detailed introduction into shape analysis within the SRV framework; a short overview can also be found in [3].

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 addition, new approaches based on deep learning have been suggested recently, and there exists an analytic solution algorithm for a certain class of curves.

In the dynamic programming approach introduced in [28] (see also [36] 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 φ_1'/φ_2' for the linear approximations of φ_1. In fact, a method with complexity $O(N^3)$ has already been proposed in [28]. Even more, a variant with complexity $O(N)$ has been presented in [5] (see also [16]), which is an iterative method 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 [21] 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 points. 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 [5]. 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.

Recently, deep learning based methods for the computation of shape distances have been suggested in [30, 20]. Once the networks have been trained, the resulting algorithms are significantly faster than both dynamic programming and gradient based approaches, and they do not suffer from issues with the initialisation of iterates. However, they are dependent on the availability of sufficiently rich training sets, and they do not come with any error estimates or convergence guarantees. Moreover, these methods are as of now limited to low dimensional situations: The method in [30] is limited to planar curves. The method in [20] is in principle able to deal with curves in arbitrary dimensions, but at the moment the results in higher dimensions are suffering from a lack of training data.

Finally, there exists an analytic algorithm for the case where both of the curves c_1 and c_2 are piecewise linear [25]. In this case, the optimal reparametrisation turns out to be piecewise linear as well, although typically with a different number of nodal points. Still, this makes it possible to compute the optimal reparametrisation exactly. Because of the high computational complexity that increases rapidly with the number of nodal points N, this algorithm is largely limited to applications where the exactness is crucial.

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* $u\colon I \times I \to \mathbb{R}$, 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 (18) below. This value function u has been shown in [10] 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 [9, 10, 34].

The main contribution of this paper is a generalisation of these schemes through a Semi-Lagrangian scheme 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.

Preliminaries

In this article, we discuss a PDE based numerical method for solving the problem

$$\sup_{\varphi_1,\varphi_2} \int_I \max\bigl\{\bigl\langle q_1\bigl(\varphi_1(t)\bigr), q_2\bigl(\varphi_2(t)\bigr)\bigr\rangle, 0\bigr\} \sqrt{\varphi_1'(t)\varphi_2'(t)}\, dt.$$

Here I is the unit interval, q_1, $q_2 \in C(I;\mathbb{R}^d)$ are given functions, and the supremum is taken over all absolutely continuous, non-decreasing functions $\varphi_i \colon I \to I$ with $\varphi_i(0) = 0$ and $\varphi_i(1) = 1$.

In this section, we will discuss how this problem appears naturally during the computation of various shape distances defined via curve transformations. We will start with a discussion of one particular curve transformation—the square-root-velocity-transform (SRVT)—and then see how the main idea can be extended to more general transforms.

The SRVT

Let $I = [0,1]$ be the unit interval and $d \in \mathbb{N}$, and denote by

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

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,

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

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,

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

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* $\operatorname{Imm}(I;\mathbb{R}^d)/G$, where G denotes the group of translations and scalings in $\mathbb{R}^d$.

In fact, one can show that it is possible to regard $\operatorname{Imm}(I;\mathbb{R}^d)/G$ as a Riemannian manifold for which dist is the geodesic distance. This also makes it possible to construct geodesics between curves: Consider c_1, $c_2 \in \operatorname{Imm}(I;\mathbb{R}^d)/G$ 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

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

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

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

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*

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

where

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

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

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

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

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

Thus we have that

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

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

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

with

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

In order to find an efficient numerical solution of the optimisation problem required for the computation of $\operatorname{dist}^{\mathcal{S}}$ in (6), we use we use a more explicit formulation: By applying the chain rule in (2), one obtains that

$$Q(c\circ\varphi)(t) = Q(c)(\varphi(t))\sqrt{\varphi'(t)}. \tag{7}$$

Thus, the optimisation problem (6) reads explicitly as

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

Since arccos is monotonically decreasing, we can alternatively compute

$$\operatorname{dist}^{\mathcal{S}}([c_1],[c_2]) = \arccos\Big(\sup_{\varphi_1,\,\varphi_2\in\mathrm{Diff}_+(I)} \int_I \Big\langle q_1\big(\varphi_1(t)\big)\sqrt{\varphi_1'(t)}, q_2\big(\varphi_2(t)\big)\sqrt{\varphi_2'(t)}\Big\rangle\, dt\Big).$$

General transforms

Many of the properties of the SRVT hinge on its behaviour under reparametrisations, that is, equation (7). Therefore it makes sense to consider general transformations

$$\begin{aligned} Q\colon\ \mathrm{Imm}(I;\mathbb{R}^d) &\to C(I;\mathbb{R}^{d'}),\\ c &\mapsto q := Q(c), \end{aligned} \tag{8}$$

that satisfy the condition

$$Q(c\circ\varphi)(t) = Q(c)(\varphi(t))\sqrt{\varphi'(t)} \tag{9}$$

for all $c\in\mathrm{Imm}(I;\mathbb{R}^d)$ and every $\varphi\in\mathrm{Diff}_+(I)$.

Specific examples of such transforms and distances that have been considered in the literature are the following:

- The *Square-Root-Velocity-Transform* (SRVT) as discussed above.
- The Q-transform, first introduced in [27], is defined as

  $$Q(c)(t) = \sqrt{|\dot c(t)|}c(t).$$

  Of particular interest is also a generalisation of this transform to surfaces (see [23, 24]), which, however, falls outside the scope of this article.

- In [29], the specific case of planar curves $d = 2$ was considered. After identifying $\mathbb{R}^2$ with $\mathbb{C}$, they define for fixed parameters a, $b > 0$ the transform
$$F_{a,b}(c)(t) = 2b\sqrt{|\dot{c}(t)|}\Big(\frac{\dot{c}(t)}{|\dot{c}(t)|}\Big)^{\frac{a}{2b}}.$$
Here $(s)^{\frac{a}{2b}}$ denotes the complex exponential of $s \in \mathbb{C} \setminus \{0\}$, or, more precisely, a suitable locally defined branch of the exponential. Details on the precise definition of this transform can be found in [29]. Note that the parameter choice $a = 2b$ recovers, up to scaling, the SRVT.
- A generalisation of the SRVT framework to Lie group valued curves was developed in [12]. Given a Lie group G with Lie algebra $\mathfrak{g}$, one can define the SRVT as a mapping from $\operatorname{Imm}(I; G)$ to $C(I; \mathfrak{g})$ by
$$Q(c)(t) = \frac{R^{-1}_{c(t)*}(\dot{c}(t))}{\sqrt{|\dot{c}(t)|}}.$$
Here R_g is the right translation by g on the Lie group G, and $R_{g*} = T_e R_g$ is the tangent map of R_g at the identity $e \in G$. Moreover, $|\cdot|$ is some norm on $\mathfrak{g}$ induced by a right invariant metric on G.

For further examples and a detailed discussion of such methods, we refer to [2].

Assume now that Q is some curve transformation satisfying (9). Given two curves c_1, $c_2 \in \operatorname{Imm}(I; \mathbb{R}^d)$ with corresponding transforms $q_i = Q(c_i)$, we can then define their Q-distance as
$$\operatorname{dist}(c_1, c_2) := \int_I |q_1(t) - q_2(t)|^2\, dt.$$
Moreover, we can define the shape space
$$\mathcal{S}(I; \mathbb{R}^d) := \operatorname{Imm}(I; \mathbb{R}^d)/(\sim \times \operatorname{Diff}_+(I)),$$
where $\sim$ is the equivalence relation
$$c_1 \sim c_2 :\iff Q(c_1) = Q(c_2).$$
In the cases of the SRVT and the $F_{a,b}$-transform, two curves are equivalent if and only if they are translates of each other; for the Q-transform the equivalence relation is trivial. Given two shapes $[c_1]$, $[c_2] \in \mathcal{S}(I; \mathbb{R}^d)$, we then define their distance as
$$\operatorname{dist}^{\mathcal{S}}([c_1], [c_2]) := \inf_{\varphi_1,\, \varphi_2 \in \operatorname{Diff}_+(I)} \operatorname{dist}(c_1 \circ \varphi_1, c_2 \circ \varphi_2). \tag{10}$$
With $q_i = Q(c_i)$ and using the property (7) of the transform Q, we can reformulate this as
$$\begin{aligned}\operatorname{dist}^{\mathcal{S}}([c_1], [c_2]) &= \inf_{\varphi_1,\, \varphi_2 \in \operatorname{Diff}_+(I)} \int_I \big|(q_1 \circ \varphi_1)\sqrt{\varphi_1'} - (q_2 \circ \varphi_2)\sqrt{\varphi_2'}\big|^2\, dt \\ &= \inf_{\varphi_1,\, \varphi_2 \in \operatorname{Diff}_+(I)} \int_I \big|(q_1 \circ \varphi_1)\big|^2 |\varphi_1'| + \big|(q_2 \circ \varphi_2)\big|^2 |\varphi_2'| \\ &\qquad - 2\big\langle q_1 \circ \varphi_1, q_2 \circ \varphi_2\big\rangle \sqrt{\varphi_1' \varphi_2'}\, dt \\ &= \|q_1\|^2_{L^2} + \|q_2\|^2_{L^2} - \sup_{\varphi_1,\, \varphi_2 \in \operatorname{Diff}_+(I)} \int_I 2\big\langle q_1 \circ \varphi_1, q_2 \circ \varphi_2\big\rangle \sqrt{\varphi_1' \varphi_2'}\, dt.\end{aligned} \tag{11}$$

As for the SRVT, it is also possible to define scaled variants of these general transforms, although the geometric meaning of the scaling may be less clear. There we define the scaled transform

$$\hat{Q}(c) = \frac{1}{\|Q(c)\|_{L^2}} Q(c),$$

which again maps $\mathrm{Imm}(I;\mathbb{R}^d)$ to the unit sphere in $L^2(I;\mathbb{R}^d)$. After defining a corresponding shape space and proceeding as for the SRVT above, we end up with the scaled shape distance

$$\mathrm{dist}^{\mathcal{S}}([c_1],[c_2]) = \arccos\left(\sup_{\varphi_1,\,\varphi_2 \in \mathrm{Diff}_+(I)} \int_I \bigl\langle (q_1 \circ \varphi_1), (q_2 \circ \varphi_2) \bigr\rangle \sqrt{\varphi_1' \varphi_2'},\, dt \right). \tag{12}$$

The variational problem

We now assume that q_1, $q_2 \colon I \to \mathbb{R}^d$ are given functions, and we define

$$J(\varphi_1,\varphi_2) := \int_I \bigl\langle q_1\bigl(\varphi_1(t)\bigr), q_2\bigl(\varphi_2(t)\bigr)\bigr\rangle \sqrt{\varphi_1'(t)\varphi_2'(t)}\, dt.$$

As discussed above, the main difficulty for the computation of both unscaled and scaled shape distances is the solution of the optimisation problem

$$\sup_{\varphi_1,\,\varphi_2 \in \mathrm{Diff}_+(I)} J(\varphi_1,\varphi_2). \tag{13}$$

In general, though, the supremum in (13) is not attained in $\mathrm{Diff}_+(I)$. However, it was shown in [7] that a relaxation of the optimisation problem (13) to a larger space of reparametrisations attains its maximum. Denote to that end

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

Lemma 1 (Bruveris 2016)**.** *Assume that q_1, $q_2 \in C(I;\mathbb{R}^d)$. Then*

$$\sup_{\varphi_1,\,\varphi_2 \in \mathrm{Diff}_+(I)} J(\varphi_1,\varphi_2) = \sup_{\varphi_1,\,\varphi_2 \in \Phi} J(\varphi_1,\varphi_2). \tag{14}$$

Moreover the optimisation problem

$$\sup_{\varphi_1,\,\varphi_2 \in \Phi} J(\varphi_1,\varphi_2) \tag{15}$$

attains a solution.

Proof. Equality (14) follows from the density of $\mathrm{Diff}_+(I)$ in $AC(I)$ together with the continuity of J with respect to the strong topology on $AC(I)$. The existence of a solution in $AC(I)$ has been shown in [7, Prop. 15] for the case where q_1 and q_2 are SRV transforms of functions in $C^1(I;\mathbb{R}^d)$. However, the proof immediately generalises to the case of arbitrary continuous functions q_i. □

In this paper we will discuss a solution method for (15) that is based on a reformulation as a Hamilton–Jacobi–Bellman equation. In order to be able apply this theory, it is necessary that the integrand in is concave with respect to the (φ_1', φ_2') variable. However, this is only the case at points where $\langle q_1(\varphi_1(t)), q_2(\varphi_2(t))\rangle \geq 0$. We therefore need one further modification of our problem. To that end, we define

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

where we set the integrand to zero whenever $\langle q_1(\varphi_1(t)), q_2(\varphi_2(t))\rangle$ is negative. Note that this is precisely the concave hull of the integrand with respect to the (φ_1', φ_2') variable.

The following result shows the relation between the two functionals J and J_c.

Lemma 2. *Let φ_1, $\varphi_2 \in \Phi$. Then there exist $\tilde{\varphi}_1$, $\tilde{\varphi}_2 \in \Phi$ such that*

$$J(\tilde{\varphi}_1, \tilde{\varphi}_2) = J_c(\varphi_1, \varphi_2).$$

Proof. See [7, Lemma 16]. □

Since $J_c \geq J$, this shows in particular that J and J_c have the same maximal value. Moreover, the proof in [7] is constructive in that it provides an explicit method for constructing the reparametrisations $\tilde{\varphi}_1$ and $\tilde{\varphi}_2$. This allows us to consider instead the problem of solving

$$\sup_{\varphi_1,\, \varphi_2 \in \Phi} J_c(\varphi_1, \varphi_2). \tag{16}$$

A General Variational Problem

The problem (16) can be seen as a special case of the variational problem

$$\sup_{\boldsymbol{\varphi} \in \mathcal{A}} \left(J(\boldsymbol{\varphi}) := \int_I f(\varphi_1(t), \varphi_2(t)) \sqrt{\varphi_1'(t)\varphi_2'(t)}\, dt \right). \tag{17}$$

Here, we denote $\boldsymbol{\varphi} = (\varphi_1, \varphi_2)$ and $\mathcal{A} = \Phi \times \Phi$. Moreover, $f\colon I \times I \to \mathbb{R}_{\geq 0}$ is a continuous, non-negative function. Although with a different motivation, this problem has been studied in [10, 9, 34, 15]; within the context of shape matching, a variant where φ_2 is set to be the identity has been discussed in [35]. In [10], a Hamilton-Jacobi-Bellman (HJB) formulation of this problem was derived, and HJB-based solvers were constructed in [10, 9, 34]. In the following, we recall the main HJB-related results from [10], and provide some useful generalisations of the results.

Dynamic Programming and the Value Function

For variational problems of the type (17), the solution can be described using dynamic programming. The starting point is the introduction of a value function $u : [0,1]^2 \to \mathbb{R}$ defined as

$$u(t, \boldsymbol{x}) := \sup_{\boldsymbol{\varphi} \in \mathcal{A}(t, \boldsymbol{x})} \int_0^t f(\varphi_1, \varphi_2) \sqrt{\varphi_1' \varphi_2'}\, dt \tag{18}$$

with

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

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

$$u(\boldsymbol{\varphi}(t)) \geq u(\boldsymbol{\varphi}(t-h)) + \int_{t-h}^{t} f(\varphi_1,\varphi_2)\sqrt{\varphi_1'\varphi_2'}\,dt, \tag{19}$$

for all $\boldsymbol{\varphi} \in \mathcal{A}$. Moreover, $\boldsymbol{\varphi}$ is a solution of (17) 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 (19) 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, we can always find a path $\boldsymbol{\varphi}$ from $\boldsymbol{y}$ to $\boldsymbol{x}$ for which $\sqrt{\varphi_1'\varphi_2'} > 0$ whenever $f(\boldsymbol{\varphi}) > 0$, which implies that $u(\boldsymbol{x}) > u(\boldsymbol{y})$. In particular, this implies that $u(\boldsymbol{x}) > 0$ if (and only if) there exists $\boldsymbol{y} < \boldsymbol{x}$ with $f(\boldsymbol{y}) > 0$.

Next it has been shown that $u(\boldsymbol{x})$ is Hölder continuous with exponent $\frac{1}{2}$ [10, Lemma 1] while $v(\boldsymbol{x}) := u(\boldsymbol{x})^2$ is Lipschitz continuous [9, Lemma 9]. Finally, we have the bound $0 \leq u(\boldsymbol{x}) \leq \|f\|_\infty \sqrt{x_1 x_2}$.

The HJB equation

For variational problems such as (17), 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

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

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

$$H(\boldsymbol{x}, \boldsymbol{p}) = \sup_{\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0}} -\boldsymbol{p} \cdot \boldsymbol{\alpha} + f(\boldsymbol{x})\sqrt{\alpha_1 \alpha_2}. \tag{20}$$

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

$$H(\boldsymbol{x}, \boldsymbol{p}) = \sup_{\boldsymbol{\alpha} \in A} -\max\{\boldsymbol{p} \cdot \boldsymbol{\alpha}, 0\} + f(\boldsymbol{x})\sqrt{\alpha_1 \alpha_2}. \tag{21}$$

Here, we have also followed the idea of [9] by replacing the inner product with its positive part in order to guarantee the uniqueness of the solution. In addition, this ensures that the viscosity subsolution property extends to the point $(1,1)$, and, since the value function already is monotonically non-decreasing, it will still be a viscosity solution of the HJB equation. This could also be fixed by changing the domain to $(0,1)^2$ as done in [13].

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 $\overline{\operatorname{cone}} \, A = \mathbb{R}^2_{\geq 0}$.
- 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 $\overline{\operatorname{cone}} \, A = \mathbb{R}^2_{\geq 0}$ 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

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

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

$$\begin{aligned} A &= \{\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0} \mid \alpha_1\alpha_2 = \text{const.}\}, \\ A &= \{\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0} \mid \alpha_1 = \text{const.}\}. \end{aligned}$$

The first choice is (implicitly) used in [10], 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 3. *Assume that A is such that assumption 1a holds. Then, the value function u is the unique viscosity solution of the hyperbolic PDE*

$$\begin{cases} H(\boldsymbol{x}, Du) = 0 \quad \textit{in } (0,1]^2, \\ u(0, x_2) = u(x_1, 0) = 0. \end{cases} \tag{22}$$

Proof. In [10, Theorems 2, 3], it has been shown that u is the unique viscosity solution of the PDE

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

where the Hamiltonian $\tilde{H}$ is given as

$$\tilde{H}(\boldsymbol{x}, \boldsymbol{p}) = -\max\{p_1, 0\}\max\{p_2, 0\} + \frac{1}{4}f(\boldsymbol{x})^2.$$

This Hamiltonian is equivalent in the viscosity sense to

$$H(\boldsymbol{x}, \boldsymbol{p}) = -\sqrt{\max\{p_1, 0\}\max\{p_2, 0\}} + \frac{1}{2}f(\boldsymbol{x}),$$

which is the explicit expression for (21) in the case where $A = \{\boldsymbol{\alpha} \in \mathbb{R}^2_{\geq 0} \mid \alpha_1\alpha_2 = \frac{1}{4}\}$. Moreover, as used in the proof of [10, Theorem 2], 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 1a. □

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 [9, 10, 34], but also as approximations to the dynamic programming principle (19).

Schemes based on $\boldsymbol{Du}$

We start by constructing numerical schemes for approximating solutions to the HJB equation. As in [9, 10, 34], we discretise the unit square into a square grid $[0,1]_h^2 := \{0, h, 2h, \dots, 1\}^2$. Here, we assume that $h = 1/N$, $N+1$ 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

$$\max_{\boldsymbol{\alpha}\in A} - \max\{D^- u(\boldsymbol{x})\boldsymbol{\alpha}, 0\} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} = 0. \tag{23}$$

Here we use the backward difference approximation

$$D^- u(\boldsymbol{x})\boldsymbol{\alpha} := \frac{u(\boldsymbol{x}) - u(\boldsymbol{x} - h\boldsymbol{\alpha})}{h}. \tag{24}$$

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 the approximation of u as $g_{\boldsymbol{x},h}[u]$. Typically, this will be a interpolation-type approximation. Inserting the approximation (24) into (23), this then gives the general scheme

$$\max_{\boldsymbol{\alpha}\in A} \frac{-\max\bigl\{u(\boldsymbol{x}) - g_{\boldsymbol{x},h}[u](\boldsymbol{x} - h\boldsymbol{\alpha}), 0\bigr\}}{h} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} = 0. \tag{25}$$

In the case where $u(\boldsymbol{x}) \geq g_{\boldsymbol{x},h}(\boldsymbol{x} - h\boldsymbol{\alpha})$ for all $\boldsymbol{\alpha} \in A$, this results, after rearranging the terms, in the expression

$$u(\boldsymbol{x}) = \max_{\boldsymbol{\alpha}\in A} g_{\boldsymbol{x},h}[u](\boldsymbol{x} - h\boldsymbol{\alpha}) + hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}. \tag{26}$$

Such schemes are typically referred to as Semi-Lagrangian schemes (see for example [18, 19, 14] for related Semi-Lagrangian schemes).

Semi-Lagrangian schemes can be seen as generalisations of the typical approach as in [9, 10, 34]. There $Du(\boldsymbol{x})$, interpreted as a gradient, is approximated numerically, and the directional derivative is then computed as the inner product of $\boldsymbol{\alpha}$ with the approximation to $Du(\boldsymbol{x})$. By using the approximation

$$g_{\boldsymbol{x},h}[u](\boldsymbol{x} - h\boldsymbol{\alpha}) = u(\boldsymbol{x}) - hD^- u(\boldsymbol{x}) \cdot \boldsymbol{\alpha}$$

for some approximate gradient $D^- u(\boldsymbol{x})$, this approach reduces to our setting.

In order to prove convergence of the schemes, we will use the classical proof of Barles & Souganidis for so-called monotone schemes (see [1]). Moreover, we will follow the standard notation for such schemes. To that end, we define the function $S_h \colon [0,1]^2 \times \mathbb{R} \times \mathbb{R}^{[0,1]_h^2} \to \mathbb{R}$,

$$S_h(\boldsymbol{x}, t, u) = \max_{\boldsymbol{\alpha}\in A} \frac{-\max\bigl\{t - g_{\boldsymbol{x},h}[u](\boldsymbol{x} - h\boldsymbol{\alpha}), 0\bigr\}}{h} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}.$$

The approximation $u_h(\boldsymbol{x})$ is then defined as the largest solution $t \in \mathbb{R}$ of the equation $S_h(\boldsymbol{x}, t, u_h) = 0$. We want to use [1, Theorem 2.1], which states that a scheme that is *monotone*, *stable* and *consistent* is also convergent. Here, we define:

- *Monotonicity*: S_h is non-decreasing in u.
- *Stability*: The scheme $S_h\bigl(\boldsymbol{x}, u_h(\boldsymbol{x}), u_h\bigr) = 0$ has a solution u_h which is bounded in the sense that $\|u_h\|_\infty \leq$ const. independent of h. Additionally, the boundary constraints $u_h(x_1, 0) = u_h(0, x_2) = 0$ also hold in the limit as $h \to 0$.

- *Consistency*: For every $\psi \in C^\infty$, we have that

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

Here we identify a function $\psi \in C^\infty$ with its evaluation on the grid $[0,1]^h$.

Given A, the scheme is completely determined by $g_{\boldsymbol{x},h}$. Accordingly, monotonicity, stability and consistency of the scheme can be inferred from the properties of $g_{\boldsymbol{x},h}$. In the following, define

$$B^-_{Ch}(\boldsymbol{x}) := \{\boldsymbol{y} \in [0,1]^2 \mid \boldsymbol{y} < \boldsymbol{x},\ |\boldsymbol{y} - \boldsymbol{x}| \le Ch\}.$$

Assumption 2. *For every $\boldsymbol{y} = \boldsymbol{x} - h\boldsymbol{\alpha}$ for $\boldsymbol{\alpha} \in A$ and $\boldsymbol{x} \in [0,1]^2_h$, the approximation $g_{\boldsymbol{x},h}$ satisfies the following properties:*

(a) Monotonicity: $g_{\boldsymbol{x},h}[u](\boldsymbol{x} - h\boldsymbol{\alpha})$ is non-decreasing in u.

(b) Localisation: there exists $C > 0$ such that for all functions ψ, ξ for which $\psi = \xi$ on $B^-_{Ch}(\boldsymbol{x})$, we have that $g_{\boldsymbol{x},h}[\psi](\boldsymbol{y}) = g_{\boldsymbol{x},h}[\xi](\boldsymbol{y})$.

(c) Boundedness: if $0 \le u(\boldsymbol{z}) \le C\sqrt{z_1 z_2}$ for all $\boldsymbol{z} \in (0,1)^2$ and some $C \ge 0$, then it holds that $0 \le g_{\boldsymbol{x},h}[u](\boldsymbol{y}) \le C\sqrt{y_1 y_2}$.

(d) Superlinear consistency: for every $\epsilon > 0$ and $\psi \in C^\infty[0,1]^2$, there exists a modulus of continuity $\omega_{\psi,\epsilon}$, such that

$$\left|\frac{g_{\boldsymbol{x},h}[\psi + t](\boldsymbol{y}) - \psi(\boldsymbol{y}) - t}{h}\right| \le \omega_{\psi,\epsilon}(h)$$

for every $|t| \le \epsilon$.

Assumption 2a ensures that the scheme is monotone, assumption 2c ensures that the scheme is bounded and that the boundary conditions hold in the limit as $h \to 0$, and assumption 2d ensures that the scheme is consistent. assumption 2b ensures that the scheme is explicit in that the approximation $u_h(\boldsymbol{x})$ only depends of values $u_h(\boldsymbol{y})$ with $\boldsymbol{y} < \boldsymbol{x}$. The locality, although a natural property of many numerical solvers, is actually not required for the convergence of the scheme. We will, however, make use of this condition for the type of schemes discussed in Section 4.2. Note moreover that, in view of the monotonicity, assumption 2c needs only hold for the function $u(\boldsymbol{z}) = C\sqrt{z_1 z_2}$.

Theorem 4. *Under assumptions 1 and 2, the scheme* (25) *is convergent.*

Proof. The scheme satisfies the following properties:

- *Monotonicity*: S_h is increasing in $g_{\boldsymbol{x},h}$ and $g_{\boldsymbol{x},h}$ is non-decreasing in u. Hence S_h is non-decreasing in u.
- *Stability*: Let $\boldsymbol{x} \in (0,1]^2$. Due to assumption 2b, the approximation $g_{\boldsymbol{x},h}$ only depends on points $\boldsymbol{y} < \boldsymbol{x}$, and therefore the scheme is explicit. Moreover, we obtain the explicit expression

$$u_h(\boldsymbol{x}) = \max_{\boldsymbol{\alpha}\in A} g_{\boldsymbol{x},h}[u_h](\boldsymbol{y} - h\boldsymbol{\alpha}) + hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}.$$

We now want to show that $0 \le u_h(\boldsymbol{x}) \le \|f\|_\infty\sqrt{x_1 x_2}$ for all $\boldsymbol{x}$. Assume to that end that $0 \le u_h(\boldsymbol{y}) \le \|f\|_\infty\sqrt{y_1 y_2}$ for all $\boldsymbol{y} < \boldsymbol{x}$. Then, using assumptions 2b and 2c, we obtain that

$$\begin{aligned}
0 \le u_h(\boldsymbol{x}) &\le \max_{\boldsymbol{\alpha}\in A} g_{\boldsymbol{x},h}[u_h](\boldsymbol{y} - h\boldsymbol{\alpha}) + hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} \\
&\le \max_{\boldsymbol{\alpha}\in A} \|f\|_\infty\sqrt{(x_1 - h\alpha_1)(x_2 - h\alpha_2)} + h\|f\|_\infty\sqrt{\alpha_1\alpha_2} \\
&\le \|f\|_\infty\sqrt{x_1 x_2}.
\end{aligned}$$

Since this holds on the initial boundary $\{\boldsymbol{x} \mid x_1 x_2 = 0\}$, the estimates

$$0 \le u_h(\boldsymbol{x}) \le \|f\|_\infty \sqrt{x_1 x_2},$$

hold by induction. These estimates hold for all $h > 0$ and therefore also in the limit as $h \to 0$. This means that the limit of solution of the scheme does actually satisfy the boundary condition $u = 0$ where $x_1 = 0$ or $x_2 = 0$ since the upper bound $\|f\|_\infty \sqrt{x_1 x_2}$ evaluates to zero at this boundary.

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

$$\begin{aligned}
&\lim_{\substack{h\to 0\\ t\to 0\\ \boldsymbol{y}\to\boldsymbol{x}}} S_h\big(\boldsymbol{y}, \psi(\boldsymbol{y}) + \xi, \psi + t\big) \\
&= \lim_{\substack{h\to 0\\ t\to 0\\ \boldsymbol{y}\to\boldsymbol{x}}} \max_{\boldsymbol{\alpha}\in A} \frac{\max\{g_{\boldsymbol{x},h}[\psi + t](\boldsymbol{y} - h\boldsymbol{\alpha}) - \psi(\boldsymbol{y}) - t, 0\}}{h} + f(\boldsymbol{y})\sqrt{\alpha_1\alpha_2} \\
&= \lim_{\substack{h\to 0\\ t\to 0\\ \boldsymbol{y}\to\boldsymbol{x}}} \max_{\boldsymbol{\alpha}\in A} \max\left\{\frac{\psi(\boldsymbol{y} - h\boldsymbol{\alpha}) - \psi(\boldsymbol{y})}{h} + \omega_{\psi,\epsilon}(h), 0\right\} + f(\boldsymbol{y})\sqrt{\alpha_1\alpha_2} \\
&\overset{(*)}{=} \max_{\boldsymbol{\alpha}\in A} \lim_{\substack{h\to 0\\ t\to 0\\ \boldsymbol{y}\to\boldsymbol{x}}} \frac{\max\{\psi(\boldsymbol{y} - h\boldsymbol{\alpha}) - \psi(\boldsymbol{y}), 0\}}{h} + \omega_{\psi,\epsilon}(h) + f(\boldsymbol{y})\sqrt{\alpha_1\alpha_2} \\
&= \max_{\boldsymbol{\alpha}\in A} \max\{-D\psi(\boldsymbol{x})(\boldsymbol{\alpha}), 0\} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} \\
&= H(\boldsymbol{x}, D\psi(\boldsymbol{x})).
\end{aligned}$$

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

Due to [1, Theorem 2.1], this proves convergence. □

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 [9], where schemes are constructed for $v := u^2$.[1]

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

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

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

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

We define therefore

[1] Additionally, a scheme for the Lipschitz continuous term $w := u/\sqrt{x_1 x_2}$ is also constructed in [9]. However, we deem this idea ill-suited for our approach due to the lack of simple closed-form expressions.

$$S_h(\boldsymbol{x},t,u) = \begin{cases} \max\limits_{\boldsymbol{\alpha}\in A} \dfrac{-\max\{t^2 - g_{\boldsymbol{x},h}[u^2](\boldsymbol{x}-h\boldsymbol{\alpha}), 0\}}{2th} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} & \text{if } t > 0, \\ f(\boldsymbol{x})\sqrt{\alpha_2\alpha_2} & \text{if } t \leq 0, \end{cases} \tag{27}$$

and consider the scheme $S_h(\boldsymbol{x}, u_h(\boldsymbol{x}), u_h) = 0$. Again, we define the approximation $u_h(\boldsymbol{x})$ to be the largest solution of this equation.

In the following, we will derive an explicit expression for $u_h(\boldsymbol{x})$. To that end note first that we will always have that $u_h(\boldsymbol{x}) \geq 0$, and $u_h(\boldsymbol{x}) = 0$ if and only if $f(\boldsymbol{x}) = 0$ and $g_{\boldsymbol{x},h}[u^2](\boldsymbol{x} - h\boldsymbol{\alpha}) \leq 0$ for all $\boldsymbol{\alpha} \in A$.

Assume now that this is not the case and thus $u_h(\boldsymbol{x}) > 0$. Then we can multiply the equation $S_h(\boldsymbol{x}, u_h(\boldsymbol{x}), u_h) = 0$ by $2u_h(\boldsymbol{x})h$ and obtain the equation

$$\max_{\boldsymbol{\alpha}\in A} -\max\{u_h(\boldsymbol{x})^2 - g_{\boldsymbol{x},h}[u_h^2](\boldsymbol{x}-h\boldsymbol{\alpha}), 0\} + 2u_h(\boldsymbol{x})hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} = 0. \tag{28}$$

For fixed $\boldsymbol{\alpha} \in A$, the sign of the argument does not change if we replace $u_h(\boldsymbol{x})^2$ by $\max\{u_h(\boldsymbol{x})^2, g_{x,h}[u^2](\boldsymbol{x}-h\boldsymbol{\alpha})\}$. Since we are interested in the largest solution of the equation, it follows that (28) is equivalent to

$$\max_{\boldsymbol{\alpha}\in A} -u_h(\boldsymbol{x})^2 + g_{\boldsymbol{x},h}[u_h^2](\boldsymbol{x}-h\boldsymbol{\alpha}) + 2u_h(\boldsymbol{x})hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} = 0. \tag{29}$$

Now we can rewrite (29) by completing the square w.r.t $u_h(\boldsymbol{x})$ to obtain that

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in A}\Big[g_{\boldsymbol{x},h}[u_h^2](\boldsymbol{x}-h\boldsymbol{\alpha}) - u_h(\boldsymbol{x})^2 + 2u_h(\boldsymbol{x})hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}\Big] \\ &\quad = \max_{\boldsymbol{\alpha}\in A}\Big[-\Big(\underbrace{u_h(\boldsymbol{x}) - hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}}_{=:F(\boldsymbol{\alpha})}\Big)^2 + \underbrace{h^2 f(\boldsymbol{x})^2\alpha_1\alpha_2 + g_{\boldsymbol{x},h}[u_h^2](\boldsymbol{x}-h\boldsymbol{\alpha})}_{=:G(\boldsymbol{\alpha})^2}\Big] \\ &\qquad\qquad = \max_{\boldsymbol{\alpha}\in A}\big[-F(\boldsymbol{\alpha})^2 + G(\boldsymbol{\alpha})^2\big] = 0, \end{aligned}$$

where $F(\boldsymbol{x}), G(\boldsymbol{x}) \geq 0$. We must 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. From $F(\boldsymbol{x}), G(\boldsymbol{y}) \geq 0$, it then follows that $-F(\boldsymbol{x}) + G(\boldsymbol{x}) \leq 0$ with equality whenever $\boldsymbol{\alpha}$ is optimal. Accordingly, the above scheme is identical to

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in A} -F(\boldsymbol{\alpha}) + G(\boldsymbol{\alpha}) \\ &\quad = \max_{\boldsymbol{\alpha}\in A} -u_h(\boldsymbol{x}) + hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} + \sqrt{h^2 f(\boldsymbol{x})^2\alpha_1\alpha_2 + g_{\boldsymbol{x},h}[u_h^2](\boldsymbol{x}-h\boldsymbol{\alpha})} = 0, \end{aligned}$$

which gives the closed form expression

$$u_h(\boldsymbol{x}) = \max_{\boldsymbol{\alpha}\in A} hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} + \sqrt{h^2 f(\boldsymbol{x})^2\alpha_1\alpha_2 + g_{\boldsymbol{x},h}[u_h^2](\boldsymbol{x}-h\boldsymbol{\alpha})}. \tag{30}$$

In particular, we have that $u_h(\boldsymbol{x}) \geq hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2}$ for all $\boldsymbol{\alpha} \in A$.

Assumption 2*. *The approximation $g_{\boldsymbol{x},h}$ satisfies assumptions 2a, 2b and 2d and in addition:*

(c) If* $0 \le u(\boldsymbol{z}) \le C\sqrt{z_1 z_2}$ *for all* $\boldsymbol{z} \in (0,1)^2$ *and some* $C \ge 0$*, then it holds that* $0 \le g_{\boldsymbol{x},h}[u^2](\boldsymbol{y}) \le C^2 y_1 y_2$.

Theorem 5. *Under assumptions 1 and 2*, the scheme* (27) *is convergent.*

Proof. As in the proof of Theorem 4, we prove monotonicity, stability and consistency separately.

Monotonicity is similar to the proof of Theorem 4.

Stability is similar to the proof of Theorem 4, with some exceptions: we now assume that $u_h(\boldsymbol{y}) \le 2\|f\|_\infty \sqrt{y_1 y_2}$ for all $\boldsymbol{y} < \boldsymbol{x}$, to obtain the estimates that

$$\begin{aligned} 0 \le u_h(\boldsymbol{x}) &= \max_{\boldsymbol{\alpha}\in A} hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} + \sqrt{hf(\boldsymbol{x})^2\alpha_1\alpha_2 + g_{\boldsymbol{x},h}[u_h^2](\boldsymbol{x} - h\boldsymbol{\alpha})} \\ &\le \max_{\boldsymbol{\alpha}\in A} 2hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} + \sqrt{g_{\boldsymbol{x},h}[u_h^2](\boldsymbol{x} - h\boldsymbol{\alpha})} \\ &\le \max_{\boldsymbol{\alpha}\in A} 2h\|f\|_\infty\sqrt{\alpha_1\alpha_2} + 2\|f\|_\infty\sqrt{(x_1 - h\alpha_1)(x_2 - h\alpha_2)} \\ &= 2\|f\|_\infty\sqrt{x_1 x_2}. \end{aligned}$$

This proves stability with a similar induction argument.

Consistency requires us to consider the cases $u(\boldsymbol{x}) = 0$ and $u(\boldsymbol{x}) > 0$ separately. 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 case with $\underline{u}(\boldsymbol{x}) > 0$ corresponds to comparison with $\psi(\boldsymbol{x}) > 0$ in [1, Theorem 2.1]. In this case, the proof of consistency as in Theorem 4 still works using the scheme (27). Wherever $\underline{u}(\boldsymbol{x}) = 0$, we do not have consistency due to the singularity of the scheme at this point. However, we have that the exact value function satisfies $u(\boldsymbol{x}) = 0$ if and only if $f(\boldsymbol{y}) = 0$ for all $\boldsymbol{y} \le \boldsymbol{x}$. Accordingly, it is sufficient to prove that this property holds for $\underline{u}$ and $\overline{u}$ as well.

Case 1: $f(\boldsymbol{y}) = 0$ for all $\boldsymbol{y} \le \boldsymbol{x}$. Then it easily follows by induction that $u_h(\boldsymbol{y}) = 0$ for all $\boldsymbol{y} \le \boldsymbol{x}$: Indeed, assume that, for some $\boldsymbol{y} \le \boldsymbol{x}$ we have that $u_h(\boldsymbol{z}) = 0$ for all $\boldsymbol{z} < \boldsymbol{y}$. Then we can use assumption 2c with $C = 0$, which implies that $g_{\boldsymbol{y},h}[u_h^2](\boldsymbol{y} - h\alpha) = 0$. Inserting this into (30) gives $u_h(\boldsymbol{y}) = 0$ for all h. Inductively, this gives that $u_h(\boldsymbol{x}) = 0$ for all h.

Case 2: Assume that $f(\boldsymbol{y}) > 0$ for some $\boldsymbol{y} < \boldsymbol{x}$. Then, there is a ball $B_K^-(\boldsymbol{y})$ such that $f(\boldsymbol{y}) \ge f_0 > 0$ on $B_K^-(\boldsymbol{y})$. Now let $h > 0$ be fixed and consider the nested balls $B_{mCh}^-(\boldsymbol{y})$ for $m = 1, 2, \ldots$, where C defines the locality of the scheme (see assumption 2b). The nested balls have two important properties:

- For every $0 \le m \le M := \lfloor K/(Ch) \rfloor$, we have that $B_{mCh}^-(\boldsymbol{y}) \subset B_K^-(\boldsymbol{y})$.
- For every $\boldsymbol{z} \in B_{(m-1)Ch}^-(\boldsymbol{y})$, we have that $B_{Ch}^-(\boldsymbol{z}) \subset B_{mCh}^-(\boldsymbol{y})$.

See Figure 1 for an illustration of the balls of interest.

We will show by induction that

$$u_h \ge \frac{1}{2}h(M - m)f_0 A_{\min} \qquad \text{on } B_{mCh}^-(\boldsymbol{y}), \tag{31}$$

where $A_{\min} := \min_{\boldsymbol{\alpha}\in A} \sqrt{\alpha_1\alpha_2}$. Since $u_h \ge 0$ by construction, this trivially holds for $m = M$.

Assume now that (31) holds for some $0 < m \le M$. Moreover, assume without loss of generality that h is sufficiently small such that $\omega_{0,\epsilon}(h) \le \frac{1}{2}hf_0 A_{\min}$

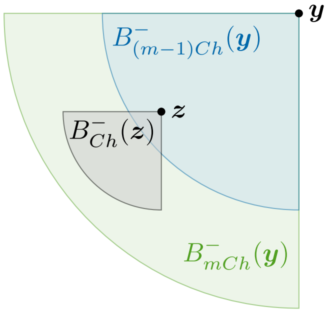
$\boldsymbol{y}$
$B^-_{(m-1)Ch}(\boldsymbol{y})$
$\boldsymbol{z}$
$B^-_{Ch}(\boldsymbol{z})$
$B^-_{mCh}(\boldsymbol{y})$

Figure 1: The balls used in the proof of Theorem 5.

where $\epsilon = (Mf_0A_{\min})^2$. Here, $\omega_{0,\epsilon}$ is the modulus of continuity for the constant function $\psi = 0$ as in assumption 2d.

For every $\boldsymbol{z} \in B^-_{(m-1)Ch}(\boldsymbol{y})$, we have that

$$\begin{aligned}
u_h(\boldsymbol{z}) &\geq \max_{\boldsymbol{\alpha}\in A} hf(\boldsymbol{z})\sqrt{\alpha_1\alpha_2} + \sqrt{g_{\boldsymbol{z},h}[u_h^2](\boldsymbol{z} - h\boldsymbol{\alpha})} \\
&\geq \max_{\boldsymbol{\alpha}\in A} hf(\boldsymbol{z})\sqrt{\alpha_1\alpha_2} + \sqrt{g_{\boldsymbol{z},h}\left[\left(\tfrac{1}{2}h(M-m)f_0A_{\min}\right)^2\right](\boldsymbol{z} - h\boldsymbol{\alpha})} \\
&\geq \max_{\boldsymbol{\alpha}\in A} hf(\boldsymbol{z})\sqrt{\alpha_1\alpha_2} + \sqrt{\left(\tfrac{1}{2}h(M-m)f_0A_{\min}\right)^2 - \omega_{0,\epsilon}(h)} \\
&\geq hf_0A_{\min} + \tfrac{1}{2}h(M-m)f_0A_{\min} - \omega_{0,\epsilon}(h) \\
&\geq hf_0A_{\min} + \tfrac{1}{2}h(M-m)f_0A_{\min} - \tfrac{1}{2}hf_0A_{\min} \\
&\geq \tfrac{1}{2}h(M-(m-1))f_0A_{\min}.
\end{aligned}$$

In the second inequality, we have used the locality of the approximation together with the induction assumption. In the third inequality, we have used the superlinear consistency assumption 2d with $\psi = 0$ and $t = \left(\frac{1}{2}h(M-m)f_0A_{\min}\right)^2$.

By induction, we therefore obtain (31), and in particular that $u_h(\boldsymbol{y}) \geq \frac{1}{2}hMf_0A_{\min}$. Since $hM = \lfloor K/(Ch)\rfloor \geq \frac{K}{2C}$ for sufficiently small h, it follows that $u_h(\boldsymbol{y}) \geq \frac{KA_{\min}}{4C}$ for all sufficiently small h.

With a similar argumentation, and using the non-negativity of f, we can now show that $u_h(\boldsymbol{x}) \geq \frac{KA_{\min}}{8C}$ for all sufficiently small h. This implies that $\underline{u}(\boldsymbol{x}) \geq \frac{KA_{\min}}{8C} > 0$, which concludes the proof. □

Proposed Schemes

Now, it remains to choose A and $g_{\boldsymbol{x},h}$ so 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 $\boldsymbol{x}_1^1 = \boldsymbol{x}$ as the current grid point for which we are solving the schemes. In addition, let $\boldsymbol{x}_1^0 = \boldsymbol{x} - h\boldsymbol{e}_2$, $\boldsymbol{x}_0^1 = \boldsymbol{x} - h\boldsymbol{e}_1$ and $\boldsymbol{x}_0^0 = \boldsymbol{x} - h\boldsymbol{e}_1 - h\boldsymbol{e}_2$ denote the other three corners of the associated grid cell. Figure 2 illustrates these grid notes in relation to each other. Lastly, denote $u_i^j = u(\boldsymbol{x}_i^j)$ and $f_i^j = f(\boldsymbol{x}_i^j)$. Until now, we have

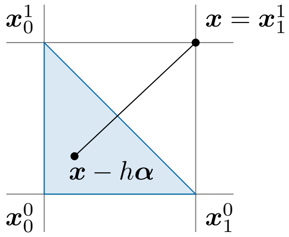
$\boldsymbol{x}_0^1$
$\boldsymbol{x} = \boldsymbol{x}_1^1$
$\boldsymbol{x} - h\boldsymbol{\alpha}$
$\boldsymbol{x}_0^0$
$\boldsymbol{x}_1^0$

Figure 2: The stencil illustrates the approximation of $Du(\boldsymbol{x})(\boldsymbol{\alpha})$. The blue triangle illustrates the interpolation region used in the approximation $g_{\boldsymbol{x},h}(\boldsymbol{x}, \boldsymbol{\alpha}, u) \approx u(\boldsymbol{x} - h\boldsymbol{\alpha})$.

assumed that f_i^j is evaluated exactly. For an approximation of this term, see appendix A.

We start by letting $g_{\boldsymbol{x},h}$ be the linear interpolation of u through the points $\boldsymbol{x}_0^1$, $\boldsymbol{x}_1^0$ and $\boldsymbol{x}_0^0$, which reads

$$\begin{aligned} g_{\boldsymbol{x},h}[u](\boldsymbol{x} - h\boldsymbol{\alpha}) &= u_0^0 + (1-\alpha_1)(u_1^0 - u_0^0) + (1-\alpha_2)(u_0^1 - u_0^0) \\ &= (\alpha_1 + \alpha_2 - 1)u_0^0 + (1-\alpha_1)u_1^0 + (1-\alpha_2)u_0^1. \end{aligned} \tag{32}$$

Since we base the approximation on a linear interpolation of u on a triangle, we want to choose A such that $\boldsymbol{x} - h\boldsymbol{\alpha}$ is inside this triangle, formally speaking (see Figure 2). This holds whenever both $0 \le \alpha_1, \alpha_2 \le 1$ and $\alpha_1 + \alpha_2 \ge 1$. For instance, each $\alpha \in A_r := \{\boldsymbol{\alpha} \in \mathbb{R}^2_{\ge 0} \mid |\boldsymbol{\alpha}|_r = 1\}$ for $1 \le r \le +\infty$ satisfies this property.

Lemma 6. *Assume that A satisfies assumption 1 and that $0 \le \alpha_1, \alpha_2 \le 1$ and $\alpha_1 + \alpha_2 \ge 0$. Then, the approximation* (32) *satisfies assumptions 2 and 2*, meaning that the resulting schemes* (25) *and* (27) *are convergent.*

Proof. Assumptions 2a, 2b and 2d are easy to prove: The coefficients of the terms u_i^j are always positive, implying monotonicity. Localisation follows directly with for example $C = 2$. A linear interpolation is of quadratic order, therefore also superlinear and consistent.

The approximation (32) is a linear interpolation. Since the function $\boldsymbol{x} \mapsto C\sqrt{x_1 x_2}$ is concave, we immediately obtain that $g_{\boldsymbol{x},h}[C\sqrt{y_1 y_2}](\boldsymbol{y}) \le C\sqrt{y_1 y_2}$ for all $\boldsymbol{y} = \boldsymbol{x} - h\boldsymbol{\alpha}$. Hence assumption 2c holds.

Assumption 2c* can be seen algebraically. Let $u(\boldsymbol{x}) = C^2 x_1 x_2$. Then, we have that

$$\begin{aligned} u_0^0 &= C^2(x_1 - h)(x_2 - h), \\ u_0^1 &= C^2 x_1 (x_2 - h), \\ u_1^0 &= C^2 (x_1 - h) x_2. \end{aligned}$$

Inserting these terms into (32), we obtain after some simplifications that

$$\begin{aligned} g_{\boldsymbol{x},h}[C^2 y_1 y_2](\boldsymbol{x} - h\boldsymbol{\alpha}) &= C^2(x_1 - h\alpha_1)(x_2 - h\alpha_2) - C^2 h^2 (1-\alpha_1)(1-\alpha_2) \\ &\le C^2(x_1 - h\alpha_2)(x_2 - h\alpha_2) \end{aligned}$$

as desired. □

We find that for the choices $A = A_1$ and $A = A_\infty$, we can solve the schemes analytically.

Schemes based on $\boldsymbol{Du}$

Choosing $A = A_1$, we obtain the following solution for the scheme, and the optimal $\boldsymbol{\alpha}^*$:

$$
\begin{aligned}
hS_h &= u_1^1 - \frac{1}{2}\left(u_0^1 + u_1^0 + \sqrt{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\right), \\
u_1^1 &= \frac{1}{2}\left(u_0^1 + u_1^0 + \sqrt{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\right), \\
\boldsymbol{\alpha}^* &= \left(\frac{1}{2}\Big(1 + \frac{u_0^1 - u_1^0}{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\Big), \frac{1}{2}\Big(1 - \frac{u_0^1 - u_1^0}{(u_0^1 - u_1^0)^2 + (hf_1^1)^2}\Big)\right).
\end{aligned}
\tag{U$_1$}
$$

Interestingly, this is exactly the original scheme proposed in [10]. Using $A = A_\infty$, we obtain with the abbreviation

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

that

$$
\begin{aligned}
hS_h &= u_1^1 - \begin{cases} u_*^* + \dfrac{(hf_1^1)^2}{4(u_*^* - u_0^0)}, & 2(u_*^* - u_0^0) > \sqrt{(hf_1^1)^2}, \\ u_0^0 + hf_1^1, & \text{otherwise}, \end{cases} \\
u_1^1 &= \begin{cases} u_*^* + \dfrac{(hf_1^1)^2}{4(u_*^* - u_0^0)}, & 2(u_*^* - u_0^0) > \sqrt{(hf_1^1)^2}, \\ u_0^0 + hf_1^1, & \text{otherwise}, \end{cases} \\
\boldsymbol{\alpha}^* &= \begin{cases} \left(1, \dfrac{hf_1^1}{2(u_0^1 - u_0^0)}\right), & u_0^1 \geq u_1^0, 2(u_0^1 - u_0^0) > \sqrt{(hf_1^1)^2}, \\ \left(\dfrac{hf_1^1}{2(u_1^0 - u_0^0)}, 1\right), & u_1^0 > u_0^1, 2(u_1^0 - u_0^0) > \sqrt{(hf_1^1)^2}, \\ (1,\ 1), & \text{otherwise}. \end{cases}
\end{aligned}
\tag{U$_\infty$}
$$

Schemes based on $\boldsymbol{D(u^2)}$

For the scheme (27), it is useful to express the schemes for $v_i^j := (u_i^j)^2$. Choosing $A = A_1$, we obtain:

$$
\begin{aligned}
2h\sqrt{v_1^1}S_h &= v_1^1 - \frac{1}{2}\left(v_0^1 + v_1^0 + \sqrt{(v_0^1 - v_1^0)^2 + (hf_1^1)^2}\right), \\
v_1^1 &= \frac{1}{2}\left(v_0^1 + v_1^0 + h^2f^2 + \sqrt{(v_0^1 - v_1^0)^2 + 2(v_0^1 - v_1^0)(hf_1^1)^2 + (hf_1^1)^4}\right), \\
\boldsymbol{\alpha}^* &= \left(\frac{1}{2}\Big(1 + \frac{v_0^1 - u_1^0}{(v_0^1 - v_1^0)^2 + 4v_1^1(hf_1^1)^2}\Big), \frac{1}{2}\Big(1 - \frac{v_0^1 - u_1^0}{(v_0^1 - v_1^0)^2 + 4v_1^1(hf_1^1)^2}\Big)\right).
\end{aligned}
\tag{V$_1$}
$$

Again, we have that the scheme using A_1 is identical to that of [9]. For $A = A_\infty$, we have with

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

that

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

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

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

Higher Order Filtered Schemes

It is known that one typically 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

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

This can be implemented by first solving $S_h^a = 0$. If this solutions satisfies $|S_h^m| \leq k\sqrt{h}$, 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:

$$S_h^a = \max_{\alpha \in A} \frac{-\max\{g_{\boldsymbol{x},\frac{h}{2}}[u](\boldsymbol{x} + h\boldsymbol{\alpha}) - g_{\boldsymbol{x},\frac{h}{2}}[u](\boldsymbol{x} - h\boldsymbol{\alpha}), 0\}}{h} + f(x)\sqrt{\alpha_1\alpha_2} = 0 \tag{33}$$

$$S_h^a = \max_{\alpha \in A} \frac{-\max\{g_{\boldsymbol{x},\frac{h}{2}}[u^2](\boldsymbol{x} + h\boldsymbol{\alpha}) - g_{\boldsymbol{x},\frac{h}{2}}[u^2](\boldsymbol{x} - h\boldsymbol{\alpha}), 0\}}{2u(\boldsymbol{x})h} + f(x)\sqrt{\alpha_1\alpha_2} = 0. \tag{34}$$

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

We approximate $g_{\boldsymbol{x},\frac{h}{2}}[u](\boldsymbol{x}+h\boldsymbol{\alpha})$ 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

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

With similar linear approximations in (34), we need an approximation of $u(\boldsymbol{x})$, present in the denominator, since $\boldsymbol{x}$ does not coincide with a grid cell. Approximating $2u(\boldsymbol{x}) = u_0^1 + u_1^0$, we obtain

$$\begin{aligned} v_1^1 &= v_0^0 + \frac{1}{2}h^2 f^2 + \sqrt{(v_0^1 - v_1^0)^2 + (2v_0^0 + v_0^1 + v_1^0)(hf_1^1)^2 + \frac{1}{4}(hf_1^1)^4}, \\ \boldsymbol{\alpha}^* &= \left(\frac{1}{2}\Big(1 + \frac{v_0^1 - u_1^0}{(v_0^1 - v_1^0)^2 + 4v_1^1(hf_1^1)^2}\Big), \frac{1}{2}\Big(1 - \frac{v_0^1 - u_1^0}{(v_0^1 - v_1^0)^2 + 4v_1^1(hf_1^1)^2}\Big)\right). \end{aligned}$$

Fully Discretised Schemes

The scheme (25) can in fact be used to formulate fully discretised schemes with some modification. We start by replacing A with a variable admissible space $A = A_h \subset \mathbb{N}_0^2 \backslash \mathbf{0}$, 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

$$u(\boldsymbol{x}) = \max_{\boldsymbol{\alpha}\in A_h} u(\boldsymbol{x} - h\boldsymbol{\alpha}) + hf(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} \tag{35}$$

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

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

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 7 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 a 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.

Numerical Computation of Geodesics

The numerical solution of the value function gives an estimate to $u(\mathbf{1})$, which in turn can be used to approximate the geodesic distance through $\operatorname{dist}^{\mathcal{S}}([c_1], [c_2]) = \arccos(u(\mathbf{1}))$. Additionally, through a backtracking method, we can use the value function to obtain an estimate of the solution $\boldsymbol{\varphi}$ of the variational problem (17). This can then be used to estimate the shape space geodesic between c_1 and c_2.

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 $\mathbb{R}^2$.

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.

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

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

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

Numerical Experiments

In Figure 3, some examples of pairs of curves and their SRVT geodesics are illustrated. The geodesics have been computed using the (V_∞) scheme. The first three pairs of curves are used as numerical experiments, labeled A, B, C. Test problems (A) and (C) uses C^1 composite Bézier curves while test problem (B) uses C^2 composite Bézier curves. For each of the test problems, the schemes were run 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).

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 [34] that filtered schemes can give an improvement for simple problems. However, for our experiments, we found that there

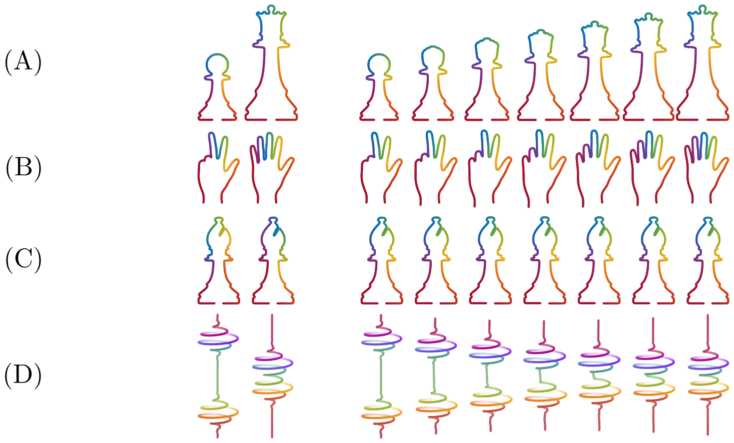

Figure 3: Left: pairs of curves coloured by initial parametrisation. Right: shape space geodesics computed using the (V_∞) scheme.

was no significant improvement of the filtered schemes compared to the best performing of the monotone schemes.[2] It seems that filtered schemes require higher-order schemes which are stable, or close to stable, on their own. While some second order schemes have been found to be stable without filtering (see [34]), this is not the case for the schemes we have considered. Therefore, we will only compare the four monotone schemes presented together with the fully discretised dynamic programming.

To implement the schemes, we used Python using NumPy with vectorised updates. The fully discretised schemes were tested for many different choices of A_h. The choice which on average gave the best performance of precision vs computation time across different test problems (including test problems in addition to the three presented here), was $A_h = \{\boldsymbol{\alpha} \in \mathbb{N}_0^2 \mid |\boldsymbol{\alpha}| \leq kh^{-r}\}$ for $k = \frac{3}{4}$ and $r = \frac{1}{2}$. For all schemes, we found that the approximation of f as described in appendix A gave better results compared to using an exact evaluation of f. These approximations were therefore used in the following experiments.

Presence of Local Solutions

For fixed computational resources available, gradient based methods have the potential to be more precise than dynamic programming based methods, given a good balance between number of discretisation points and number of iterations. However, gradient based methods will not necessarily find global solutions, if local solutions are present. Therefore, it is important to assess whether this is the case for our variational problem. In order to do this, we consider the total value function

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

[2] There is in some cases a small improvement, but this is outweighed by the added computational time.

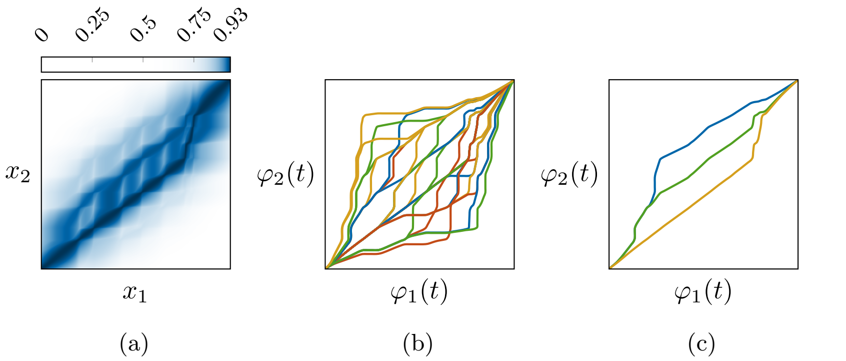

Figure 4: The total value function (a), some local maxima of (17) (b), and three local maxima near the diagonal (c).

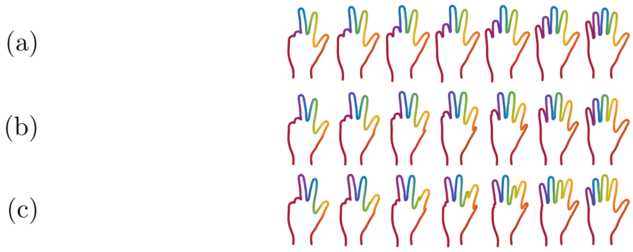

Figure 5: Pre-shape geodesics with parametrisation corresponding to the three local maxima visualised in Figure 4c.

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 (17) 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 (17). Note that all maxima of u_{tot} are inherently flat, meaning that the maxima actually come as paths of maxima.

The total value function is easy to compute. The "standard" value function (18) was defined by maximising over all paths from $\mathbf{0}$ to $\boldsymbol{x}$. Alternatively, we can define a reversed value function where we optimise over all paths from $\boldsymbol{x}$ to $\mathbf{1}$. 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 (17).

For test problem (B), the total value function was estimated using $h = 5 \cdot 10^{-4}$. The estimate is visualised in Figure 4a and the local maxima of (17) are visualised in Figure 4a. Note that a highly nonlinear colormap has been used in Figure 4a 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, and computed the resulting curve space geodesics. The result is visualised in Figure 5. 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.

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 6. 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 [9], 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 [9], 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.

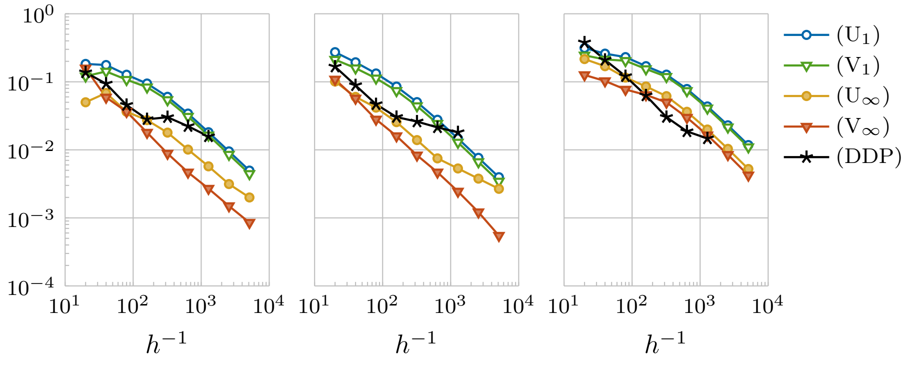

Figure 6: Convergence of u_h for test problems (A) to (C).

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:

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

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 7 and 8, respectively.

We seem to have numerical convergence for all methods considered. The error does not decrease monotonically in all cases (e.g. schemes U_∞ and DDP for test problem (A) and V_∞ for test problem (B)), but we believe this to be due to cancellation effects. This can occur when two sources of numerical errors, one contributing positively, the other negatively, cancel each other out, leading to a superficially low apparent error.

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.[3]

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.

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

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

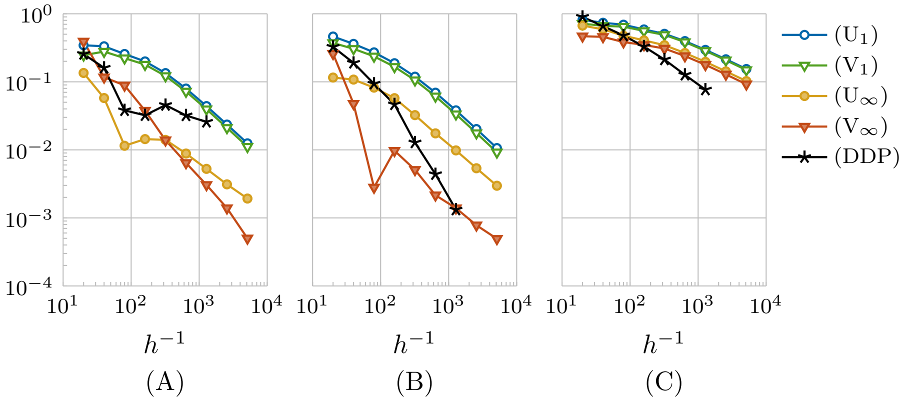

Figure 7: Convergence of $\arccos u_h(\mathbf{1})$ for test problems (A) to (C).

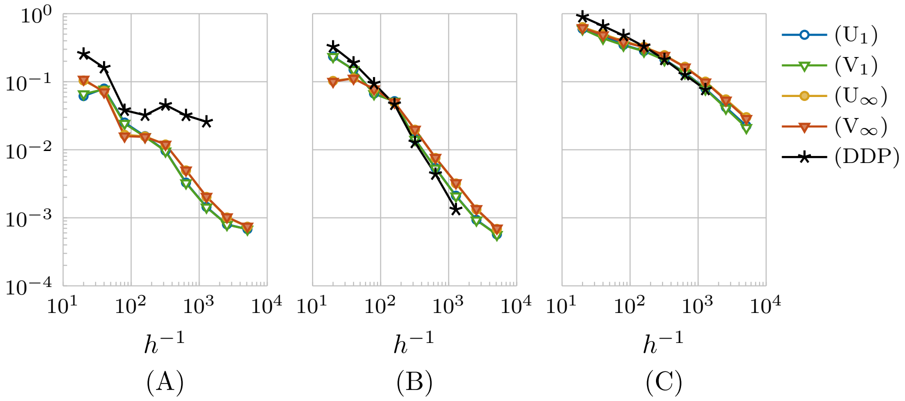

Figure 8: Convergence of $\arccos J_h(\varphi_h)$ for test problems (A) to (C).

geodesics

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

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

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

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 9, 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.

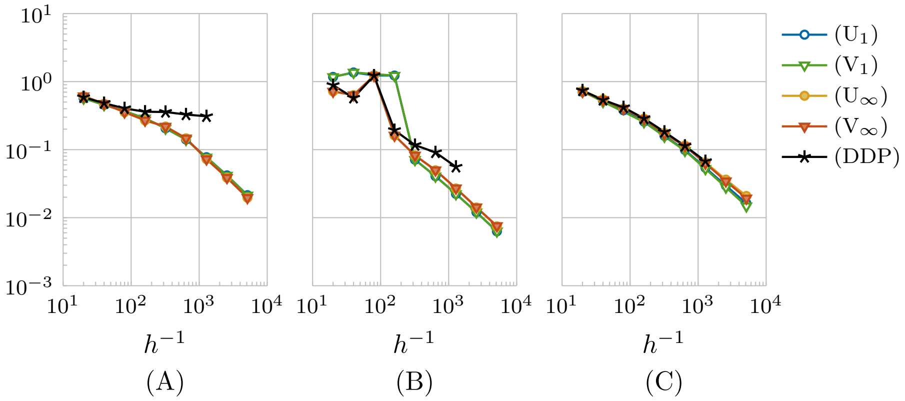

Figure 9: Convergence of γ_h for test problems (A) to (C).

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 asymptotically $\mathcal{O}(|A_h|N^2)$, while for the semi-discretised schemes, the computational complexity is $\mathcal{O}(N^2)$. If A_h is fixed in size, the two methods are asymptotically equal in complexity. However, in this case, the fully discretised scheme will converge to solutions of a different HJB equation, namely

$$\max_{\alpha \in A_h} -\max\{Du(\boldsymbol{x})\boldsymbol{\alpha}, 0\} + f(\boldsymbol{x})\sqrt{\alpha_1\alpha_2} = 0.$$

In order for the fully discretised scheme to converge to the solution of the correct HJB equation, A_h must extend in size as $N \to \infty$. A possible set of conditions for this can be found in appendix B. Regardless of the choice of A_h, this will lead to a method with a computational complexity strictly worse than $\mathcal{O}(N^2)$.

We note that dynamic programming methods with linear computational complexity have been developed [16, 5]. However, these methods are not global solvers, meaning that they are more suitable to be compared to other non-global solvers. If the semi-discretised schemes discussed in this article are to be compared with fully discretised schemes, they should be compared to the fully discretised schemes with global convergence.

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 10. As one can see, the semi-discretised scheme (V$_\infty$) performs significantly better than the fully discretised method. We recognise that this does not prove that our method is

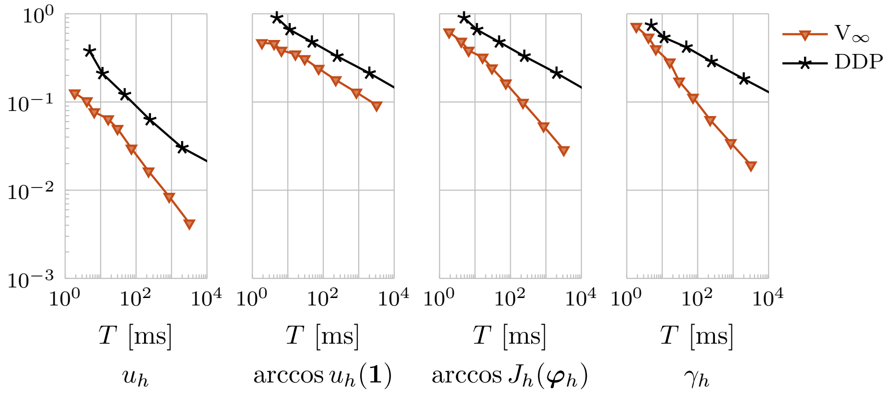

Figure 10: Work-precision diagrams for test problem (C). T denotes the computation time.

always more efficient, and there might be choices of A_h which performs better than the choice used in these experiments. However, the choice of A_h is a hyper-parameter which must be chosen selectively for the given problem in order to have optimal performance. Our schemes have no hyper-parameters and work well "out of the box".

Conclusion

In this article, we have shown how a 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 [10, 9]. These are based on the Hamilton-Jacobi-Bellman equation for the value function of the problem. However, whereas the schemes of [10, 9] 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 [9, schemes S2, S3].
- Assessing theoretical convergence of the backtracking method similar to the work done in [8].

- 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 [16, 5, 6].

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, Martins Bruveris, Stephen Marsland, and Peter W. Michor. Constructing reparameterization invariant metrics on spaces of plane curves. *Differential Geom. Appl.*, 34:139–165, 2014.

[3] Martin Bauer, Nicolas Charon, Eric Klassen, and Alice Le Brigant. Intrinsic Riemannian metrics on spaces of curves: theory and computation, 2020. arXiv:2003.05590.

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

[5] 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.

[6] Javier Bernal, James Lawrence, Gunay Dogan, and Robert Hagwood. On computing elastic shape distances between curves in d-dimensional space, 2021. NIST Technical Note 2164.

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

[8] Jeff Calder. Directed last passage percolation with discontinuous weights. *J. Stat. Phys.*, 158(4):903–949, 2015.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Brendan Cook and Jeff Calder. Rates of convergence for the continuum limit of nondominated sorting, 2020. arXiv:2006.05953.

[14] Kristian Debrabant and Espen R. Jakobsen. Semi-lagrangian schemes for linear and fully non-linear hamilton-jacobi-bellman equations, 2014.

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

[16] 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.

[17] 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.

[18] Maurizio Falcone. A numerical approach to the infinite horizon problem of deterministic control theory. *Applied Mathematics and Optimization*, 15(1):1–13, Jan 1987.

[19] Maurizio Falcone and Roberto Ferretti. Semi-lagrangian schemes for hamiltonjacobi equations, discrete representation formulae and godunov methods. *Journal of Computational Physics*, 175:559–575, 01 2002.

[20] Emmanuel Hartman, Yashil Sukurdeep, Nicolas Charon, Eric Klassen, and Martin Bauer. Supervised deep learning of elastic srv distances on the shape space of curves, 2021. arXiv:2101.04929.

[21] 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.

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

[23] S. Kurtek, E. Klassen, Z. Ding, S.W. Jacobson, J.L. Jacobson, M.J. Avison, and A. Srivastava. Parameterization-invariant shape comparisons of anatomical surfaces. *IEEE Transactions on Medical Imaging*, 30(3):849–858, 2011. cited By 52.

[24] Sebastian Kurtek, Eric Klassen, John C. Gore, Zhaohua Ding, and Anuj Srivastava. Elastic geodesic paths in shape space of parameterized surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1717–1730, 2012.

[25] Sayani Lahiri, Daniel Robinson, and Eric Klassen. Precise matching of PL curves in $\mathbb{R}^N$ in the square root velocity framework. *Geom. Imaging Comput.*, 2(3):133–186, 2015.

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

[27] Meena Mani, Sebastian Kurtek, Christian Barillot, and Anuj Srivastava. A comprehensive Riemannian framework for the analysis of white matter fiber tracts. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1101–1104, 2010.

[28] Washington Mio, Anuj Srivastava, and Shantanu Joshi. On Shape of Plane Elastic Curves. *International Journal of Computer Vision*, 73(3):307324, 2007.

[29] Tom Needham and Sebastian Kurtek. Simplifying transforms for general elastic metrics on the space of plane curves. *SIAM J. Imaging Sci.*, 13(1):445–473, 2020.

[30] Elvis Nunez and Shantanu H. Joshi. Deep learning of warping functions for shape analysis. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3782–3790, 2020.

[31] 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.

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

[33] 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.

[34] 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.

[35] Alain Trouvé and Laurent Younes. On a class of diffeomorphic matching problems in one dimension. *SIAM J. Control Optim.*, 39(4):1112–1135, 2000.

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

Approximating the SRVTs

The schemes presented in this article are based on exact computation of the forcing term $f(\boldsymbol{x})$, which in turn 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 as compared to using exact evaluations of f.

Convergence for Fully Discretised Schemes

We express the scheme (35) as the largest solution of $S_h = 0$ with

$$S_h = \max_{\boldsymbol{\alpha}\in A_h} \frac{\max\{u(\boldsymbol{x} - h\boldsymbol{\alpha}) - u(\boldsymbol{x}), 0\}}{h|\boldsymbol{\alpha}|} + f(\boldsymbol{x})\frac{\sqrt{\alpha_1\alpha_2}}{|\boldsymbol{\alpha}|}. \tag{36}$$

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 $\boldsymbol{\beta} \in \mathbb{R}^2_{\geq 0}$ with $|\boldsymbol{\beta}| = 1$ and every $\epsilon > 0$, there exists $h_0 > 0$ such that for every $0 < h \leq h_0$, there is $\boldsymbol{\alpha} \in A_h$ with $|\boldsymbol{\alpha}/|\boldsymbol{\alpha}| - \boldsymbol{\beta}| < \epsilon$.*

Theorem 7. *Under assumption 3, the scheme* (36) *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

$$\begin{aligned} u_h(\boldsymbol{y}) \leq u_h(\boldsymbol{x}) &= u_h(\boldsymbol{y}) + f(\boldsymbol{x})\sqrt{(x_1 - y_1)(x_2 - y_2)} \\ &\leq u_h(\boldsymbol{y}) + \|f\|_\infty \sqrt{(x_1 - y_1)(x_2 - y_2)}. \end{aligned}$$

  Inductively, this gives that $0 \leq u(\boldsymbol{x}) \leq \|f\|_\infty \sqrt{x_1 x_2}$.
- *Consistency*: We have that

$$\begin{aligned} &S_h\big(\boldsymbol{y}, \psi(\boldsymbol{y}) + \xi, \psi + \xi\big) \\ &\qquad = \max_{\boldsymbol{\alpha}\in A_h} \frac{\max\{\psi(\boldsymbol{y} - h\boldsymbol{\alpha}) - \psi(\boldsymbol{y}), 0\}}{h|\boldsymbol{\alpha}|} + f(\boldsymbol{y})\frac{\sqrt{\alpha_1\alpha_2}}{|\boldsymbol{\alpha}|} \\ &\qquad = \max_{\boldsymbol{\alpha}\in A_h} -\max\Big\{D\psi(\boldsymbol{y})\frac{\boldsymbol{\alpha}}{|\boldsymbol{\alpha}|} + O(h|\boldsymbol{\alpha}|), 0\Big\} + f(\boldsymbol{y})\frac{\sqrt{\alpha_1\alpha_2}}{|\boldsymbol{\alpha}|}. \end{aligned}$$

  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 are uniformly continuous in y. This, combined with assumption 3d, gives that

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

with $r = 2$. Due to [1, Theorem 2.1], this proves convergence. □