



Equivalent Extensions of Hamilton–Jacobi–Bellman Equations on Hypersurfaces

Lindsay Martin¹ · Yen-Hsi Richard Tsai²

Received: 13 October 2019 / Revised: 8 June 2020 / Accepted: 4 August 2020 / Published online: 20 August 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

We present a new formulation for the computation of solutions of a class of Hamilton Jacobi Bellman (HJB) equations on closed smooth surfaces of co-dimension one. For the class of equations considered in this paper, the viscosity solution of the HJB equation is equivalent to the value function of a corresponding optimal control problem. In this work, we extend the optimal control problem given on the surface to an equivalent one defined in a sufficiently thin narrow band of the co-dimensional one surface. The extension is done appropriately so that the corresponding HJB equation, in the narrow band, has a unique viscosity solution which is identical to the constant normal extension of the value function of the original optimal control problem. With this framework, one can easily use existing (high order) numerical methods developed on Cartesian grids to solve HJB equations on surfaces, with a computational cost that scales with the dimension of the surfaces. This framework also provides a systematic way for solving HJB equations on the unstructured point clouds that are sampled from the surface.

Keywords Hamilton–Jacobi–Bellman equations · Eikonal equations · Partial differential equations on surfaces · Optimal control

Mathematics Subject Classification Primary 70H20 · 49J20 · 35R01 · 58E10 · 65N06

1 Introduction

Hamilton–Jacobi–Bellman equations have many applications in optimal control, seismology, geometrical optics, etc. From the solutions of HJB equations on surfaces, the corresponding characteristics curves can be extracted and used in many applications. Some examples include

✉ Lindsay Martin
lmartin@math.utexas.edu

Yen-Hsi Richard Tsai
ytsai@math.utexas.edu

¹ Department of Mathematics, The University of Texas at Austin, Austin, TX 78712, USA

² Department of Mathematics and Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA

mesh generation [15], path planning [19,37], and brain mapping [4,21]. The equations are fully nonlinear and classical solutions typically do not exist. The unique viscosity solution [10] is often sought after. Sophisticated algorithms have been developed for computing the viscosity solution of HJB equations defined in Euclidean space.

Let $\Omega \subset \mathbb{R}^n$ where $n = 2$ or 3 be a bounded and connected open set with smooth closed boundary $\Gamma = \partial\Omega$. Our goal is to compute solutions of the following HJB equation defined on smooth surfaces, with given Dirichlet boundary conditions:

$$\min_{\mathbf{a} \in A_{\mathbf{x}}} \left\{ r(\mathbf{x}, \mathbf{a}) + \nabla_{\Gamma} u(\mathbf{x}) \cdot f(\mathbf{x}, \mathbf{a}) \right\} = 0, \quad \mathbf{x} \in \Gamma \setminus \mathcal{T} \quad (1)$$

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \mathcal{T}. \quad (2)$$

The set $A_{\mathbf{x}} := S^{n-1} \cap T_{\mathbf{x}}\Gamma$ is a compact set where S^{n-1} is the $(n-1)$ -dimensional unit sphere in $\mathbb{R}^n$ and $T_{\mathbf{x}}\Gamma$ is the tangent space of Γ at the point $\mathbf{x} \in \Gamma$. We define $A := S^{n-1} \cap TM(\Gamma)$ to be the unit tangent bundle of Γ where $TM(\Gamma)$ is the tangent bundle of Γ . Here $r, f : \Gamma \times A \rightarrow \mathbb{R}$, $\mathcal{T}$ is a closed subset of Γ , and $\nabla_{\Gamma} u$ is the surface gradient on Γ [11].

For $\epsilon > 0$, define the narrow band of Γ by

$$T_{\epsilon} := \{\mathbf{z} \in \mathbb{R}^n : \min_{\mathbf{x} \in \Gamma} \|\mathbf{x} - \mathbf{z}\| < \epsilon\}.$$

In this paper, we shall derive a Hamiltonian, $\bar{H}$, and extensions $\bar{\mathcal{T}}$ and $\bar{g}$ of $\mathcal{T}$ and g , respectively, such that the viscosity solution to

$$\bar{H}(\mathbf{z}, \nabla v(\mathbf{z})) = 0, \quad \mathbf{z} \in T_{\epsilon} \setminus \bar{\mathcal{T}} \quad (3)$$

$$v(\mathbf{z}) = \bar{g}(\mathbf{z}), \quad \mathbf{z} \in \bar{\mathcal{T}} \quad (4)$$

is the constant normal extension of the solution to (1)–(2), for any positive and sufficiently small ϵ .

Our contribution includes a theory for how optional control problems on surfaces can be extended into “equivalent” ones defined in a narrow band of the surface. Depending on whether the problem is isotropic or anisotropic, we must take careful consideration of extending the control space. We show for the anisotropic or most general case, the same control space used in the surface problem must be used in defining the extended problem. However, in the isotropic case the extension is equivalent even when we appropriately extend the control space. After defining the extension of optimal control problems, we then extend the corresponding HJB equation defined on the narrow band of the surface. The main advantage of this approach is that to compute solutions of HJB equations on surfaces, we are able to use Cartesian grids and existing methods (including high order methods) which solve HJB equations in Euclidean space. This allows us to avoid unnecessary patching and triangulation for solving the surface HJB equation. We show that in fact the narrow band can be very thin, i.e., its thickness is of order h , where h is the grid size.

Before proceeding, we will first define the closest point mapping that is used in the extensions. Define the closest point mapping, $P_{\Gamma} : T_{\epsilon} \rightarrow \Gamma$, by

$$P_{\Gamma}(\mathbf{z}) := \arg \min_{\mathbf{y} \in \Gamma} \|\mathbf{z} - \mathbf{y}\|.$$

Here, ϵ must be chosen so that the closest point mapping is unique. Then we define the signed distance function to Γ , $d_{\Gamma} : T_{\epsilon} \rightarrow \mathbb{R}$, by

$$d_{\Gamma}(\mathbf{z}) = \begin{cases} \|\mathbf{z} - P_{\Gamma}(\mathbf{z})\| & \text{if } \mathbf{z} \in \bar{\Omega}^c. \\ -\|\mathbf{z} - P_{\Gamma}(\mathbf{z})\| & \text{if } \mathbf{z} \in \Omega. \end{cases}$$

For $|\eta| < \epsilon$, we will define the parallel surface, Γ_η , by

$$\Gamma_\eta := \{\mathbf{z} \in T_\epsilon \mid d_\Gamma(\mathbf{z}) = \eta\}.$$

Closest point mappings are easily derived in the context of level set methods [30], and there are a variety of fast and high order methods available to compute them from the distance function to Γ [8,33,36,37,39], i.e.,

$$P_\Gamma(\mathbf{z}) = \mathbf{z} - d_\Gamma(\mathbf{z})\nabla d_\Gamma(\mathbf{z}).$$

Finally, we define the constant normal extension of function on Γ , $u : \Gamma \rightarrow \mathbb{R}$, to be $\bar{u} : T_\epsilon \rightarrow \mathbb{R}$ given by

$$\bar{u}(\mathbf{z}) := u(P_\Gamma(\mathbf{z})).$$

Again, our goal is to define an extended HJB equation on T_ϵ so that the solution is the constant normal extension to the solution of the HJB equation on the surface.

The rest of the paper is organized as follows: In Sect. 2, we present the optimal control problem which models controlled motion on a surface and define the related class of HJB equations. We then extend the control problem and HJB equations on the narrow band, T_ϵ . In Sect. 3, we present a special case of the problem when the HJB equation reduces to the Eikonal equation and the distance function on the surface is desired. Finally, we give some numerical simulations in Sect. 4.

1.1 Previous Work: Solving HJB Equations in Euclidean Space

There is extensive work on developing numerical methods to compute solutions to (1) where the domain is a bounded open set in $\mathbb{R}^n$. In the view of a optimal control problem, one can obtain a semi-Lagrangian discretization for HJB equations which gives a large system of coupled nonlinear equations. Extensive studies of semi-Lagrangian techniques are found in [3,12–14]. Solving the discretized system using fixed-point iterations is expensive, thus fast marching methods (FMMs) and fast sweeping methods (FSMs) were developed.

FMMs are a variant of Dijkstra's method and use a heapsort algorithm to determine the order in which the grid nodes are updated. The algorithm was first developed on Cartesian grids [33,37] to solve Eikonal equations which are a special case of (1) when the functions r and f are isotropic, i.e., $r(\mathbf{x}, \mathbf{a}) = r(\mathbf{x})$ and $f(\mathbf{x}, \mathbf{a}) = f(\mathbf{x})$. The solution to the Eikonal equation is well known to be the distance function to the set $\mathcal{T}$. FMMs have complexity $\mathcal{O}(N \log N)$, where N is the total number of unknowns [31,37]. Based on the Godunov upwind numerical scheme developed to solve Eikonal equations on triangulated domains in [5], FMMs were extended to acute triangulated meshes in [17]. However, FMMs do not directly apply to the anisotropic case. FMMs were applied to compute solutions of a class of "axis-aligned" anisotropic HJ equations in [1] on orthogonal grids, and ordered upwind methods (OUMs) were developed to solve a class of general HJB equations on any triangulated surface [34]. In [6,7], related PDEs are solved using similar fast methods.

FSMs were developed in [36,40] to compute solutions of a class of strictly convex HJ equations, including Eikonal equation, on Cartesian grids. The FSM is an iterative method that relies on an upwind discretization of the PDE and Gauss-Seidel style updates with different orderings of grid nodes. The idea is to avoid using heapsort and, instead, update the grid nodes in different orderings (sweeps). In each sweep, characteristics that go in similar directions are propagated automatically. In [22], a FSM was developed for Eikonal equations using a discontinuous Galerkin (DG) local solver for computing the distance function.

For more general Hamilton–Jacobi equations, one may use a Lax–Friedrichs type numerical Hamiltonian [16] on Cartesian grids. The resulting Lax–Friedrichs sweeping scheme is typically more diffusive and requires more iterations. FSMs have complexity $\mathcal{O}(N)$ with the caveat that the constant hidden in the notation may be large depending on the characteristics of the PDE.

1.2 Previous Work: Solving HJB Equations on Surfaces

Several versions of FMMs and FSMs can be applied to computing the distance function on surfaces. As already mentioned, the FMM in [17] only works for solving Eikonal equations on acute triangulated surfaces. In [35], FMMs equation were extended to solve Eikonal equations on parametric surfaces where the discretization is done on a Cartesian grid in the parametric plane. The FSM in [36] and the FMM in [29] can compute the distance function on surfaces that are defined as the graph of a smooth function. Another approach to computing the distance function on surfaces is to solve a corresponding evolutionary HJ equation where the distance function is the steady state solution [8]. However, with the addition of the time variable this method is not computationally optimal.

OUMs are applicable to more general anisotropic HJB equations on triangulated surfaces. However, if the surface is given in implicit form, triangulation may be unnecessary. We want to avoid triangulation and have the ability to handle general surface representations. Our motivation is to derive equivalent extended HJB equations on a narrow band of the surface such that a variety of meshes and methods can be used to solve the equivalent equations in the narrow band. In particular, the methods developed for Cartesian grids (surveyed in Sect. 1.1) can then be used.

In [27], the idea of extending the surface by a small offset, ϵ , to a narrow band is used to compute distance functions on surface. On the narrow band, an extended distance function is computed, but it is not the constant normal extension of the distance function on the surface. Therefore, the formulation introduces an analytical error. It is proven that the error between the two distance functions is controlled by ϵ , the width of the narrow band. In order for the method to converge as the grid size goes to zero, it is required that $\epsilon = Ch^\gamma$ where h is the grid size and $\gamma \in (0, 1)$. The advantage of our framework is that there no analytical error introduced. Our formulation is convergent with respect to the grid size, h , and for narrow bands of order h . We are also able to perform high order computations for the distance function on surfaces.

The idea of extending functions or differential operators defined on surfaces, via closest point mapping, to the embedding Euclidean space, is used in [9,24,32] to compute solutions to parabolic and elliptic PDEs. Our approach is inspired by the work in [9] where variational integrals defined originally on surfaces are extended to ones defined in the narrow band, and the Euler–Lagrange equations of the extended problem are solved by standard numerical methods. Due to the way the extensions are defined, the resulting solution to each equation is automatically the constant normal extension of the solution to the variational problem on the surface.

1.3 Computing PDEs “on point clouds”

Point clouds arise in many applications including facial recognition, manufacturing, medicine, and geosciences and can be acquired easily through modern sensing devices such as laser scanners and cell phones. A typical approach, particularly predominant in the com-

puter graphics community, is to create triangular meshes from the data points and seek to define partial derivatives in certain suitable ways. Among myriad of papers using triangular surface meshes, we mention [5], an early paper which is the most related to the topic of this paper. Another school of approaches aim at solving differential equations on surfaces, using only a finite set of points sampled from the surfaces, without globally reconstructing the surface.

In [28], the algorithm developed in [27] is extended to compute distance functions on surfaces represented as point clouds. In [23], a framework for computing solutions to PDEs on point clouds based on a local approximation of the manifold was presented. A local mesh algorithm was developed in [20] that solves PDEs on point clouds where any of the existing methods valid on triangular meshes discussed in Sect. 1.2 can be used to compute the solutions to HJB equations once the local mesh is determined.

Our new proposed non-parametric formulation also provides a convenient way to solve equations “on point clouds.” That is we can solve HJB equations which are defined on a smooth closed surface, but the only available information about the surface is a finite set of points that are reasonably distributed over the surface. The discussion of the implementation of our method applied to point clouds is in Sect. 4.1.4. There, it is assumed that there is no noise perturbing the point cloud in the surface normal directions, and that the point clouds are evenly distributed over the surfaces. The generalization of the proposed algorithm to more general point clouds is the subject of another paper.

2 Static Hamilton–Jacobi–Bellman Equations on Surfaces

2.1 Modeling Controlled Motion on a Surface

First, we define the optimal control problem from which the HJB equations can be derived. For each $\mathbf{x} \in \Gamma$, let $A_{\mathbf{x}}$ be set of all unit tangent vectors of Γ at $\mathbf{x}$, i.e.,

$$A_{\mathbf{x}} = S^{n-1} \cap T_{\mathbf{x}}\Gamma.$$

Note each $A_{\mathbf{x}}$ is a compact set in $\mathbb{R}^n$ for each $\mathbf{x} \in \Gamma$. Define $A := \bigcup_{\mathbf{x} \in \Gamma} A_{\mathbf{x}}$, then $A = S^{n-1} \cap TM(\Gamma)$ where $TM(\Gamma)$ is the tangent bundle of Γ . The set of admissible controls is given by:

$$\mathcal{A} = \{\text{measurable functions } \mathbf{a} : [0, \infty) \rightarrow A\}.$$

We are interested in the trajectories governed by the dynamical system:

$$\frac{d\mathbf{y}}{dt}(t) = f(\mathbf{y}(t), \mathbf{a}(t))\mathbf{a}(t), \quad t > 0 \quad (5)$$

$$\mathbf{y}(0) = \mathbf{x}, \quad (6)$$

where $f : \Gamma \times A \rightarrow \mathbb{R}$ represents the velocity and $\mathbf{a}(t) \in A_{\mathbf{y}(t)}$, which ensures that $\mathbf{y}(t) \in \Gamma$ for all $t \geq 0$. Denote the solution to (5)–(6) (which exists under the assumption (A1) below) by $\mathbf{y}_{\mathbf{x}}(t, \mathbf{a}(t))$. For each $\mathbf{x} \in \Gamma$ define the following subset of the admissible controls:

$$\mathcal{A}_{\mathbf{x}} := \{\mathbf{a}(\cdot) \in \mathcal{A} \mid \mathbf{y}_{\mathbf{x}}(t, \mathbf{a}(t)) \in \Gamma \text{ for all } t \geq 0\}.$$

Then the requirement that $\mathbf{a}(t) \in A_{\mathbf{y}(t)}$ for all $t \geq 0$ in the dynamical system (5)–(6) is equivalent to $\mathbf{a}(\cdot) \in \mathcal{A}_{\mathbf{x}}$. Refer to $\mathbf{y}(\cdot)$ that satisfy (5)–(6) where $\mathbf{a}(\cdot) \in \mathcal{A}_{\mathbf{x}}$ as an admissible

path on Γ . For the rest of the paper, we will suppress writing the dependence of $\mathbf{y}_{\mathbf{x}}(t, \mathbf{a}(t))$ on $\mathbf{a}(\cdot)$ when there is no confusion in the context.

Let $r : \Gamma \times A \rightarrow \mathbb{R}$ denote a running cost per unit time and $g : \mathcal{T} \rightarrow \mathbb{R}$ be an exit time penalty when the state reaches a closed target set $\mathcal{T} \subset \Gamma$. The total cost associated with initial state $\mathbf{x}$ and control $\mathbf{a}(\cdot) \in \mathcal{A}_{\mathbf{x}}$ to reach $\mathcal{T}$ is given by

$$C(\mathbf{x}, \mathbf{a}(\cdot)) = \int_0^T r(\mathbf{y}_{\mathbf{x}}(t), \mathbf{a}(t)) dt + g(\mathbf{y}_{\mathbf{x}}(T)), \quad (7)$$

where $T = \min\{t \mid \mathbf{y}_{\mathbf{x}}(t) \in \mathcal{T}\}$.

We have the following assumptions on f , r , and g :

$$\left\{ \begin{array}{l} f \text{ and } r \text{ are Lipschitz continuous in their first arguments.} \\ g \text{ is lower semicontinuous and } \min_{\mathcal{T}} g < \infty. \\ \text{There exists constants } R_1, R_2, F_1, \text{ and } F_2 \text{ such that:} \\ 0 < R_1 \leq r(\mathbf{x}, \mathbf{a}) \leq R_2, \text{ for all } \mathbf{x} \in \Gamma \text{ and } \mathbf{a} \in A, \\ 0 < F_1 \leq f(\mathbf{x}, \mathbf{a}) \leq F_2, \text{ for all } \mathbf{x} \in \Gamma \text{ and } \mathbf{a} \in A. \end{array} \right. \quad (\text{A1})$$

The value function $u : \Gamma \rightarrow \mathbb{R}$ is defined to be the minimal total cost to $\mathcal{T}$ starting at $\mathbf{x}$ and is given by

$$u(\mathbf{x}) = \inf_{\mathbf{a}(\cdot) \in \mathcal{A}_{\mathbf{x}}} C(\mathbf{x}, \mathbf{a}(\cdot)). \quad (8)$$

Under the assumption (A1), an optimal control does not necessarily exist. If for the given f and r , we have that the set $V(\mathbf{x}) = \{f(\mathbf{x}, \mathbf{a})\mathbf{a}/r(\mathbf{x}, \mathbf{a}) \mid \mathbf{a} \in A_{\mathbf{x}}\}$ is strictly convex for each $\mathbf{x} \in \Gamma$, then an optimal control is guaranteed [2]. This property is trivially satisfied in the case of isotropic f and r . Regardless of whether an optimal control exists, we can still derive the corresponding HJB equation.

The dynamic programming principle [2] for the value function states that for sufficiently small $\tau > 0$,

$$u(\mathbf{x}) = \inf_{\mathbf{a}(\cdot) \in \mathcal{A}_{\mathbf{x}}} \left\{ \int_0^\tau r(\mathbf{y}_{\mathbf{x}}(t), \mathbf{a}(t)) dt + u(\mathbf{y}_{\mathbf{x}}(\tau)) \right\}. \quad (9)$$

The corresponding Hamilton–Jacobi–Bellman equation on Γ is

$$\min_{\mathbf{a} \in A_{\mathbf{x}}} \left\{ r(\mathbf{x}, \mathbf{a}) + \nabla_{\Gamma} u(\mathbf{x}) \cdot f(\mathbf{x}, \mathbf{a}) \mathbf{a} \right\} = 0, \quad \mathbf{x} \in \Gamma \setminus \mathcal{T}. \quad (10)$$

The boundary condition is

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \mathcal{T}. \quad (11)$$

Note that in (10) we can take the minimum since $A_{\mathbf{x}}$ is compact.

In general, classical solutions of (10)–(11) do not exist, and the unique viscosity solution is sought after [10]. A detailed discussion of viscosity solutions of Hamilton–Jacobi equations on manifolds can be found in [25]. Under the assumption (A1), it is a classic result that the value function u coincides with the unique viscosity solution of (10)–(11) [2]. In the case of isotropic running cost and speed, then (10) reduces to the Eikonal equation on the surface:

$$\|\nabla_{\Gamma} u(\mathbf{x})\| = \frac{r(\mathbf{x})}{f(\mathbf{x})}. \quad (12)$$

We will expand more on the Eikonal equation in the next section.

2.2 Extension to T_ϵ

We derive a Hamiltonian, $\overline{H}$, on T_ϵ such that if $v : T_\epsilon \rightarrow \mathbb{R}$ is the unique viscosity solution of

$$\overline{H}(\mathbf{z}, \nabla v(\mathbf{z})) = 0$$

with boundary conditions defined appropriately, then

$$v(\mathbf{z}) = \overline{u}(\mathbf{z}) := u(P_\Gamma(\mathbf{z})),$$

where u is the viscosity solution to (10)–(11).

First, we extend the optimal control problem on the surface to one in the narrow band, T_ϵ . We start by defining a tensor that will be essential in formulating an equivalent optimal control problem on T_ϵ .

Definition 2.1 Let $\mathbf{t}_1(\mathbf{z})$ and $\mathbf{t}_2(\mathbf{z})$ be two orthonormal tangent vectors corresponding to the directions that yield the principle curvatures of Γ_η at a point $\mathbf{z}$. Let $\mathbf{n}(\mathbf{z})$ be the unit normal vector to the tangent plane at $\mathbf{z}$. Let $\sigma_1(\mathbf{z})$ and $\sigma_2(\mathbf{z})$ be the singular values of $P'_\Gamma(\mathbf{z})$, the Jacobian matrix of P_Γ at $\mathbf{z}$. Then for any real number μ , define the tensor B by

$$B(\mathbf{z}, \mu) = B_0(\mathbf{z}) + \mu B_1(\mathbf{z}), \quad (13)$$

where

$$\begin{aligned} B_0(\mathbf{z}) &= \sigma_1^{-1}(\mathbf{z}) \mathbf{t}_1(\mathbf{z}) \otimes \mathbf{t}_1(\mathbf{z}) + \sigma_2^{-1}(\mathbf{z}) \mathbf{t}_2(\mathbf{z}) \otimes \mathbf{t}_2(\mathbf{z}), \\ B_1(\mathbf{z}) &= \mathbf{n}(\mathbf{z}) \otimes \mathbf{n}(\mathbf{z}). \end{aligned}$$

In the above definition, B_0 corresponds to a weighted orthogonal projection onto the tangent plane of Γ_η at $\mathbf{z}$, and B_1 is the projection along the normal of Γ_η passing through $\mathbf{z}$. In [18], it is proven that

$$\sigma_i = 1 - d_\Gamma(\mathbf{z}) \kappa_i(\mathbf{z})$$

where $\kappa_1(\mathbf{z})$ and $\kappa_2(\mathbf{z})$ are the principal curvatures of the parallel surface Γ_η . Note that when $\mathbf{z} \in \Gamma$, $\sigma_1 = \sigma_2 = 1$. For the sake of notation, we will suppress the dependence of the tangent vectors, normal vectors, and singular values on the point $\mathbf{z} \in T_\epsilon$.

Assuming ϵ small enough, we have that the tangent planes coincide for $\mathbf{z} \in T_\epsilon$ and $\mathbf{x} \in \Gamma$ if $P_\Gamma(\mathbf{z}) = \mathbf{x}$. Thus, for the extended optimal control problem, the set of compact control values at each $\mathbf{z} \in T_\epsilon$ is $A_{P_\Gamma(\mathbf{z})}$. Then A and $\mathcal{A}$ are as in Sect. 2.1. Now, consider the following extended dynamical system in T_ϵ :

$$\frac{d\mathbf{y}}{dt}(t) = \overline{f}(\mathbf{y}(t), \mathbf{a}(t)) B(\mathbf{y}(t), \mu) \mathbf{a}(t), \quad t > 0 \quad (14)$$

$$\mathbf{y}(0) = \mathbf{z}, \quad \mathbf{z} \in T_\epsilon. \quad (15)$$

Here $\overline{f} : T_\epsilon \times A \rightarrow \mathbb{R}$ is given by $\overline{f}(\mathbf{z}, \mathbf{a}) = f(P_\Gamma(\mathbf{z}), \mathbf{a})$, and we have that $\mathbf{a}(t) \in A_{\mathbf{y}(t)}$ for all $t \geq 0$. This last requirement ensures that the trajectory remains in Γ_η for all $t \geq 0$ where $\eta = d_\Gamma(\mathbf{z})$. The inclusion of the tensor $B(\mathbf{y}(t), \mu)$ in the dynamical system above adjusts the velocity according to the curvature of the surface. This has the implication that equivalent paths will have the same total cost.

Denote the solution to (14)–(15) (which exists under the assumption (B1) below) by $\mathbf{y}_z(t, \mathbf{a}(t))$. For each $\mathbf{z} \in T_\epsilon$ with $d_\Gamma(\mathbf{z}) = \eta$, we have the following subset of admissible controls:

$$\mathcal{A}_z := \{\mathbf{a} \in \mathcal{A} \mid \mathbf{y}_z(t, \mathbf{a}(t)) \in \Gamma_\eta\}.$$

Then the requirement that $\mathbf{a}(t) \in A_{\mathbf{y}_z(t)}$ for all $t \geq 0$ in the dynamical system (14)–(15) is equivalent to $\mathbf{a}(\cdot) \in \mathcal{A}_z$. Since $A_x = A_{P_\Gamma(\mathbf{z})}$, we have $\mathcal{A}_z = \mathcal{A}_{P_\Gamma(\mathbf{z})}$. Refer to $\mathbf{y}(\cdot)$ that satisfy (14)–(15) where $\mathbf{a}(\cdot) \in \mathcal{A}_z$ as an admissible path on T_ϵ . Again, for the rest of the paper we will write suppress writing the dependence of $\mathbf{y}_z(t, \mathbf{a}(t))$ on $\mathbf{a}(\cdot)$. There is no confusion between the notation for admissible paths on Γ and T_ϵ . The initial point will indicate which dynamical system the path solves.

Note that if the initial point $\mathbf{z} \in \Gamma$, $\eta = 0$ and

$$B(\mathbf{y}_z(t), \mu)\mathbf{a}(t) = \mathbf{a}(t),$$

since $\mathbf{a}(t) \in T_{\mathbf{y}_z(t)}$ and $\sigma_1 = \sigma_2 = 1$. Thus, (14)–(15) reduces to the dynamical system on Γ given in (5)–(6), and an admissible path on T_ϵ with $\mathbf{a}(\cdot) \in \mathcal{A}_z$ is also an admissible path on Γ with the same $\mathbf{a}(\cdot) \in \mathcal{A}_z$.

Now, let $\bar{T} = \{\mathbf{z} \in T_\epsilon \mid P_\Gamma(\mathbf{z}) \in \mathcal{T}\}$. Let the running cost and exit time penalty on T_ϵ be given by $\bar{r} : T_\epsilon \times A \rightarrow \mathbb{R}$ where $\bar{r}(\mathbf{z}, \mathbf{a}) = r(P_\Gamma(\mathbf{z}), \mathbf{a})$ and $\bar{g} : T_\epsilon \rightarrow \mathbb{R}$ where $\bar{g}(\mathbf{z}) = g(P_\Gamma(\mathbf{z}))$, respectively. Then the total cost associated with initial state $\mathbf{z}$ and control $\mathbf{a}(\cdot) \in \mathcal{A}_z$ to reach $\bar{T}$ is given by

$$\bar{C}(\mathbf{z}, \mathbf{a}(\cdot)) := \int_0^{\bar{T}} \bar{r}(\mathbf{y}_z(t), \mathbf{a}(t))dt + \bar{g}(\mathbf{y}_z(\bar{T}))$$

where $\bar{T} = \min\{t \mid \mathbf{y}_z(t) \in \bar{T}\}$. Note that if $\mathbf{z} \in \Gamma$, then $\bar{C}(\mathbf{z}, \mathbf{a}(\cdot)) = C(\mathbf{z}, \mathbf{a}(\cdot))$. The value function $v : T_\epsilon \rightarrow \mathbb{R}$ is the minimal total cost to $\bar{T}$ starting at $\mathbf{z}$ and is given by

$$v(\mathbf{z}) = \inf_{\mathbf{a}(\cdot) \in \mathcal{A}_z} \bar{C}(\mathbf{z}, \mathbf{a}(\cdot)). \quad (16)$$

We have that the assumption (A1) implies:

$$\left\{ \begin{array}{l} \text{The mappings } (\mathbf{z}, \mathbf{a}) \mapsto \bar{r}(\mathbf{z}, \mathbf{a}) \text{ and } (\mathbf{z}, \mathbf{a}) \mapsto \bar{f}(\mathbf{z}, \mathbf{a})B(\mathbf{z}, \mu)\mathbf{a} \\ \text{are Lipschitz continuous in their first arguments.} \\ \bar{g} \text{ is lower semicontinuous and } \min_{\bar{T}} \bar{g} < \infty. \\ 0 < R_1 \leq \bar{r}(\mathbf{z}, \mathbf{a}) \leq R_2, \text{ for all } \mathbf{z} \in T_\epsilon \text{ and } \mathbf{a} \in A, \\ 0 < G_1 \leq \bar{g}(\mathbf{z}) \leq G_2, \text{ for all } \mathbf{z} \in \bar{T}, \\ \text{There exists constants } \bar{F}_1, \bar{F}_2 \text{ such that:} \\ 0 < \bar{F}_1 \leq \|\bar{f}(\mathbf{z}, \mathbf{a})B(\mathbf{z}, \mu)\mathbf{a}\| \leq \bar{F}_2, \text{ for all } \mathbf{z} \in T_\epsilon \text{ and } \mathbf{a} \in A. \end{array} \right. \quad (B1)$$

Again, while (B1) does not guarantee the existence of an optimal control, the assumption does ensure that the value function and corresponding viscosity solution of the HJB equation coincide. If we also assume the set $V(\mathbf{x})$ is strictly convex, then $\bar{V}(\mathbf{z}) = \{\bar{f}(\mathbf{z}, \mathbf{a})B(\mathbf{z}, \mu)\mathbf{a}/\bar{r}(\mathbf{z}, \mathbf{a}) \mid \mathbf{a} \in \mathcal{A}_z\}$ is strictly convex for each $\mathbf{z} \in T_\epsilon$. Thus, if an optimal control exists in the problem on Γ , it also exists for the extended problem on T_ϵ .

Before deriving the corresponding HJB equation, we prove that the value function, v , is the constant normal extension of the value function of the optimal control problem defined on Γ . The following theorem states that two admissible paths, $\mathbf{y}_{z_1}(\cdot)$ and $\mathbf{y}_{z_2}(\cdot)$, on T_ϵ with equivalent initial points in T_ϵ and equivalent controls stay equivalent for all time.

Theorem 2.1 Suppose that $\mathbf{z}_1, \mathbf{z}_2 \in T_\epsilon$ such that $P_\Gamma(\mathbf{z}_1) = P_\Gamma(\mathbf{z}_2)$. Let $\mathbf{a}_1(\cdot) \in \mathcal{A}_{\mathbf{z}_1}$ and $\mathbf{a}_2(\cdot) \in \mathcal{A}_{\mathbf{z}_2}$ such that $\mathbf{a}_1 \equiv \mathbf{a}_2$. Suppose $\mathbf{y}_{\mathbf{z}_1} : [0, \infty) \rightarrow T_\epsilon$ solves (14) with $\mathbf{y}_{\mathbf{z}_1}(0) = \mathbf{z}_1$ and $\mathbf{y}_{\mathbf{z}_2} : [0, \infty) \rightarrow T_\epsilon$ solves (14) with $\mathbf{y}_{\mathbf{z}_2}(0) = \mathbf{z}_2$. Then $P_\Gamma(\mathbf{y}_{\mathbf{z}_1}(t)) = P_\Gamma(\mathbf{y}_{\mathbf{z}_2}(t))$ for all $t \geq 0$.

Proof Let $\mathbf{y}_1 : [0, \infty) \rightarrow \Gamma$ and $\mathbf{y}_2 : [0, \infty) \rightarrow \Gamma$ be given by

$$\mathbf{y}_1(t) = P_\Gamma(\mathbf{y}_{\mathbf{z}_1}(t))$$

and

$$\mathbf{y}_2(t) = P_\Gamma(\mathbf{y}_{\mathbf{z}_2}(t))$$

for all $t \geq 0$, respectively. We need to show that $\mathbf{y}_1(t) = \mathbf{y}_2(t)$ for all $t \geq 0$. We compute $\mathbf{y}'_1$. We have the following singular value decomposition for P'_Γ : $P'_\Gamma = U\Sigma U^T$ where

$$U = [\mathbf{t}_1 \ \mathbf{t}_2 \ \mathbf{n}] \text{ and } \Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Since $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{n}$ are orthonormal, for $i, j = 1, 2$,

$$(\mathbf{t}_i \otimes \mathbf{t}_i)(\mathbf{t}_j \otimes \mathbf{t}_j) = \begin{cases} \mathbf{t}_i \otimes \mathbf{t}_i, & i = j, \\ 0, & i \neq j, \end{cases}$$

and for $i = 1, 2$

$$(\mathbf{t}_i \otimes \mathbf{t}_i)(\mathbf{n} \otimes \mathbf{n}) = 0.$$

Using the above, we have

$$\begin{aligned} \mathbf{y}'_1(t) &= (P_\Gamma(\mathbf{y}_{\mathbf{z}_1}(t)))' \\ &= P'_\Gamma \frac{d\mathbf{y}_{\mathbf{z}_1}}{dt}(t) \\ &= P'_\Gamma \bar{f}(\mathbf{y}_{\mathbf{z}_1}(t), \mathbf{a}_1(t)) B(\mathbf{y}_{\mathbf{z}_1}(t), \mu) \mathbf{a}_1(t) \\ &= P'_\Gamma B(\mathbf{y}_{\mathbf{z}_1}(t), \mu) \bar{f}(\mathbf{y}_{\mathbf{z}_1}(t), \mathbf{a}_1(t)) \mathbf{a}_1(t) \\ &= (\mathbf{t}_1 \otimes \mathbf{t}_1 + \mathbf{t}_2 \otimes \mathbf{t}_2) f(\mathbf{y}_1(t), \mathbf{a}_1(t)) \mathbf{a}_1(t) \\ &= f(\mathbf{y}_1(t), \mathbf{a}_1(t)) (\mathbf{t}_1 \otimes \mathbf{t}_1 + \mathbf{t}_2 \otimes \mathbf{t}_2) \mathbf{a}_1(t) \\ &= f(\mathbf{y}_1(t), \mathbf{a}_1(t)) \mathbf{a}_1(t). \end{aligned}$$

The last equality is true since $\mathbf{t}_1 \otimes \mathbf{t}_1 + \mathbf{t}_2 \otimes \mathbf{t}_2$ is the orthogonal projection of $\mathbf{a}_1(t)$ onto the tangent space at $\mathbf{y}_{\mathbf{z}_1}(t)$ and $\mathbf{a}_1(t) \in A_{\mathbf{y}_{\mathbf{z}_1}(t)} \subset T_{\mathbf{y}_{\mathbf{z}_1}(t)}\Gamma$. Thus $\mathbf{y}_1$ is an admissible path on Γ where $\mathbf{y}_1(0) = \mathbf{x} = P_\Gamma(\mathbf{z}_1) = P_\Gamma(\mathbf{z}_2)$ and $\mathbf{a}_1(\cdot) \in \mathcal{A}_{\mathbf{x}} = \mathcal{A}_{P_\Gamma(\mathbf{z}_1)}$. By the same reasoning $\mathbf{y}_2$ also an admissible path with $\mathbf{y}_2(0) = \mathbf{x}$ and $\mathbf{a}_2(\cdot) \in \mathcal{A}_{\mathbf{x}} = \mathcal{A}_{P_\Gamma(\mathbf{z}_2)}$. We assume (A1). Thus, the solution to (5)–(6) is unique. Since $\mathbf{a}_1 \equiv \mathbf{a}_2$, $\mathbf{y}_1(t) = \mathbf{y}_2(t)$ for all $t \geq 0$. $\square$

Now, it easily follows that the value function defined on T_ϵ is the constant normal extension of the value function defined on Γ .

Corollary 2.2 If $u : \Gamma \rightarrow \mathbb{R}$ is the value function on Γ given by (8), then

$$v(\mathbf{z}) = \bar{u}(\mathbf{z}) := u(P_\Gamma(\mathbf{z}))$$

where v is the value function on T_ϵ given by (16).

Proof Let $\mathbf{z} \in T_\epsilon$ and $\mathbf{a}(\cdot) \in \mathcal{A}_{\mathbf{z}}$. Then $\mathbf{a}(\cdot) \in \mathcal{A}_{P_\Gamma(\mathbf{z})}$ since $\mathcal{A}_{\mathbf{z}} = \mathcal{A}_{P_\Gamma(\mathbf{z})}$. Then we have

$$\overline{C}(\mathbf{z}, \mathbf{a}(\cdot)) = \int_0^{\overline{T}} \bar{r}(\mathbf{y}_{\mathbf{z}}(t), \mathbf{a}(t)) dt + \bar{g}(\mathbf{y}_{\mathbf{z}}(\overline{T}))$$

and

$$\overline{C}(P_\Gamma(\mathbf{z}), \mathbf{a}(\cdot)) = \int_0^T \bar{r}(\mathbf{y}_{P_\Gamma(\mathbf{z})}(t), \mathbf{a}(t)) dt + \bar{g}(\mathbf{y}_{P_\Gamma(\mathbf{z})}(T)),$$

where $\mathbf{y}_{\mathbf{z}}(\cdot)$ and $\mathbf{y}_{P_\Gamma(\mathbf{z})}(\cdot)$ are admissible paths on T_ϵ with $\mathbf{y}_{\mathbf{z}}(0) = \mathbf{z}$, $\mathbf{y}_{P_\Gamma(\mathbf{z})}(0) = P_\Gamma(\mathbf{z})$, respectively. Recall, that $\mathbf{y}_{P_\Gamma(\mathbf{z})}$ is also an admissible path on Γ so that $\overline{C}(P_\Gamma(\mathbf{z}), \mathbf{a}(\cdot)) = C(P_\Gamma(\mathbf{z}), \mathbf{a}(\cdot))$.

Theorem 2.1 implies that $P_\Gamma(\mathbf{y}_{\mathbf{z}}(t)) = \mathbf{y}_{P_\Gamma(\mathbf{z})}(t)$ for all $t \geq 0$. Since

$$\overline{T} = \{\mathbf{z} \in T_\epsilon \mid P_\Gamma(\mathbf{z}) \in \mathcal{T}\},$$

we have

$$\min\{t \mid P_\Gamma(\mathbf{y}_{\mathbf{z}}(t)) = \mathbf{y}_{P_\Gamma(\mathbf{z})}(t) \in \mathcal{T}\} = \min\{t \mid \mathbf{y}_{\mathbf{z}}(t) \in \overline{T}\}.$$

Therefore, $\overline{T} = T$, and

$$\begin{aligned} \overline{C}(\mathbf{z}, \mathbf{a}(\cdot)) &= \overline{C}(P_\Gamma(\mathbf{z}), \mathbf{a}(\cdot)) \\ &= C(P_\Gamma(\mathbf{z}), \mathbf{a}(\cdot)). \end{aligned}$$

Finally, we have

$$\begin{aligned} v(\mathbf{z}) &= \inf_{\mathbf{a}(\cdot) \in \mathcal{A}_{\mathbf{z}}} \overline{C}(\mathbf{z}, \mathbf{a}(\cdot)) \\ &= \inf_{\mathbf{a}(\cdot) \in \mathcal{A}_{P_\Gamma(\mathbf{z})}} C(P_\Gamma(\mathbf{z}), \mathbf{a}(\cdot)) \\ &= u(P_\Gamma(\mathbf{z})). \end{aligned}$$

□

Now that we have showed that the value function on the narrow band is equivalent to the value function on the surface, we define the Hamilton–Jacobi–Bellman equation on T_ϵ associated the value function v . Again, the dynamic programming principle states that for sufficiently small $\tau > 0$, we have

$$v(\mathbf{z}) = \inf_{\mathbf{a}(\cdot) \in \mathcal{A}_{\mathbf{z}}} \left\{ \int_0^\tau \bar{r}(\mathbf{y}_{\mathbf{z}}(t), \mathbf{a}(t)) dt + v(\mathbf{y}_{\mathbf{z}}(\tau)) \right\}. \quad (17)$$

The corresponding Hamilton–Jacobi–Bellman equation on T_ϵ is

$$\min_{\mathbf{a} \in \mathcal{A}_{\mathbf{z}}} \left\{ \bar{r}(\mathbf{z}, \mathbf{a}) + \nabla v(\mathbf{z}) \cdot \bar{f}(\mathbf{z}, \mathbf{a}) B(\mathbf{z}, \mu) \mathbf{a} \right\} = 0, \quad \mathbf{z} \in T_\epsilon \setminus \overline{T}. \quad (18)$$

The boundary condition is

$$v(\mathbf{z}) = \bar{g}(\mathbf{z}), \quad \mathbf{z} \in \overline{T}. \quad (19)$$

The assumption (B1) implies that the value function coincides with the unique viscosity solution [2]. Corollary 2.2 implies that the viscosity solution of (18)–(19) is given by

$$v(\mathbf{z}) = u(P_\Gamma(\mathbf{z})), \quad (20)$$

where u is the viscosity solution of (10)–(11) on T_ϵ . An interesting result is that we can prove (20) without Corollary 2.2, i.e., using only the derived HJB equations and properties of viscosity solutions.

We begin with the following lemma that relates the surface gradients for parallel surfaces at equivalent points:

Lemma 2.3 *Assume ϵ is small enough so that P_Γ is differentiable, and let $|\eta| < \epsilon$. Given $\phi_0 \in C^1(\Gamma)$, define $\phi_\eta \in C^1(\Gamma_\eta)$ by $\phi_\eta(\mathbf{z}) := \phi_0(P_\Gamma(\mathbf{z}))$. Then for $\mathbf{z} \in \Gamma_\eta$,*

$$\nabla_\Gamma \phi_0(P_\Gamma(\mathbf{z})) = B(\mathbf{z}, \mu) \nabla_{\Gamma_\eta} \phi_\eta(\mathbf{z}). \quad (21)$$

Proof Let $\bar{\phi}_0 : T_\epsilon \rightarrow \mathbb{R}$ be the constant normal extension of ϕ_0 . Then $\bar{\phi}_0 \in C^1(T_\epsilon)$ and $\bar{\phi}_0|_{\Gamma_\eta} = \phi_\eta$. From Theorem A.1 in [18], we have that $\nabla_\Gamma \phi_0(P_\Gamma(\mathbf{z})) = B(\mathbf{z}, \mu) \nabla \bar{\phi}_0(\mathbf{z})$. Since $\bar{\phi}_0$ is the constant normal extension of ϕ_0 , $\nabla \bar{\phi}_0(\mathbf{z}) \in T_\mathbf{z} \Gamma_\eta$. Thus, if $\mathbf{z} \in \Gamma_\eta$, $\nabla \bar{\phi}_0(\mathbf{z}) = \nabla_{\Gamma_\eta} \phi_\eta(\mathbf{z})$, and (21) holds. $\square$

The following theorem proves (20) directly using the definition of viscosity solutions.

Theorem 2.4 *If $u : \Gamma \rightarrow \mathbb{R}$ is the viscosity solution to (10)–(11), then the constant normal extension of u , $\bar{u} : T_\epsilon \rightarrow \mathbb{R}$, is the viscosity solution to (18)–(19).*

Proof We have $u(\mathbf{x}) = g(\mathbf{x})$ for $\mathbf{x} \in \mathcal{T}$ and $P_\Gamma(\mathbf{z}) \in \mathcal{T}$ if $\mathbf{z} \in \bar{\mathcal{T}}$. Then for $\mathbf{z} \in \bar{\mathcal{T}}$,

$$\bar{u}(\mathbf{z}) = u(P_\Gamma(\mathbf{z})) = g(P_\Gamma(\mathbf{z})) = \bar{g}(\mathbf{z}).$$

Thus, $\bar{u}$ satisfies the boundary conditions (19). Next, we will show that $\bar{u}$ is a viscosity subsolution of (18).

Assume $\mathbf{z}_0 \in T_\epsilon$ and $\phi \in C^1(T_\epsilon)$ such that $\bar{u} - \phi$ has a strict local maximum at $\mathbf{z}_0$ and $(\bar{u} - \phi)(\mathbf{z}_0) = 0$. We need to show that

$$\min_{\mathbf{a} \in A_{\mathbf{z}_0}} \left\{ \bar{r}(\mathbf{z}_0, \mathbf{a}) + \nabla \phi(\mathbf{z}_0) \cdot \bar{f}(\mathbf{z}_0, \mathbf{a}) B(\mathbf{z}_0, \mu) \mathbf{a} \right\} \leq 0. \quad (22)$$

Let $\eta = d_\Gamma(\mathbf{z}_0)$ and $\phi_0 : \Gamma \rightarrow \mathbb{R}$ be the restriction to Γ of the normal extension of $\phi|_{\Gamma_\eta}$. Then we have that $\phi_0 \in C^1(\Gamma)$ and $\phi_\eta := \phi|_{\Gamma_\eta} \in C^1(\Gamma_\eta)$ such that $\phi_\eta(\mathbf{z}) = \phi_0(P_\Gamma(\mathbf{z}))$ for $\mathbf{z} \in \Gamma_\eta$.

First, we will prove

$$B(\mathbf{z}_0, \mu) \nabla \phi(\mathbf{z}_0) \cdot \mathbf{a} = B(\mathbf{z}_0, \mu) \nabla_{\Gamma_\eta} \phi_\eta(\mathbf{z}_0) \cdot \mathbf{a}. \quad (23)$$

Since $\mathbf{a} \in A_{\mathbf{z}_0} \subset T_{\mathbf{z}_0} \Gamma$, we have $\mathbf{a} \cdot \mathbf{n} = 0$ and

$$B(\mathbf{z}_0, \mu) \mathbf{n} \otimes \mathbf{n} \nabla \phi(\mathbf{z}_0) \cdot \mathbf{a} = 0.$$

Now, $\nabla_{\Gamma_\eta} \phi_\eta(\mathbf{z}_0) = (I - \mathbf{n} \otimes \mathbf{n}) \nabla \phi(\mathbf{z}_0)$. Therefore, (23) holds.

Lemma 2.3 implies

$$B(\mathbf{z}_0, \mu) \nabla_{\Gamma_\eta} \phi_\eta(\mathbf{z}_0) = \nabla_\Gamma \phi_0(P_\Gamma(\mathbf{z}_0)). \quad (24)$$

Let $\mathbf{x}_0 = P_\Gamma(\mathbf{z}_0)$. Then using the fact that $B(\mathbf{z}_0, \mu)$ is symmetric, (23), and (24), we have

$$\begin{aligned} & \min_{\mathbf{a} \in A_{\mathbf{z}_0}} \left\{ \bar{r}(\mathbf{z}_0, \mathbf{a}) + \nabla \phi(\mathbf{z}_0) \cdot \bar{f}(\mathbf{z}_0, \mathbf{a}) B(\mathbf{z}_0, \mu) \mathbf{a} \right\} \\ &= \min_{\mathbf{a} \in A_{\mathbf{z}_0}} \left\{ \bar{r}(\mathbf{z}_0, \mathbf{a}) + B(\mathbf{z}_0, \mu) \nabla \phi(\mathbf{z}_0) \cdot \bar{f}(\mathbf{z}_0, \mathbf{a}) \mathbf{a} \right\} \end{aligned}$$

$$\begin{aligned}
&= \min_{\mathbf{a} \in A_{\mathbf{z}_0}} \left\{ \bar{r}(\mathbf{z}_0, \mathbf{a}) + B(\mathbf{z}_0, \mu) \nabla_{\Gamma_\eta} \phi_\eta(\mathbf{z}_0) \cdot \bar{f}(\mathbf{z}_0, \mathbf{a}) \mathbf{a} \right\} \\
&= \min_{\mathbf{a} \in A_{\mathbf{z}_0}} \left\{ \bar{r}(\mathbf{z}_0, \mathbf{a}) + \nabla_\Gamma \phi_0(P_\Gamma(\mathbf{z}_0)) \cdot \bar{f}(\mathbf{z}_0, \mathbf{a}) \mathbf{a} \right\} \\
&= \min_{\mathbf{a} \in A_{\mathbf{x}_0}} \left\{ r(\mathbf{x}_0, \mathbf{a}) + \nabla_\Gamma \phi_0(\mathbf{x}_0) \cdot f(\mathbf{x}_0, \mathbf{a}) \mathbf{a} \right\}.
\end{aligned}$$

Now, $\mathbf{x}_0 \in \Gamma$, $\phi_0 \in C^1(\Gamma)$ such that $u - \phi_0$ has a strict local maximum at $\mathbf{x}_0$ and

$$(u - \phi_0)(\mathbf{x}_0) = (\bar{u} - \phi)(\mathbf{z}_0) = 0.$$

Since u is a viscosity subsolution of (10),

$$\min_{\mathbf{a} \in A_{\mathbf{x}_0}} \left\{ r(\mathbf{x}_0, \mathbf{a}) + \nabla_\Gamma \phi_0(\mathbf{x}_0) \cdot f(\mathbf{x}_0, \mathbf{a}) \mathbf{a} \right\} \leq 0.$$

Thus, (22) holds. The same argument can be applied to show that $\bar{u}$ is a viscosity supersolution of (10). $\square$

In the next section, we will present a special case of the above formulations for when the speed and running cost functions are isotropic. The HJB equation on Γ reduces to the Eikonal equation. A noteworthy revelation is that if we ensure that the extended speed and cost functions are isotropic as well, the control values in the extended optimal control problem do not have to be restricted to be in the tangent bundle of Γ , i.e., the extended set of controls is $\tilde{A} = S^{n-1}$. In this setup, the value function is the constant normal extension of the value function on Γ . Before we proceeding, we present a counter example to show that $\bar{u}$ is not necessarily the value function of the extended control problem if the extended speed function $\bar{f}$ is anisotropic and $\tilde{A} = S^{n-1}$.

Example

Consider the following control problems on Γ and T_ϵ : Let

$$\Gamma = (-1, 1) \subset \mathbb{R}$$

and

$$\mathcal{T} = \{1\}.$$

Let $\epsilon > \sqrt{3}/3$, then

$$T_\epsilon = (-1, 1) \times (-\epsilon, \epsilon)$$

and

$$\bar{\mathcal{T}} = \{1\} \times (-\epsilon, \epsilon).$$

For the control problem on Γ , the set of controls is $A = \{(-1, 0), (1, 0)\}$ and $\mathcal{A}$ is the usual set of admissible controls. Define $f : \Gamma \times A \rightarrow \mathbb{R}$ by $f(\mathbf{x}, \mathbf{a}) = \sqrt{3}$ and suppose that we have unit running cost, i.e., $r(\mathbf{x}, \mathbf{a}) = 1$. Then it is clear that $u(0) = \sqrt{3}/3$ which is the minimum travel time from $\mathbf{x} = 0$ to $\mathcal{T}$ while traveling at speed $\sqrt{3}$.

Now for the extended control problem, let the control values be the set $\tilde{A} = S^1$. Then the set of admissible controls is given by

$$\tilde{\mathcal{A}} := \{ \text{measurable functions } \tilde{a} : [0, \infty) \rightarrow \tilde{A} \}.$$

Define $\bar{f} : T_\epsilon \times \tilde{\mathcal{A}} \rightarrow \mathbb{R}$ such that $\bar{f}(\mathbf{z}, \tilde{\mathbf{a}}) = \bar{f}(\tilde{\mathbf{a}})$ and the set $\{\bar{f}(\tilde{\mathbf{a}})\tilde{\mathbf{a}} \mid \tilde{\mathbf{a}} \in \tilde{\mathcal{A}}\}$ is the ellipse given by $x^2 + \frac{y^2}{9} = 1$ rotated at the origin clockwise by the angle $\pi/3$. Then $\bar{f}(\mathbf{a}) = \sqrt{3}$ if $\mathbf{a} \in A$ so that $\bar{f}(\mathbf{z}, \mathbf{a}) = \bar{f}(P_\Gamma \mathbf{z}, \mathbf{a})$ for $\mathbf{a} \in A$. Again, assume unit running cost.

Let v be the value function for this extended control problem. If $v = \bar{u}$, then we should have $v((0, 0)) = u(0)$. However, note that $\bar{f}(\mathbf{z}, (\sqrt{3}/2, 1/2)) = 3$ and the Euclidean distance between $(0, 0)$ and $(1, \sqrt{3}/3)$ is $2\sqrt{3}/3$. Thus,

$$\bar{C}((0, 0), \tilde{a}(\cdot)) = \frac{2\sqrt{3}}{9}$$

when $\tilde{a}(\cdot) \equiv (\sqrt{3}/2, 1/2)$. By the definition of the value function we must have

$$v((0, 0)) \leq \frac{2\sqrt{3}}{9} < \frac{\sqrt{3}}{3} = u(0).$$

Therefore, $\bar{u}$ cannot be the value function for the extended control problem.

3 The Isotropic Case

We derive the optimal control problems and HJB equations corresponding to the case when the speed and cost functions are isotropic. As mentioned at the end of Sect. 2.2, we will show that value function on T_ϵ is the normal extension of the value function on Γ even when we do not restrict the controls to the tangent space in the extended optimal control problem.

3.1 The Controlled Dynamics on Γ for Isotropic Speed and Cost Functions

The optimal control problem formulation is exactly the same as in Sect. 2.1, and (10) reduces to the Eikonal equation on the surface:

$$||\nabla_\Gamma u(\mathbf{x})|| = \frac{r(\mathbf{x})}{f(\mathbf{x})}. \quad (25)$$

We have the HJB equation

$$H(\mathbf{x}, \nabla_\Gamma u(\mathbf{x})) = r(\mathbf{x}) - f(\mathbf{x})||\nabla_\Gamma u(\mathbf{x})|| = 0, \quad \mathbf{x} \in \Gamma, \quad (26)$$

and the boundary condition is

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \mathcal{T}. \quad (27)$$

Assuming (A1), the value function u coincides with the unique viscosity solution of (26)–(27). We also have that $V(\mathbf{x})$ is strictly convex for all $\mathbf{x} \in \Gamma$. Thus, an optimal control is guaranteed.

3.2 Extension to T_ϵ in the Isotropic Case

We derive the extended Hamiltonian, $\bar{H}$, on T_ϵ from the extended isotropic optimal control problem such that if $v : T_\epsilon \rightarrow \mathbb{R}$ is the unique viscosity solution of

$$\bar{H}(\mathbf{x}, \nabla v(\mathbf{z})) = 0$$

with boundary conditions defined appropriately, then

$$v(\mathbf{z}) = u(P_\Gamma(\mathbf{z})),$$

where u is the solution to (26)–(27).

The notations $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{n}$, σ_1 , and σ_2 are as in Sect. 2.2. The extended set of compact control values is $\tilde{\mathcal{A}} = S^{n-1}$, and the set of admissible controls is given by:

$$\tilde{\mathcal{A}} = \{\text{measurable functions: } \tilde{\mathbf{a}} : [0, \infty) \rightarrow \tilde{\mathcal{A}}\}.$$

This is where the formulation differs from the previous section. We do not restrict the controls to belong to the tangent bundle of Γ . We extend the speed, cost, and exit time penalty functions as above: $\bar{f}(\mathbf{z}) = f(P_\Gamma(\mathbf{z}))$, $\bar{r}(\mathbf{z}) = r(P_\Gamma(\mathbf{z}))$, and $\bar{g}(\mathbf{z}) = g(P_\Gamma(\mathbf{z}))$. The dynamical system is given by

$$\frac{d\mathbf{y}}{dt}(t) = \bar{f}(\mathbf{y}(t))B(\mathbf{y}(t), \mu)\tilde{\mathbf{a}}(t), \quad t > 0 \quad (28)$$

$$\mathbf{y}(0) = \mathbf{z}, \quad \mathbf{z} \in T_\epsilon, \quad (29)$$

where $\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}$. Admissible paths on T_ϵ with extended controls are solutions to (28)–(29) denoted by $\mathbf{y}_\mathbf{z}(t)$. Note that the admissible paths are not restricted to the parallel surface, Γ_η , to which the initial point $\mathbf{z}$ belongs.

The total cost function associated to the initial state $\mathbf{z} \in T_\epsilon$ and control $\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}$ is the same as in Sect. 2.2. The value function $v : T_\epsilon \rightarrow \mathbb{R}$ is the minimal total cost to $\bar{T}$ starting at $\mathbf{z}$ and is given by

$$v(\mathbf{z}) = \inf_{\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}} \bar{C}(\mathbf{z}, \tilde{\mathbf{a}}(\cdot)). \quad (30)$$

We have that (A1) implies the following is true:

$$\left\{ \begin{array}{l} \text{The mapping } (\mathbf{z}, \tilde{\mathbf{a}}) \mapsto \bar{f}(\mathbf{z})B(\mathbf{z}, \mu)\tilde{\mathbf{a}} \text{ is Lipschitz continuous} \\ \text{in the first argument and, } \bar{r} \text{ is Lipschitz continuous.} \\ \bar{g} \text{ is lower semicontinuous and } \min_{\mathcal{T}} \bar{g} < \infty. \\ 0 < R_1 \leq \bar{r}(\mathbf{z}) \leq R_2, \text{ for all } \mathbf{z} \in T_\epsilon. \\ \text{There exists constants } \bar{F}_1, \bar{F}_2 \text{ such that:} \\ 0 < \bar{F}_1 \leq \|\bar{f}(\mathbf{z})B(\mathbf{z}, \mu)\tilde{\mathbf{a}}\| \leq \bar{F}_2, \text{ for all } \mathbf{z} \in T_\epsilon \text{ and } \tilde{\mathbf{a}} \in \tilde{\mathcal{A}}. \end{array} \right. \quad (\text{C1})$$

We have that (C1) implies the value function, v , coincides with the unique viscosity solution of (26)–(27) [2]. We also have that $\bar{V}(\mathbf{z}) = \{\bar{f}(\mathbf{z})\tilde{\mathbf{a}}/\bar{r}(\mathbf{z}) \mid \tilde{\mathbf{a}} \in \tilde{\mathcal{A}}\}$ is strictly convex for all $\mathbf{z} \in \Gamma$. Thus, an optimal control is guaranteed.

We now prove that the value function, v , is a constant extension of a function on Γ . However, unlike in the previous section, it is not immediate from the following proofs that $v = \bar{u}$, where u is the value function on Γ .

Theorem 3.1 *Given $\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}$ and $\mathbf{z}_1, \mathbf{z}_2 \in T_\epsilon$ such that $P_\Gamma(\mathbf{z}_1) = P_\Gamma(\mathbf{z}_2)$, let*

$$\mathbf{y}_{\mathbf{z}_1} : [0, \infty) \rightarrow T_\epsilon$$

and

$$\mathbf{y}_{\mathbf{z}_2} : [0, \infty) \rightarrow T_\epsilon$$

solve (28) with $\mathbf{y}_{\mathbf{z}_1}(0) = \mathbf{z}_1$ and $\mathbf{y}_{\mathbf{z}_2}(0) = \mathbf{z}_2$, respectively. Then $P_\Gamma(\mathbf{y}_{\mathbf{z}_1}(t)) = P_\Gamma(\mathbf{y}_{\mathbf{z}_2}(t))$ for all $t \geq 0$.

The proof of Theorem 3.1 is analogous to the proof of Theorem 2.1 except the control $\tilde{\mathbf{a}}(\cdot)$ belongs to the extended control space, $\tilde{\mathcal{A}}$. A consequence of the control belonging to

the extended space is that now the projected paths $P_\Gamma \mathbf{y}_z(\cdot)$ may not solve (5). This is due to the fact that $(\mathbf{t}_1 \otimes \mathbf{t}_1 + \mathbf{t}_2 \otimes \mathbf{t}_2) \tilde{\mathbf{a}}(t) \neq \tilde{\mathbf{a}}(t)$ if $\tilde{\mathbf{a}}(t) \notin T_{P_\Gamma(\mathbf{y}_z(t))}$ where $\mathbf{t}_1$ and $\mathbf{t}_2$ are the basis tangent vectors of $T_{P_\Gamma(\mathbf{y}_z(t))}$. We now prove that the extended value function in the isotropic case is constant along the normals of Γ .

Corollary 3.2 *If $P_\Gamma(\mathbf{z}_1) = P_\Gamma(\mathbf{z}_2)$ for $\mathbf{z}_1, \mathbf{z}_2 \in T_\epsilon$, then $v(\mathbf{z}_1) = v(\mathbf{z}_2)$.*

Proof Let $\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}$. We have

$$\bar{C}(\mathbf{z}_1, \tilde{\mathbf{a}}(\cdot)) = \int_0^{\bar{T}_1} \bar{r}(\mathbf{y}_{z_1}(t)) dt + \bar{g}(\mathbf{y}_{z_1}(\bar{T}_1))$$

and

$$\bar{C}(\mathbf{z}_2, \tilde{\mathbf{a}}(\cdot)) = \int_0^{\bar{T}_2} \bar{r}(\mathbf{y}_{z_2}(t)) dt + \bar{g}(\mathbf{y}_{z_2}(\bar{T}_2)),$$

where $\mathbf{y}_{z_1}(\cdot)$ and $\mathbf{y}_{z_2}(\cdot)$ are admissible paths on T_ϵ with extended control, $\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}$, and $\mathbf{y}_{z_1}(0) = \mathbf{z}_1, \mathbf{y}_{z_2}(0) = \mathbf{z}_2$ where $P_\Gamma(\mathbf{z}_1) = P_\Gamma(\mathbf{z}_2)$.

Theorem 3.1 implies that $P_\Gamma(\mathbf{y}_{z_1}(t)) = P_\Gamma(\mathbf{y}_{z_2}(t))$ for all $t \geq 0$. Since

$$\bar{T} = \{\mathbf{z} \in T_\epsilon \mid P_\Gamma(\mathbf{z}) \in \mathcal{T}\},$$

we have

$$\min\{t \mid P_\Gamma(\mathbf{y}_{z_1}(t)) = P_\Gamma(\mathbf{y}_{z_2}(t)) \in \mathcal{T}\} = \min\{t \mid \mathbf{y}_{z_1}(t) \in \bar{T}\} = \min\{t \mid \mathbf{y}_{z_2}(t) \in \bar{T}\}.$$

Therefore, $\bar{T}_1 = \bar{T}_2$ and $\bar{C}(\mathbf{z}_1, \tilde{\mathbf{a}}(\cdot)) = \bar{C}(\mathbf{z}_2, \tilde{\mathbf{a}}(\cdot))$ for $\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}$. Now,

$$v(\mathbf{z}_1) = \inf_{\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}} \bar{C}(\mathbf{z}_1, \tilde{\mathbf{a}}(\cdot)) = \inf_{\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}} \bar{C}(\mathbf{z}_2, \tilde{\mathbf{a}}(\cdot)) = v(\mathbf{z}_2).$$

□

In the above proof, it is not immediate that we can take the infimum over $\mathcal{A} \subset \tilde{\mathcal{A}}$, which would imply that v is the normal extension of u , the value function on Γ . We will use the corresponding HJB equations and Corollary 3.2 to show that $v = \bar{u}$.

We define the HJB equation on T_ϵ associated the value function v . Again, the dynamic programming principle states that for sufficiently small $\tau > 0$ we have

$$v(\mathbf{z}) = \inf_{\tilde{\mathbf{a}}(\cdot) \in \tilde{\mathcal{A}}} \left\{ \int_0^\tau \bar{r}(\mathbf{y}_z(t)) dt + v(\mathbf{y}_z(\tau)) \right\}. \quad (31)$$

We get the Hamilton–Jacobi–Bellman equation on T_ϵ :

$$\min_{\tilde{\mathbf{a}} \in \tilde{\mathcal{A}}} \left\{ \bar{r}(\mathbf{z}) + \nabla v(\mathbf{z}) \cdot B(\mathbf{z}, \mu) \bar{f}(\mathbf{z}) \tilde{\mathbf{a}} \right\} = 0. \quad (32)$$

Since the speed and cost functions are isotropic, (32) reduces to an anisotropic Eikonal equation:

$$\bar{H}(\mathbf{z}, \nabla v(\mathbf{z})) = \bar{r}(\mathbf{z}) - \bar{f}(\mathbf{z}) \|B(\mathbf{z}, \mu) \nabla v(\mathbf{z})\| = 0. \quad (33)$$

The boundary condition is

$$v(\mathbf{z}) = \bar{g}(\mathbf{z}), \quad \mathbf{z} \in \bar{T}. \quad (34)$$

Since the value function coincides with the unique viscosity solution, Corollary 3.2 implies that the viscosity solution of (33)–(34) is a constant along the normals of Γ , i.e., there is a function $w : \Gamma \rightarrow \mathbb{R}$ such that for $\mathbf{z} \in T_\epsilon$,

$$v(\mathbf{z}) = \bar{w}(\mathbf{z}) := w(P_\Gamma(\mathbf{z})).$$

We now prove that $w = u$ where u is the viscosity solution of (26)–(27).

Theorem 3.3 *If $u : \Gamma \rightarrow \mathbb{R}$ is the viscosity solution to (26)–(27), then the constant normal extension of u , $\bar{u} : T_\epsilon \rightarrow \mathbb{R}$, is the viscosity solution to (33)–(34).*

Proof Just as in Theorem 2.4, $\bar{u}$ satisfies the boundary conditions (34). Assume for contradiction that $\bar{u}$ is not the viscosity solution to (33). But from Corollary 3.2 we have that viscosity solution of (33)–(34), $v = \bar{w}$ for some $w \in C^1(\Gamma)$. Therefore, if $\bar{u}$ is not the unique viscosity solution then we have $u \neq w$. For contradiction, we will show that w is the viscosity solution of (26) and thus, $u = w$.

Let $\mathbf{x}_0 \in \Gamma$ and $\phi \in C^1(\Gamma)$ such that $w - \phi$ has a local maximum at $\mathbf{x}_0$. Consider the normal extensions of w and ϕ , then $\bar{w} - \bar{\phi}$ has a local max at $\mathbf{x}_0$. Since $\bar{w}$ is a viscosity subsolution to (33) and $\bar{\phi} \in C^1(T_\epsilon)$, we have that

$$\bar{r}(\mathbf{x}_0) - \bar{f}(\mathbf{x}_0) + B(\mathbf{x}_0, \mu) \nabla \bar{\phi}(\mathbf{x}_0) \leq 0. \quad (35)$$

Since $\mathbf{x}_0 \in \Gamma$, $\sigma_1 = \sigma_2 = 1$ and $\nabla \bar{\phi}(\mathbf{x}_0) \in T_{\mathbf{x}_0} \Gamma$. Therefore,

$$B(\mathbf{x}_0, \mu) = \mathbf{t}_1 \otimes \mathbf{t}_1 + \mathbf{t}_2 \otimes \mathbf{t}_2 + \mu \mathbf{n} \otimes \mathbf{n},$$

and we have that $B(\mathbf{x}_0, \mu) \nabla \bar{\phi}(\mathbf{x}_0) = \nabla \bar{\phi}(\mathbf{x}_0) = \nabla_\Gamma \phi(\mathbf{x}_0)$. Thus,

$$r(\mathbf{x}_0) - f(\mathbf{x}_0) + \nabla_\Gamma \phi(\mathbf{x}_0) \leq 0, \quad (36)$$

and w is a viscosity subsolution. The same argument can be applied to show that w is a viscosity supersolution. Therefore, w is the viscosity solution to (26) and hence $w = u$. $\square$

To summarize, we have shown that we may appropriately extend the control space in the case of isotropic speed and running costs. This has the implication that even though the admissible paths on T_ϵ have the “choice” to leave the tangent space of Γ , the optimal paths in the extended control problem remain on or parallel to Γ .

4 Numerical Implementation and Simulations

4.1 Numerical Setup

In the implementation of our new framework, we use a uniform Cartesian grids and denote the grid step size by h . T_ϵ^h denotes the part of the grid in the narrow band of radius ϵ around the given surface.

We use the Lax-Friedrichs fast sweeping methods in [16] and a high order version in [39] for our simulations. It is also possible to an upwind scheme generalized from [36]. Unless we mention otherwise, we use the standard first order finite differencing in the approximation of the partial derivatives for the Lax-Friedrichs numerical Hamiltonians.¹

¹ All code used to produce the numerical simulations can be found at <https://github.com/lindsmart/MartinTsaiExtHJB>.

We present several examples of our new formulation performed on surfaces of co-dimension one in three dimensions. For the first three examples, we compute the solution to the Eikonal equation on the surface. As shown in Sect. 3, the equivalent equation on T_ϵ is

$$||B(\mathbf{z}, \mu) \nabla v(\mathbf{z})|| = 1, \quad \mathbf{z} \in T_\epsilon \setminus \overline{T} \quad (37)$$

$$v(\mathbf{z}) = \bar{g}(\mathbf{z}), \quad \mathbf{z} \in \overline{T}. \quad (38)$$

Again,

$$B(\mathbf{z}, \mu) = \sigma_1^{-1} \mathbf{t}_1 \otimes \mathbf{t}_1 + \sigma_2^{-1} \mathbf{t}_2 \otimes \mathbf{t}_2 + \mu \mathbf{n} \otimes \mathbf{n}.$$

Since the desired solution, v , has been proven to be constant along the normals of the surface, Γ , $\mathbf{n} \otimes \mathbf{n} \nabla v(\mathbf{z}) = 0$. Therefore, μ can be any real number. We let $\mu = 1$ in all of our computations.

Approximating the solution to (37)–(38) requires the computation of the singular values and vectors of the derivative of the closest point mapping, P'_Γ . We defer the discussion of these approximations to Sect. 4.1.4. In the last example, we apply our framework to solve an HJB equation with an anisotropic speed function.

4.1.1 Boundary Closure

We note that the analytical formulation of the HJB equations does not require boundary conditions on ∂T_ϵ . However, since we are using Cartesian grids we must take careful consideration of the discretization near the boundary, ∂T_ϵ^h . When approximating the partial derivatives of v , a neighboring grid node may lie outside of T_ϵ^h . We will call these ghost nodes. In our implementation, we provide a boundary closure procedure to enforce the fact that the solution is a constant along normal function. For each ghost node, we perform the following procedure:

- (1) Project the node into the narrow band.
- (2) The value at the ghost node is then calculated by interpolating grid values surrounding the projected point. Formally, we must use an interpolation scheme of order higher to the discretization of the PDE on T_ϵ^h .

Next, we describe how the boundary closure procedure affects the numerical accuracy. Let $\mathbf{z}_i$ be a ghost node, and $\mathbf{z}_i^\alpha = \mathbf{z}_i - \alpha \mathbf{n}_i$ be the projection of the ghost node into T_ϵ . Denote the solution at $\mathbf{z}_i^\alpha$ by $v_i^\alpha := v(\mathbf{z}_i^\alpha)$. Suppose that we use a first order scheme to discretize the HJB equation on T_ϵ^h . Then the interpolation used in the approximation of v_i^α should yield at least second order in h accuracy, in order to maintain formally the first order accuracy. This is due to the amplification by a factor of h^{-1} in the finite difference scheme errors of the approximation of the values at the neighboring grid nodes of $\mathbf{z}_i$.

4.1.2 Depth of Projected Points and Bounding the Thickness of the Narrow Band

When choosing α for the the projected ghost node discussed above, a necessary condition is that the nearby surrounding nodes used to interpolate the value of v_i^α must be in T_ϵ^h . In our numerical simulations, we use cubic interpolation to approximate $v(\mathbf{z}_i^\alpha)$. Thus, it requires 4^3 nearby grid nodes for three dimensional cases. It can be shown that if $|d_\Gamma(\mathbf{z}_i) - \alpha| \leq \epsilon - 2\sqrt{n}h$, where n is the dimension. Then the inner nodes used to interpolate the value at each ghost node are in T_ϵ^h . We suggest to use a projected node as close to the boundary node as possible.

Therefore in our numerical simulations, we choose $\alpha = d_\Gamma(\mathbf{z}_i) - (\epsilon - 2\sqrt{nh})$. We discuss the effect of the choice of α in Sect. 4.2.

The maximal value of ϵ is restricted by the curvatures of Γ . We must have ϵ smaller than the reach of the surface, which depends on the reciprocal of the maximal curvatures. The minimal value of ϵ is determined by the finite difference stencils in approximating the partial derivatives of the PDE and the boundary closure procedure. Both stencils require that the narrow band is sufficiently “thick” relative to the mesh size h with $\epsilon > 2\sqrt{nh}$.

The resulting numerical discretization of the HBJ equation on T_ϵ^h is convergent for any ϵ in the interval described above; i.e. it is convergent for very thin $\mathcal{O}(h)$ narrow bands as well as for thicker one whose widths are independent of h . We remark that this convergent regime is very different that requiring $\epsilon \sim \mathcal{O}(h^p)$, $0 < p < 1$, in the related work of [18], dealing with singular integrals.

4.1.3 Approximation Errors

The error computed by the proposed algorithm can be written as the sum of errors corresponding to different approximations:

$$\text{Error} = E_{\text{model}}(\epsilon) + E_\Delta(h) + E_\Gamma(h) + E_{L_p}(h, \epsilon), \quad (39)$$

where

- E_{model} relates to how the surface PDE is approximated in the tubular neighborhood by an extended PDE and the boundary conditions. Our main contribution is in deriving the extension for which $E_{\text{model}} \equiv 0$ for any ϵ that is smaller than the reach of the surface;
- E_Δ corresponds to the numerical error for approximating the extended PDE problem. Our extension allows for higher order (in h) methods to be applied;
- E_Γ relates to errors in approximating the surface and its geometric properties. This error corresponds to approximation of the closest point mapping to the surface and its curvatures. Since these quantities are computed based on finite differencing using the grid, the error depends on h ;
- E_{L_p} is the numerical error in approximating the L_p norm of the computed errors. Our formulation allows for a very accurate approximation to this error term.

4.1.4 Application to Solving HJB on Point Clouds

We now describe how we compute the closest point mapping when the surface is represented as a point cloud. (See Fig. 1.) Denote the point cloud by $\Gamma_K \subset \Gamma$, with K indicating the total number of points in the set. Here, we shall assume that the K points uniformly distribute on the surface. The purpose of the exposition here is to point out a promising application of the method to point cloud data. A full analysis of the algorithm in this area warrants a separate paper.

We approximate P_Γ using the following strategy:

- (1) For each $\mathbf{z}_i \in T_\epsilon^h$, estimate Γ_K locally on the grid nodes. Call the local surface associated to the point $\mathbf{z}_i$, $\Gamma_K(\mathbf{z}_i)$.
- (2) Compute the closest point of $\mathbf{z}_i$ on the local surface, $\Gamma_K(\mathbf{z}_i)$.

In our implementation, we use biquadratic interpolation at each $\mathbf{z}_i$ to compute $\Gamma_N(\mathbf{z}_i)$. Another option to locally estimate the surface is least squares as in [23]. We then use Newton’s

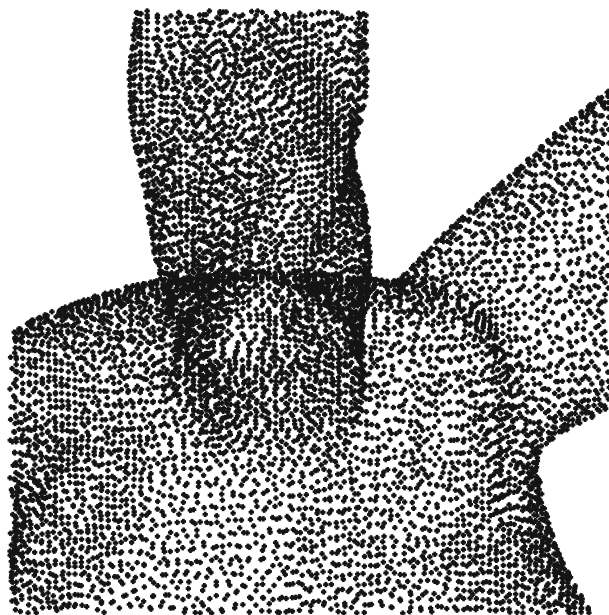


Fig. 1 Example of a point cloud for the Stanford bunny zoomed in near the top of the head.

method to obtain the closest point of each $\mathbf{z}_i$ on $\Gamma_N(\mathbf{z}_i)$. Denote the closest point mappings by $P_{\Gamma_N}^{\text{int}}(\mathbf{z}_i)$. Then on each grid node $\mathbf{z}_i$, the equation is discretized as usual, with $P'_{\Gamma}(\mathbf{z}_i)$ being approximated by finite differences of $P_{\Gamma_N}^{\text{int}}(\mathbf{z}_i)$. We use a fourth-order finite difference scheme to estimate P'_{Γ} and singular value decomposition to obtain the singular values and singular vectors of P'_{Γ} . We show the effect of this approximation procedure in Sect. 4.2.

In any case, the term, E_{Γ} , in (39) now depends on $1/K$, which corresponds to the uniform spacing of the points. Therefore, with a fixed point cloud, refinement of Cartesian grids eventually will not further improve the accuracy in the numerical solutions when compared to the solution on the idealized smooth surface sampled by the point cloud. In Sect. 4.3, we present some results revealing the effect of different surface sampling densities.

If noise is introduced to the point cloud, and if the amount of perturbation in the original “surface normals” is significant, more sophisticated surface fitting is required; e.g. adopting a similar strategy proposed in [23]. We will investigate this important issue in a future manuscript. Nevertheless, in Sect. 4.3, we present some results revealing the effect of noise in the point cloud.

4.2 Example 1.1

First, we present a numerical convergence study. We approximate the solution to (37)–(38), where Γ is a sphere centered at $(0, 0, 0)$, with radius, $r_0 = 0.5$, and $\mathcal{T}$ is a point on the sphere. The exact singular values and vectors in $B(\mathbf{z}, 1)$ are used. In the case of a sphere

$$\sigma_1 = \sigma_2 = 1 - \frac{r(\mathbf{z}) - r_0}{r(\mathbf{z})} = \frac{r_0}{r(\mathbf{z})}, \quad (40)$$

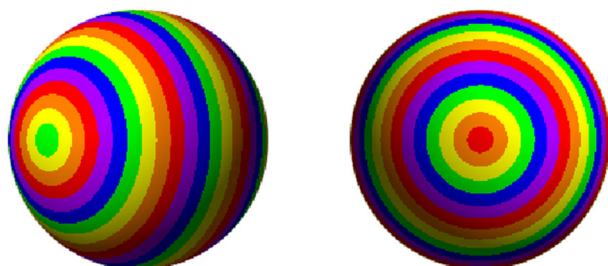


Fig. 2 Two view points of the distance function on a sphere.

where $r(\mathbf{z})$ is the radius of the sphere going through $\mathbf{z}$ centered at $(0, 0, 0)$. Figure 2 shows the distance function to a point on the sphere for two different view points.

Since the solution is constant along the normals of Γ , we can easily estimate the L_1 error. Using the formulation in [18], the estimation of the L_1 error is given by the following formula:

$$\|u - u^h\|_{L_1(\Gamma)} \approx \sum_{\mathbf{z}_i \in T_\epsilon^h} (u(P_\Gamma(\mathbf{z}_i)) - v_i^h) K_\epsilon(d_\Gamma(\mathbf{z}_i)) J(\mathbf{z}_i) h^3,$$

where u is the exact solution and u^h is the approximated solution on Γ using the approximate solution, v_i^h , of the extended equation with grid size h . Here,

$$K_\epsilon(d_\Gamma(\mathbf{z}_i)) = A * \exp\left(\frac{2}{d_\Gamma(\mathbf{z}_i)^2 - 1}\right)$$

where $A = 7.513931532835806$ and

$$J(\mathbf{z}_i) = \sigma_1 \sigma_2,$$

where σ_1, σ_2 are given in (40).

Because all the characteristics of the solution emanate from the source point on the sphere, the errors at the source point will propagate through the solution in the entire computational domain. Therefore, we initialize the solution near $\overline{T}$ with the exact solution if $u(\mathbf{z}) < 0.2$. The L_1 and L_∞ errors are reported in Table 1. We see that first order convergence is achieved in the L_1 error. However, because of the singularity of the solution at the point opposite the source point on the sphere, the order in the L_∞ error is less than one. The results verify that the boundary closure procedure does not influence the overall order of the scheme.

We report the number of iterations it takes for the L_∞ norm of the difference in the solution between successive iterations is less than $1.0e - 13$. One iteration includes one sweep of the grid in each of the eight sweeping directions. The timing for the numerical experiments scales linearly to the number of grid nodes in the narrow band, T_ϵ , and the number of iterations used in the Lax-Friedrichs scheme. The computations were performed on a MacBook laptop with a 1.6 GHz Intel CPU on a single core using Julia. One Lax-Friedrichs iteration took on average 0.34260 seconds for the case $N = 101$ in Table 1. We note that our code was not optimized for efficiency.

Another advantage of our setup is that we can use existing high order methods to compute the solution to HJB equations on surfaces. In Table 2, we present the L_1 and L_∞ errors and orders for the third order method given in [39]. We initialize the solution near $\overline{T}$ to the exact solution if $u(\mathbf{z}) < 0.2$. We see that third order is achieved in the L_1 error. Again, because of the singularity in the solution at the point opposite the source point on the sphere, the order in the L_∞ error is first order.

Table 1 Errors and orders of accuracy for the distance function on a sphere, computed by the first order Lax-Friedrichs sweeping algorithm. We use $h = 2/(N - 1)$ and $\epsilon = 4h$

N	Number of points in T_ϵ	L_1 error	order	L_∞ error	Order	Iterations
101	63302	0.02319		0.03741		32
201	252310	0.01138	1.0345	0.02213	0.76291	49
301	566458	7.5443e-3	1.0158	0.01611	0.78624	65
401	1571974	5.6562e-3	1.0041	0.01280	0.79978	80

Table 2 Errors and orders of accuracy for the distance function on a sphere, computed by the third order Lax-Friedrichs sweeping algorithm. We use $h = 2/(N - 1)$ and $\epsilon = 11h$

N	Number of points in T_ϵ	L_1 error	order	L_∞ error	Order	Iterations
101	183810	5.1129e-4		0.02779		65
201	702626	6.6194e-5	2.9706	0.01137	1.2987	92
301	1566014	2.0373e-5	2.9182	7.1199e-3	1.1587	100
401	2776370	8.5992e-6	3.0069	5.1806e-3	1.1085	119

4.3 Example 1.2

In this example, we compute the distance function on the sphere when the sphere is represented as a point cloud. We show the effect of our method when using different uniform samplings of the sphere.² The results are tabulated in Table 3 when the sampling of the sphere has 400, 800, and 1600 points. We again initialize the solution by exact solutions around the same neighborhood of $\bar{T}$ as in Sect. 4.2. By comparing the L_1 error from Table 1 for $N = 101$ and the L_1 error in Table 3 for $N = 101$ and no noise, it appears that the dominating source of error is the numerical discretization of the PDE. As the grid is refined, we see that it appears the error related to approximating the surface begins to dominate. This is apparent by comparing the L_1 error from Table 1 for $N = 301$ and the L_1 error from Table 3 for $N = 301$ with no noise. Notice as the point cloud becomes more dense and the accuracy of the surface approximation increases, it appears the dominating error is again due to the numerical discretization of the PDE.

In Table 3, we also consider when noise is introduced to the point clouds of different sampling densities. The noise is applied by adding $\delta * (-1 + 2 * \text{rand}()) / 2$ to each point in the point clouds, where $\delta = 0.001, 0.005$ and $\text{rand}()$ is a uniformly distributed random number in $(0, 1)$. We can see if the noise is too large the error begins to deteriorate, and we need a more sophisticated surface fitting algorithm in our method.

4.4 Example 1.3

Next, we study the affect that the depth of projected ghost nodes has on the overall error for the distance function on the sphere. The set up is the same as in the previous section. Here, we choose a 101^3 sized grid with $\epsilon = 11h$ where $h = 2/100$. We estimate the L_1 error for when the boundary closure procedure is carried out at four different depths. Recall that a

² The uniform samplings of the sphere were computed using the code provided at <https://github.com/AntonSemechko/S2-Sampling-Toolbox>.

Table 3 L_1 errors for the first order Lax-Friedrichs scheme using different densities of uniformly sampled point cloud representations of the sphere and different noises applied to the point clouds. Here, K is the number of points in the point cloud

δ	N	L_1 errors using Γ_K		
		$K = 400$	$K = 800$	$K = 1600$
0	101	0.02332	0.02317	0.02318
	201	0.01172	0.01144	0.01139
	301	8.0413e-3	7.6406e-3	7.5649e-3
0.001	101	0.02349	0.02321	0.02313
	201	0.01192	0.01153	0.01148
	301	8.2866e-3	7.7936e-3	7.75321e-3
0.005	101	0.02464	0.02364	0.02404
	201	0.01389	0.01339	0.01531
	301	0.01082	0.01265	0.01279

The noise is applied by adding $\delta * (-1 + 2 * \text{rand}()) / 2$ to each point in the point clouds. We use $h = 2 / (N - 1)$ and $\epsilon = 4h$

Table 4 Comparison of different choices of α for the sphere on a 101^3 grid. Here, $\epsilon = 11h$ and $h = 2/100$.

α	$d(\mathbf{z}) - (11 - 2\sqrt{3})h$	$d(\mathbf{z}) - 3h$	$d(\mathbf{z})$	$d(\mathbf{z}) + 3h$
L_1 error	0.02298	0.02303	0.02320	0.02349

ghost node, $\mathbf{z}_i$, is outside of T_ϵ^h and is projected into T_ϵ along the normal of the surface at that point, i.e.,

$$\mathbf{z}_i \mapsto \mathbf{z}_i - \alpha \mathbf{n}_i.$$

Table 4 shows the L_1 error for the depths:

$$\alpha = d(\mathbf{z}) - (10 - 2\sqrt{3})h, d(\mathbf{z}) - 3h, d(\mathbf{z}), d(\mathbf{z}) + 3h.$$

We can see that the error the solution is not very sensitive to the choice of depth. In our simulations, we chose $\alpha = \epsilon - 2\sqrt{3}h$.

4.5 Example 1.4

Finally, we compare the following: (1) our method on the sphere, using exact singular values and vectors as in the set up for Table 1; (2) our method with the point cloud procedure described in 4.1.4 and finite differences to estimate P'_Γ ; (3) the method in [27]. Since method (3) requires $\epsilon = Ch^\gamma$ where $C = \sqrt{3}$ and $\gamma \in (0, 1)$, to fairly compare the error of the methods, we increase the width of the narrow band to $h = 2h^{0.7}$, which is used in the convergence analysis of (3) in [27]. We initialize the experiments using methods (1), (2), and (3) with exact values given in a box of length h around the source point/points in the initial sets. We also are not able to compute the L_1 error for method (3) since the solution is not constant along the normals of Γ . Therefore, we can only report the L_∞ error, and we use trilinear interpolation to approximate the solutions on Γ for (3). The errors are reported in Table 5. We can also see that estimating P_Γ from a point cloud and P'_Γ from finite differences does not greatly influence the L_1 error of the method.

Table 5 Comparison of methods on a 101^3 grid: (1) Our method using exact closest point mapping and exact singular values and vectors; (2) Our method using closest point mapping sampled from an equally distributed point cloud of 400 points and finite differences to compute singular values and vectors; (3) method in [27]. Here, $\epsilon = 2h^{0.7}$ and $h = 2/100$.

Method	(1)	(2)	(3)
L_∞ error	0.03226	0.03334	0.17072
L_1 error	0.07787	0.07667	

4.6 Example 2

In Figs. 3, 4, and 5, the distance function to a source point is shown on various surfaces. The contours shown are equally spaced and parallel. The solutions for the torus and bunny are computed on a 201^3 grid, and a 101^3 grid is used for the elephant, mask, human skull, and dinosaur skull. All computations use a narrow band width of $\epsilon = 4h$ and point cloud representations for the surfaces. The number of points in the point clouds of the torus³ and bunny⁴ are 178,350 and 228,096, respectively. For the elephant, mask, human skull, and dinosaur skull⁵, we used point clouds with 65,292, 1,199,988, 234,618, and 139,491 points, respectively. In Fig. 3, a cross section of the solution is displayed to show that the solution is indeed constant along the normals of the surface. Our new framework also allows us to sort point clouds. Figure 3 displays level “belts” of the point cloud, i.e., points in the point cloud whose distance from the source point lies in given interval.

Another advantage of our formulation is that when we compute the characteristics, known as geodesics, of the Eikonal equation via the extended Eikonal equation on the narrowband, the paths remain on the surface if the initial point lies on Γ . Since the solution to (37)–(38) is constant along the normals of Γ , the gradient always belongs to the tangent spaces of Γ or its parallel surfaces, Γ_η . We have that $B(\mathbf{x}, \mu)\mathbf{a} = \mathbf{a}$ for $\mathbf{x} \in \Gamma$ and $\mathbf{a} \in A_{\mathbf{x}} = T_{\mathbf{x}}\Gamma \cap S^{n-1}$ since $\sigma_1 = \sigma_2 = 1$. Thus, if the initial point is $\mathbf{z} \in \Gamma$, the geodesic on the surface can be obtained by solving the dynamical system

$$\frac{d\mathbf{y}}{dt}(t) = \frac{-\nabla v(\mathbf{y}(t))}{\|\nabla v(\mathbf{y}(t))\|}, \quad t > 0 \quad (41)$$

$$\mathbf{y}(0) = \mathbf{z}, \quad \mathbf{z} \in \Gamma. \quad (42)$$

In Fig. 6, we consider the case when we have two source points and display the solution and some geodesics. Recall that in Theorem 2.1 we showed that paths with equivalent initial points remain equivalent for all time. We show a visualization of this property in Fig. 7. In the next section we will compare geodesics of the HJB equation given different anisotropic speed functions.

³ The point cloud for the torus was generated using the standard parametrization of a torus.

⁴ The point cloud for the Stanford bunny is generated from a refinement of the triangulated Stanford bunny from <https://casual-effects.com/data/> [26].

⁵ The point clouds for the surfaces in Fig. 5 were generated from the triangulations downloaded at <https://www.myminifactory.com/scantheworld/>.

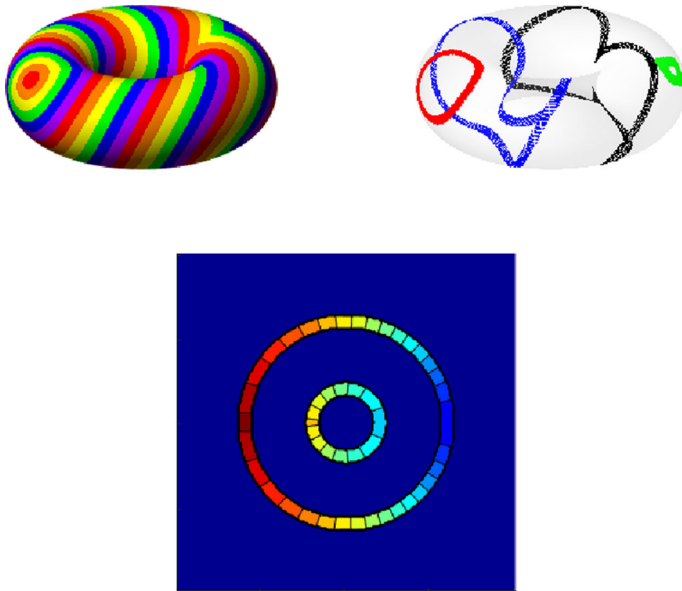


Fig. 3 Top left: Distance function on a torus. Top right: Points from the corresponding point cloud whose distance from the source point lies in the interval $(0.09, 1.1)$, $(0.29, 3.1)$, $(0.49, 5.1)$, or $(0.69, 7.1)$. Bottom: Solution slice at $z=5$ with contours showing that the solution is indeed constant along normal.

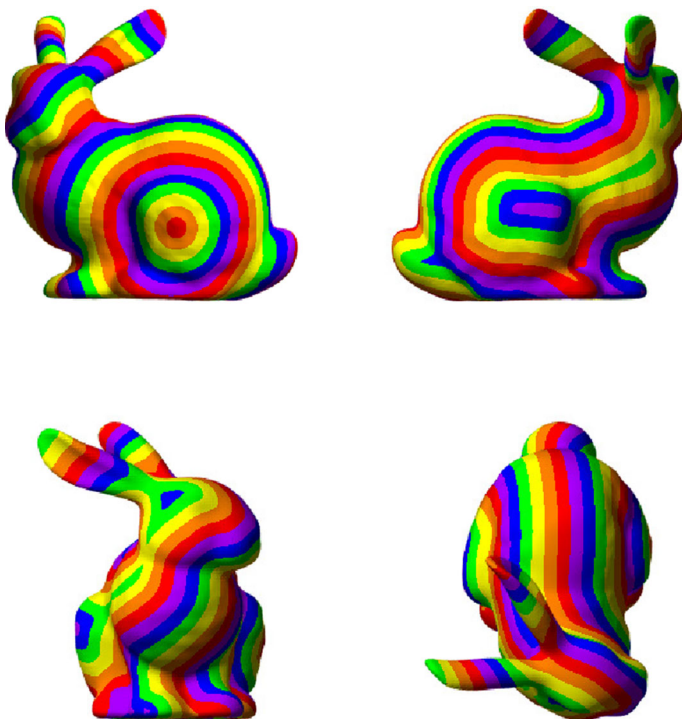


Fig. 4 Distance function on the Stanford bunny with four viewpoints.

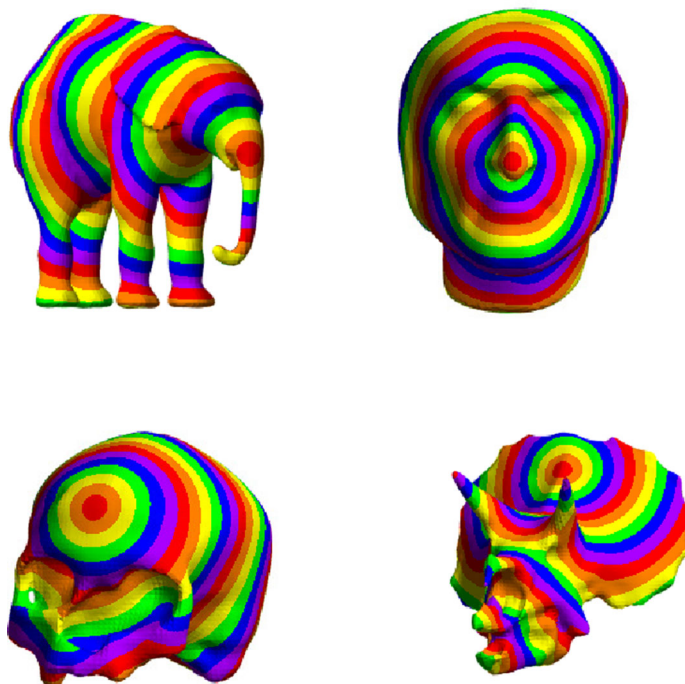


Fig. 5 Distance function on a elephant, mask, human skull, and dinosaur skull.

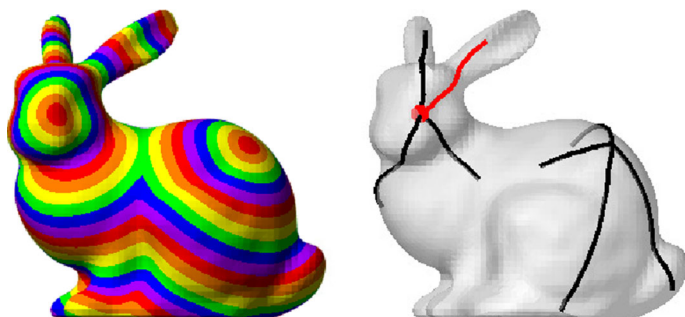
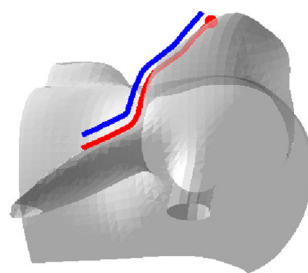


Fig. 6 Left: The distance function for two source points. Right: Eight geodesics computed on the Stanford bunny.

Fig. 7 Verification of Theorem 2.1, i.e., paths with equivalent starting points stay equivalent for all time. We plot the red geodesic from Fig. 6, and the initial point of the blue parallel path is offset by 0.015 along the normal of the bunny at that point.



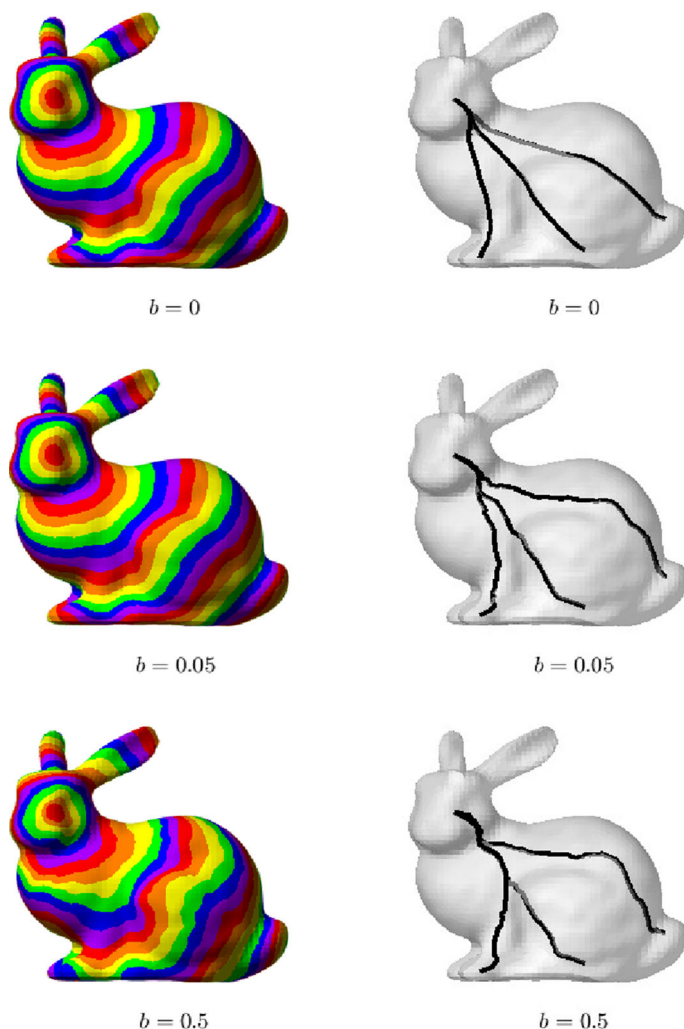


Fig. 8 Anisotropic example for the curvature based speed function showing the contours and geodesics for values of $b = 0, 0.05, 0.5$.

4.7 Example 3

Finally, we test implement the framework on an anisotropic speed function. We solve

$$\min_{\mathbf{a} \in A_z} \left\{ \nabla v(\mathbf{z}) \cdot \bar{f}(\mathbf{z}, \mathbf{a}) B(\mathbf{z}, \mu) \mathbf{a} + 1 \right\} = 0, \quad \mathbf{z} \in T_\epsilon \setminus \bar{T}, \quad (43)$$

$$v(\mathbf{z}) = \bar{g}(\mathbf{z}), \quad \mathbf{z} \in \bar{T}, \quad (44)$$

where $f : \Gamma \times A \rightarrow \mathbb{R}$ is a curvature based speed function on the surface. We use the speed function proposed in [38]. The speeds are fast on low curvature areas of the surface and slow

on high curvature areas. This means that the corresponding “shortest” paths traverse areas of the lowest curvatures. These paths are typically longer than the geodesics of the surface.

The normal curvature of Γ at $\mathbf{x}$ in the direction $\mathbf{a}$ is given by

$$\kappa_{\mathbf{a}}(\mathbf{x}) = \mathbf{a}^T \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix} \mathbf{a}$$

where κ_1 and κ_2 are the principal curvatures of Γ at $\mathbf{x}$. Then the curvature-minimizing speed function is given by

$$f(\mathbf{x}, \mathbf{a}) = \exp(-b\|\kappa_{\mathbf{a}}(\mathbf{x})\|),$$

where b is a positive constant. When $b = 0$, (43) reduces Eikonal equation on the surface. A larger value of b corresponds to a greater difference in speeds between areas of low and high curvature. We display the solutions for varying b values to a source point on the bunny in Fig. 8.

In this example, the set $V(\mathbf{x}) = \{f(\mathbf{x}, \mathbf{a})\mathbf{a} \mid \mathbf{a} \in A_{\mathbf{z}}\}$ is not necessarily convex for all $\mathbf{x}$ in Γ . Therefore, an optimal control may not exist. However, we can still extract suboptimal paths, called anisotropic geodesics, whose total cost is arbitrarily close to the value function at the starting point. Just as in the isotropic case, the anisotropic geodesics computed from the extended HJB equation on the narrow band will lie on the surface since $B(\mathbf{x}, \mathbf{a})\mathbf{a} = \mathbf{a}$ for $\mathbf{x} \in \Gamma$. The paths can easily be extracted because we compute the minimizing control at each grid node when solving (43)–(44). Once we have the optimal (or suboptimal) control values, $\mathbf{a}^*(\cdot)$, we then solve dynamical system

$$\begin{aligned} \frac{d\mathbf{y}}{dt}(t) &= f(\mathbf{y}(t), \mathbf{a}^*(t))\mathbf{a}^*(t), \quad t > 0 \\ \mathbf{y}(0) &= \mathbf{z}, \quad \mathbf{z} \in \Gamma. \end{aligned}$$

We plot three anisotropic geodesics for each b value in Fig. 8. We can see as b increases the Euclidean distances of the paths are longer, and the paths start to seek out the narrow valleys of the bunny.

5 Summary and Conclusion

In this paper, we presented a new formulation to compute solutions of a class of HJB equations on smooth hypersurfaces. We extend the HJB equation’s associated optimal control problem from the surface to an equivalent problem defined in a sufficiently “thin” narrow band around the surface, in the embedding Euclidean space. The extension was done so that the resulting value function is the constant normal extension of the value function defined in the optimal control problem on the surface. We presented the formulations for the general anisotropic equation and showed that the viscosity solution of the HJB equation on the narrow band is the constant normal extension of the viscosity solution on the surface, independently of the optimal control problems. We also presented the isotropic case and showed there is no need to restrict the control space in order to have an equivalent formulation. The proposed approach is independent of surface representation and can be used to compute and define optimal control problems on uniformly distributed point clouds sampled from some smooth surface. It is also clear that our proposed extension approach can be applied in to time dependent equations arising from the finite horizon control problems. Together with [9], our extension approach provides a good framework for solving to mean field games to high order accuracy on complicated and non-parametrized surfaces.

With this new framework, we are able to use Cartesian grids and the existing methods for computing solutions to HJB equations in Euclidean space on a very thin narrow band to solve HJB equations on surfaces coupled with a simple boundary closure procedure. Our numerical examples verify that the boundary closure procedure does not influence the overall order of the method. We also show that our formulation allows one to easily solve the surface HJB equations to high order accuracy.

Acknowledgements The authors are supported partially by National Science Foundation Grant DMS-1720171. Tsai also thanks National Center for Theoretical Study, Taipei for hosting his visits, in which some of the ideas presented in this paper originated.

References

1. Alton, K., Mitchell, I.M.: Fast marching methods for stationary Hamilton-Jacobi equations with axis-aligned anisotropy. *SIAM J. Numer. Anal.*, 47(1):363–385, (2008/09)
2. Bardi, M., Capuzzo-Dolcetta, I.: *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Systems & Control: Foundations & Applications. Birkhäuser Boston, Inc., Boston, MA. With appendices by Maurizio Falcone and Pierpaolo Soravia (1997)
3. Bardi, M., Falcone, M.: An approximation scheme for the minimum time function. *SIAM J. Control Optim.* **28**(4), 950–965 (1990)
4. Bartesaghi, A., Sapiro, G.: A system for the generation of curves on 3d brain images. *Hum. Brain Mapp.* **14**(1), 1–15 (2001)
5. Barth, T.J., Sethian, J.A.: Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains. *J. Comput. Phys.* **145**(1), 1–40 (1998)
6. Carlini, E., Falcone, M., Ferretti, R.: A time-adaptive semi-Lagrangian approximation to mean curvature motion. In *Numerical mathematics and advanced applications*, pages 732–739. Springer, Berlin, (2006)
7. Carlini, E., Falcone, M., Forcadell, N., Monneau, R.: Convergence of a generalized fast-marching method for an eikonal equation with a velocity-changing sign. *SIAM J. Numer. Anal.* **46**(6), 2920–2952 (2008)
8. Cheng, L.-T., Burchard, P., Merriman, B., Osher, S.: Motion of curves constrained on surfaces using a level-set approach. *J. Comput. Phys.* **175**(2), 604–644 (2002)
9. Chu, J., Tsai, R.: Volumetric variational principles for a class of partial differential equations defined on surfaces and curves. *Res. Math. Sci.*, 5(2):Paper No. 19, 38, (2018)
10. Crandall, M.G., Lions, P.-L.: Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.* **277**(1), 1–42 (1983)
11. Deckelnick, K., Elliott, C.M., Miura, T.-H., Styles, V.: Hamilton-Jacobi equations on an evolving surface. *Math. Comput.* **1**, 01 (2019)
12. Falcone, M.: A numerical approach to the infinite horizon problem of deterministic control theory. *Appl. Math. Optim.* **15**(1), 1–13 (1987)
13. Falcone, M.: The minimum time problem and its applications to front propagation. In *Motion by mean curvature and related topics (Trento, 1992)*, pp. 70–88. de Gruyter, Berlin (1994)
14. Falcone, M., Ferretti, R.: Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations. *Numer. Math.* **67**(3), 315–344 (1994)
15. Chu, J., Tsai, R.: Hamilton-Jacobi equations on a manifold and applications to grid generation or refinement. *SIAM J. Sci. Comput.* **23**(6), 2055–2073 (2002)
16. Kao, C.Y., Osher, S., Qian, J.: Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations. *J. Comput. Phys.* **196**(1), 367–391 (2004)
17. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci. USA* **95**(15), 8431–8435 (1998)
18. Kublik, C., Tsai, R.: Integration over curves and surfaces defined by the closest point mapping. *Res. Math. Sci.*, 3:Paper No. 3, 17, (2016)
19. Kumar, A., Vladimirovsky, A.: An efficient method for multiobjective optimal control and optimal control subject to integral constraints. *J. Comput. Math.* **28**(4), 517–551 (2010)
20. Lai, R., Liang, J., Zhao, H.: A local mesh method for solving PDEs on point clouds. *Inverse Probl. Imaging* **7**(3), 737–755 (2013)
21. Lenglet, C., Prados, E., Pons, J.-P., Deriche, R., Faugeras, O.: Brain connectivity mapping using Riemannian geometry, control theory, and PDEs. *SIAM J. Imaging Sci.* **2**(2), 285–322 (2009)

22. Li, F., Shu, C.-W., Zhang, Y.-T., Zhao, H.: A second order discontinuous Galerkin fast sweeping method for Eikonal equations. *J. Comput. Phys.* **227**(17), 8191–8208 (2008)
23. Liang, J., Zhao, H.: Solving partial differential equations on point clouds. *SIAM J. Sci. Comput.* **35**(3), A1461–A1486 (2013)
24. Macdonald, C.B., Ruuth, S.J.: The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM J. Sci. Comput.*, 31(6):4330–4350, (2009/10)
25. Mantegazza, C., Mennucci, A.C.: Hamilton-Jacobi equations and distance functions on Riemannian manifolds. *Appl. Math. Optim.* **47**(1), 1–25 (2003)
26. McGuire, M.: Computer graphics archive, July 2017. <https://casual-effects.com/data>
27. Mémoli, F., Sapiro, G.: Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *J. Comput. Phys.* **173**(2), 730–764 (2001)
28. Mémoli, F., Sapiro, G.: Distance functions and geodesics on submanifolds of $\mathbb{R}^d$ and point clouds. *SIAM J. Appl. Math.* **65**(4), 1227–1260 (2005)
29. Mirebeau, J.-M.: Anisotropic fast-marching on Cartesian grids using lattice basis reduction. *SIAM J. Numer. Anal.* **52**(4), 1573–1599 (2014)
30. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **79**(1), 12–49 (1988)
31. Rouy, E., Tourin, A.: A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.* **29**(3), 867–884 (1992)
32. Ruuth, S.J., Merriman, B.: A simple embedding method for solving partial differential equations on surfaces. *J. Comput. Phys.* **227**(3), 1943–1961 (2008)
33. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci. USA* **93**(4), 1591–1595 (1996)
34. Sethian, J.A., Vladimirsky, A.: Ordered upwind methods for static Hamilton-Jacobi equations: theory and algorithms. *SIAM J. Numer. Anal.* **41**(1), 325–363 (2003)
35. Spira, A., Kimmel, R.: An efficient solution to the eikonal equation on parametric manifolds. *Interfaces Free Bound.* **6**(3), 315–327 (2004)
36. Tsai, Y.-H.R., Cheng, L.-T., Osher, S., Zhao, H.-K.: Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Anal.* **41**(2), 673–694 (2003)
37. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Trans. Automat. Control* **40**(9), 1528–1538 (1995)
38. Yoo, S.W., Seong, J., Sung, M., Shin, S.Y., Cohen, E.: A triangulation-invariant method for anisotropic geodesic map computation on surface meshes. *IEEE Trans. Vis. Comput. Graph.* **18**(10), 1664–1677 (2012)
39. Zhang, Y.-T., Zhao, H.-K., Qian, J.: High order fast sweeping methods for static Hamilton-Jacobi equations. *J. Sci. Comput.* **29**(1), 25–56 (2006)
40. Zhao, H.: A fast sweeping method for eikonal equations. *Math. Comp.* **74**(250), 603–627 (2005)