SIAM J. SCI. COMPUT.
Vol. 39, No. 4, pp. A1616–A1646

A LEVEL SET BASED VARIATIONAL PRINCIPAL FLOW METHOD FOR NONPARAMETRIC DIMENSION REDUCTION ON RIEMANNIAN MANIFOLDS*

HAO LIU†, ZHIGANG YAO‡, SHINGYU LEUNG†, AND TONY F. CHAN§

Abstract. We propose a variational formulation for dimension reduction on Riemannian manifolds. The algorithm is developed based on the level set method together with a recently developed principal flow algorithm. The original principal flow algorithm is a Lagrangian technique which extends the principal component analysis (PCA) to dimension reduction on Riemannian manifolds. We propose to incorporate the level set method to obtain a fully implicit formulation so that the overall algorithm can naturally handle various topological changes in the curve evolution. The variational formulation consists of two terms which try to balance the contributions from both the dataset itself and the principal direction by the PCA. We will demonstrate that the method is insensitive to the initial guess and is robust enough for noisy data.

Key words. partial differential equations, implicit surfaces, level set method, dimensional reduction, PCA

AMS subject classifications. 35, 49

DOI. 10.1137/16M107236X

1. Introduction.

Principal component analysis (PCA) is a linear technique for reducing the dimension of a set of high dimensional data in the standard Euclidean space. The idea is to determine the linear projection to the data that maximizes the variance in the projected space. In many applications, however, we have data constrained on a low dimensional manifold embedded in some high dimensional space. One simple example is when the collected data are landmarks identifying the shape of certain objects. To better understand the shape deformation, one needs to map the data to the intrinsic manifold space rather than directly work with the measurements in the vector space.

In the past decade or so, there has been a vast variety of nonlinear approaches developed to better handle nonlinear complex data by primarily extending the standard PCA to the manifold scenario. Strictly speaking, the majority of these approaches have been focused on extending the PCA to the manifold cases in which the embedding is assumed to be known. Given such a known embedding or manifold, the principal geodesic analysis (PGA) [12, 17] defines the first principal component on the manifold to be the geodesic passing through the mean of the given data and with the minimum average distance to the sample points. In this sense, the PGA essentially replaces the line of the PCA in the Euclidean space with a geodesic on manifolds.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section April 26, 2016; accepted for publication (in revised form) June 13, 2017; published electronically August 24, 2017.

http://www.siam.org/journals/sisc/39-4/M107236.html

Funding: The second author's work was supported by MOE Tier 1 (R-155-000-154-133) and MOE Tier 2 (R-155-000-184-112). The third author's work was supported in part by the Hong Kong RGC grants 16303114 and 16309316.

†Department of Mathematics, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (hliuas@ust.hk, masyleung@ust.hk).

‡Department of Statistics and Applied Probability, National University of Singapore, 21 Lower Kent Ridge Road, Singapore 117546 (zhigang.yao@nus.edu.sg).

§Office of the President, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (tonyfchan@ust.hk).

Another method is the so-called tangent space PCA [14, 11], which first projects all the sample data onto the tangent space at the mean and then interprets the first principal component as the direction of maximal variability on the tangent space. Finally, the method maps this component back to the manifold. Not restricted to a single geodesic or curve, the method introduced in [12] decomposes the data variation by finding a sequence of nested linear submanifolds locally around the mean. A special case of using a collection of nested spheres can be found in [19]. Principal curves/surfaces, introduced in [15] as a dimensionality reduction tool in Euclidean space, are self-consistent smooth curves passing through the middle of the data. By extending [15], the authors in [16] have recently showed that the principal geodesic is a classic principal curve on Riemannian manifolds. Some other approaches more or less aim to directly fit a curve to the given sampling data on the manifold [23, 22, 41]. We refer interested readers to [30, 24, 13, 44] for a more complete discussion and a review on the application to dimensionality reduction.

In a recent work, a principal flow [36] approach has been proposed to determine a curve on the manifold passing through the mean of the data such that the tangent vector along the curve fits locally the first eigenvector of a local tangent space PCA. An extension to higher dimensions has been recently proposed in [46] by the concept of principal submanifold. This paper aims to incorporate the level set method [35, 38, 34] with the principal flow to determine a codimension-one surface Γ on the manifold that *best fits* the dataset. Instead of dealing with an explicit parametrization of a curve or a surface, the level set method was originally developed to simulate evolution of the interface by solving equations on a fixed Cartesian mesh. Because of the flexibility in modeling the topological change in the interface evolution, it has become a very popular tool in fields including multiphase flow simulation, shape optimization, computational geometry, computer graphics, etc. [38, 34]. The resulting method proposed in this work will, therefore, be a fully implicit and Eulerian approach which does not require any explicit parametrization of the underlying Riemannian manifold nor any assumption on the structure of the low dimensional representation of the given data.

Mathematically, we consider a complete Riemannian manifold $\mathcal{M}$ of dimension r, embedded in $\mathbb{R}^m$ such that $\mathcal{M}$ is implicitly represented by $\{\mathbf{x} \in \mathbb{R}^m : \psi(\mathbf{x}) = 0\}$ for a given function $\psi : \mathbb{R}^m \to \mathbb{R}^{m-r}$. Let $\mathcal{S} = \{\mathbf{x}_i \in \mathcal{M}, 1 \le i \le N\}$ be a configuration of N points on $\mathcal{M}$ such that there exists a connected open set $\mathcal{B} \subset \mathcal{M}$ covering $\mathcal{S}$. In this paper, even though we concentrate only on the case where $r = 2$ and $m = 3$, i.e., point data sampling a curve on a two dimensional manifold in the three dimensional space, the approach can be generalized to higher dimensional spaces in a rather straightforward way.

When $\mathcal{M}$ is simply the standard $\mathbb{R}^m$, the dimensional reduction problem becomes a problem of shape reconstruction if one assumes that the data points sample a codimension-one manifold. This is an important question in fields such as computer graphics or medical imaging [50, 49, 34]. The most widely used approach is to determine a triangulated representation of the surface based on techniques like the Voronoi diagrams or the Delaunay triangulation [1, 10]. However, when there is noise in the given data or when the topology of the underlying manifold is not known a priori, such an approach might not give a robust reconstruction. Another possibility is a variational approach developed in [50, 49] which incorporates the level set method [35, 38, 34]. Such an approach can give an implicit representation of the codimension-one surface by an isosurface of a level set function. The method requires only an unsigned distance function to the data points. It has been shown to be robust

to the noise in the given data and can also capture various topologies in the underlying surface. When the distance function in the method is replaced by the so-called edge detector, the method relates closely to the snake model and the geodesic active contour in the field of image segmentation [21, 5]. Based on this relationship and some recent convexification strategies [25, 6, 3], a convex functional for point cloud reconstruction has recently been proposed in [29] which can automatically lead to the global minimizer of the corresponding variational functional.

All these methods, however, have been developed solely for reconstructing data in the standard Euclidean space without specifying any constraint in the data. In this paper, we are going to follow a similar strategy but are proposing a variational approach for reconstructing data with an extra constraint which states that the given data lies on a submanifold in a Riemannian space. When $r = 2$ and $m = 3$, our method indeed shares some similarities with the method introduced in [2, 4, 9] for modeling evolution of curves in the three dimensional space. But we in this work focus mainly on the construction of the velocity field on the Riemannian manifold which drives the curve to the data.

This paper is organized as follows. Section 2 summarizes the necessary background of the propose approach. We will briefly discuss both the variational method for point cloud reconstruction based on the level set method and the principal flow algorithm for dimensional reduction. We then give our proposed algorithm for nonparametric dimension reduction in section 3. Some implementation details will be stated in section 4. Finally, we will show some numerical solutions in section 6 to demonstrate the effectiveness of the proposed method.

2. Background. Our proposed approach is developed based on two recent ideas. The first one is a variational approach developed in [50] which aims to reconstruct a two dimensional surface from point cloud data in $\mathbb{R}^3$ using the level set method [35]. The other one is the principal flow proposed in [36]. In this section, we summarize and review these two algorithms for data processing.

2.1. A variational method for Euclidean point cloud reconstruction using the level set method. Given an unconnected point cloud in an Euclidean space, the aim here is to reconstruct the underlying codimensional-one surface. The problem is very ill-conditioned in the sense that the solution might usually be nonunique. Some reconstruction algorithms might be too sensitive to the noise in the given data. To develop a robust algorithm for data in the Euclidean space, [50] has proposed a variational method based on the level set formulation. The main idea is to determine a surface Γ which minimizes the energy

$$E(\Gamma) = \left[\int_{\Gamma} d^p(\mathbf{x})\, ds \right]^{\frac{1}{p}},$$

where $d(\mathbf{x})$ is the distance function from $\mathbf{x}$ to the data set given by

$$d(\mathbf{x}) = \min_{\mathbf{x}_i \in \mathcal{S}} \|\mathbf{x} - \mathbf{x}_i\|_2 .$$

A straightforward way for the minimization process is to parameterize the curve or the surface and then to optimize the discretized problem using the gradient descent. Without any a priori knowledge on the topology of the curve, however, it might be tricky to numerically implement such approach. The main challenge is that one will need to remesh the triangulation by reconnecting the sampling points of the surface if

the topology of the interface does change in the evolution. To handle such change in the topology, [50] has reformulated the problem using the level set formulation. Let $\phi(\mathbf{x})$ be a level set function defined in $\mathbb{R}^m$ such that Γ is represented by zero level set of ϕ, i.e., $\Gamma = \phi^{-1}(0)$ and

$$\begin{array}{ll} \phi(\mathbf{x}) > 0 & \text{outside } \Gamma\,, \\ \phi(\mathbf{x}) < 0 & \text{inside } \Gamma\,. \end{array}$$

Now, the energy can then be expressed in term of the level set function ϕ,

$$E(\phi) = \left[\int_{\mathbb{R}^m} d^p(\mathbf{x})\delta(\phi)\, d\mathbf{x}\right]^{\frac{1}{p}},$$

where $\delta(x)$ is the Dirac delta function. Finally the corresponding Euler–Lagrange equation from the optimality condition is given by

$$\frac{\partial \phi}{\partial t} = \frac{1}{p}\,\delta(\phi)\,\left[\int d^p(\mathbf{y})\delta(\phi)\|\nabla\phi\| d\mathbf{y}\right]^{\frac{1}{p}-1} \nabla\cdot\left[d^p(\mathbf{x})\frac{\nabla\phi}{\|\nabla\phi\|}\right].$$

2.2. Principal flow. The principal flow extends the notion of the PCA to Riemannian manifolds [36]. On one hand, the flow on the manifold has the advantage of being principal in the sense that it follows the main direction of the data points locally. On the other hand, the method is able to accommodate the curve fitting characteristic on the manifold. Given a set of points $\mathcal{S} = \{\mathbf{x}_i \in \mathcal{M}, 1 \le i \le N\}$, a principal flow is parameterized as a one dimensional curve, Γ, over a class of flows parameterized as

$$\begin{aligned} \Gamma(\mathbf{x}, v) = \Big\{ &\eta : [0, r] \to \mathcal{M}, \eta \in C^2(\mathcal{M}), \eta(s) \neq \eta(s') \text{ for } s \neq s', \\ &\eta(0) = x, \dot{\eta}(0) = v, l(\eta[0, t]) = t\ \forall 0 \le t \le r \le 1 \Big\}, \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a chosen starting point (e.g., the Fréchet mean $\bar{\mathbf{x}}$) and v is the initial tangent unit at $\mathbf{x}$. This can be seen as a type of flows passing through the center of the data cloud, and at any point on the flow it points to the main direction of data variation.

Technically, the principal flow incorporates two ingredients: a local covariance matrix and a vector field. In a neighborhood of the point $\mathbf{x}$ at the scale h, a local tangent covariance matrix is defined as

$$\Sigma_h(\mathbf{x}) = \frac{1}{\sum_i \kappa_h(\mathbf{x}_i, \mathbf{x})} \sum_i \mathbf{log}_{\mathbf{x}}(\mathbf{x}_i) \otimes \mathbf{log}_{\mathbf{x}}(\mathbf{x}_i)\kappa_h(\mathbf{x}_i, \mathbf{x}),$$

where $\mathbf{y} \otimes \mathbf{y} := \mathbf{y}\mathbf{y}^T$, $\kappa_h(\mathbf{x}_i, \mathbf{x}) = K(h^{-1}\,\|\mathbf{log}_{\mathbf{x}}\mathbf{x}_i - \mathbf{x}\|)$ with a smooth nonincreasing univariate kernel K on $[0, \infty)$.

PROPOSITION 2.1 (differentiability of the vector field). *The eigenvector $e(\mathbf{x}_0)$ (or eigenvalue $\lambda(\mathbf{x}_0)$) of the local covariance matrix $\Sigma_h(\mathbf{x}_0)$ can be extended to a vector field defined on the open neighborhood $\mathbf{x}_0(h)$ of $\mathbf{x}_0$ such that for any $\mathbf{x} \in \mathbf{x}_0(h)$,*

- *$W : \mathbf{x}_0(h) \to \mathbb{R}^m$ is a differentiable function,*
- *$\Sigma_h(\mathbf{x})W(\mathbf{x}) = \lambda(\mathbf{x})W(\mathbf{x})$.*

The recent work [36] determines the principal flow over the candidate flow set (1) by solving the following two variational problems:

$$\eta_1 = \arg\sup_{\eta\in\Gamma(\bar{\mathbf{x}},v_1)} \int_0^{l(\eta)} \langle \dot{\eta}(t), W(\eta(t))\rangle \, dt\,,$$
$$\eta_2 = \arg\inf_{\eta\in\Gamma(\bar{\mathbf{x}},v_2)} \int_0^{l(\eta)} \langle \dot{\eta}(t), W(\eta(t))\rangle \, dt$$

where η_1 and η_2 are the two opposite curves corresponding to the solution of the variational problem, respectively; $v_1 = e_1(\bar{\mathbf{x}})$, $v_2 = -v_1$, and $l(\eta)$ is the length of the curve η. W defined in the Proposition 2.1 is the extension of the eigenvector $e_1(\bar{\mathbf{x}})$ to the neighborhood $\bar{\mathbf{x}}(h)$ of $\bar{\mathbf{x}}$. To obtain either η_1 or η_2, [36] has showed that it is necessary to find the critical point of the Lagrangian variational problem of

$$\mathcal{L}_1(W,\eta) = \int_0^{l(\eta)} \langle \dot{\eta}(t), W(\eta(t))\rangle \, dt \tag{2}$$

subject to boundary conditions under certain reparameterization of the candidate set (1), which does not change the integration in (2). (See more details in [36].) The unique solution of (2) under mild conditions is then shown to reduce to an ODE problem via the Euler–Lagrange method, where a numerical algorithm has also been provided.

3. The proposed formulation. In this work, we develop a variational approach for point cloud reconstruction on manifold $\mathcal{M}$ by extending the variational level set method in [50] and also incorporating the principal flow idea in [36] as a data fidelity. There are two main ingredients in the formulation. One is to extend the definition of the distance function which computes the *unsigned* distance from a grid point to the data set. Since the data points are defined on a manifold, one has to first replace the Euclidean distance in the original definition by the geodesic distance on $\mathcal{M}$. This can be easily done using a recent embedding approach developed in [45]. In the original variation level set reconstruction of point cloud, one uses only the data point location for defining the mismatch in the functional. Because of the limitation in the information provided, the gradient descent approach can be easily trapped in a local minimizer. To drive the reconstruction out from a local minimizer of the functional, we propose to smooth the information from the data point location. The second main ingredient of the proposed approach is to incorporate the information from the PCA to drive our result to better fit the data points.

Mathematically, we propose the following energy given by

$$E(\Gamma) = \left[\int_\Gamma d^p(\mathbf{x}(s))\, ds\right]^{\frac{1}{p}} + \lambda \left[\int_\Gamma |\mathbf{n}(\mathbf{x}(s))\cdot \mathbf{p}(\mathbf{x}(s))|^2 \, ds\right]^{\frac{1}{2}} \tag{3}$$

with the constraint $\Gamma \in \mathcal{M}$. In the energy, $p \geq 1$ is an integer, and $\mathbf{p}(\mathbf{x}(s))$ is the principal direction estimated from the data points in a small neighborhood of $\mathbf{x}(s)$ along the curve Γ parameterized by s. The vector $\mathbf{n}(\mathbf{x}(s))$ along Γ is the normal direction of the reconstructed curve on $\mathcal{M}$, and $\lambda \geq 0$ controls the weight of this directional fidelity term to the location of the interface itself.

The first term is similar to the one proposed in [50], which also tries to fit a surface to the given data points. If the curve Γ is close to the data, the first line integral will be small. However, the current model is more restrictive than the original one in

[50] since we now have an extra nonlinear constraint imposed on Γ. In the original formulation [50], the function $d(\mathbf{x})$ is a data matching function which equals zero if $\mathbf{x} \in \mathcal{S}$ is one of a given data set and is straightly positive for $\mathbf{x} \notin \mathcal{S}$. A simple candidate is to set $d(\mathbf{x}) = g(\mathbf{x})$, where $g(\mathbf{x}) = dist(\mathbf{x}, \mathcal{S}) : \mathcal{M} \to [0, \infty]$ is the geodesic distance function on $\mathcal{M}$ to the dataset $\mathcal{S}$. Mathematically, this function can be obtained by solving the surface eikonal equation

$$\|\nabla_{\mathcal{M}} g\| = 1 \tag{4}$$

together with the boundary condition $g(\mathbf{x}) = 0$ for $\mathbf{x} \in \mathcal{S}$, where $\nabla_{\mathcal{M}}$ is the surface gradient operator defined on the manifold $\mathcal{M}$. To have a more robust algorithm, we propose to relax the above condition and define $d(\mathbf{x})$ to be the smoothed version of $g(\mathbf{x})$ such that $d(\mathbf{x}) = G_\sigma \star g(\mathbf{x})$ for some Gaussian kernel G_σ with standard deviation σ. We will discuss a numerical algorithm to find the geodesic distance $g(\mathbf{x})$ and the justification of such a choice of $d(\mathbf{x})$ in the next section.

The second term is a fidelity term which tries to enforce the estimated principal direction $\mathbf{p}(\mathbf{x})$ to be perpendicular to the normal vector along the curve Γ at $\mathbf{x}$. In the case when the data points are living in the three dimensional space, i.e., $m = 3$, with the dimension of the manifold $\mathcal{M}$ being two, we can obtain a vector on the tangent plane at $\mathbf{x}$ (denoted by $T_{\mathbf{x}}\mathcal{M}$) perpendicular to $\mathbf{p}(\mathbf{x})$. If we denote this vector by $\mathbf{p}^\perp(\mathbf{x})$ such that $\mathbf{p}^\perp(\mathbf{x}) \in T_{\mathbf{x}}\mathcal{M}$ and $\mathbf{p}(\mathbf{x}) \cdot \mathbf{p}^\perp(\mathbf{x}) = 0$, the fidelity term can be rewritten as

$$\left[\int_\Gamma \left|1 - \mathbf{n}(s) \cdot \mathbf{p}^\perp(s)\right|^2 ds\right]^{\frac{1}{2}},$$

which is similar to some variational formulations for surface processing via normal map [42, 43]. For a general scenario where both m and the dimension of $\mathcal{M}$ are unknown, the normal direction of the reconstructed surface $\mathbf{p}^\perp$ might not be readily available. It is therefore more natural to directly incorporate our model with the vector $\mathbf{p}$ estimated immediately from the PCA, as introduced in this work.

In the level set formulation, we represent the manifold $\mathcal{M}$ implicitly by a level set function $\psi(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^m$ such that $\|\nabla\psi\| = 1$. Moreover, following [2, 4, 9], we introduce a second level set function $\phi(\mathbf{x})$ so that the curve Γ on $\mathcal{M} = \psi^{-1}(0)$ is implicitly represented by the intersection of these level set functions, i.e.,

$$\Gamma = \{\mathbf{x} \in \mathbb{R}^m : \psi(\mathbf{x}) = \phi(\mathbf{x}) = 0\}.$$

Therefore, the minimization problem (3) can now be rewritten as

$$\begin{aligned} &E(\phi; \lambda) \\ &\quad = \left[\int_{\mathbb{R}^m} d^p(\mathbf{x})\delta(\phi)\delta(\psi)\|\nabla\phi\|\, d\mathbf{x}\right]^{\frac{1}{p}} + \lambda\left[\int_{\mathbb{R}^m} |\mathbf{n}(\mathbf{x}) \cdot \mathbf{p}(\mathbf{x})|^2 \delta(\phi)\delta(\psi)\|\nabla\phi\|\, d\mathbf{x}\right]^{\frac{1}{2}} \\ &\quad = \left[\int_{\mathcal{M}_\epsilon} d^p(\mathbf{x})\delta(\phi)\delta(\psi)\|\nabla\phi\|\, d\mathbf{x}\right]^{\frac{1}{p}} + \lambda\left[\int_{\mathcal{M}_\epsilon} |\mathbf{n}(\mathbf{x}) \cdot \mathbf{p}(\mathbf{x})|^2 \delta(\phi)\delta(\psi)\|\nabla\phi\|\, d\mathbf{x}\right]^{\frac{1}{2}}, \end{aligned} \tag{5}$$

where the integral along Γ is converted into an integral in the full m-dimensional space $\mathbb{R}^m$ or an ϵ-neighborhood of the manifold, denoted by $\mathcal{M}_\epsilon$, using the Dirac delta functions $\delta(\psi(\mathbf{x}))$ and $\delta(\phi(\mathbf{x}))$. The normal vector $\mathbf{n}(s)$ along Γ is now converted into the normal vector $\mathbf{n}(\mathbf{x})$ along each level curve of $\phi(\mathbf{x})$ and can be easily computed using $\nabla\phi/\|\nabla\phi\|$.

To minimize (5), we first derive the variation of E with respect to the level set equation ϕ in a direction h. Formally, we obtain

$$\left(\frac{\delta E}{\delta \phi}, h\right) = -\gamma_1 I_1 - \lambda\gamma_2 \left(I_2 + I_3\right) ,$$

where

$$\begin{aligned}
I_1 &= \int_{\mathcal{M}_\epsilon} \delta(\phi)\nabla\cdot\left[d^p(\mathbf{x})\delta(\psi)\frac{\nabla\phi}{\|\nabla\phi\|}\right] h\, d\mathbf{x}\,,\\
I_2 &= \int_{\mathcal{M}_\epsilon} \left\{2\left(\mathbf{p}\cdot\frac{\nabla\phi}{\|\nabla\phi\|}\right)\nabla\cdot\left[\delta(\phi)\delta(\psi)\mathcal{P}\mathbf{p}\right]\right\} h\, d\mathbf{x}\,,\\
I_3 &= \int_{\mathcal{M}_\epsilon} \left\{\delta(\phi)\nabla\cdot\left[\delta(\psi)\left(\frac{\nabla\phi}{\|\nabla\phi\|}\otimes\frac{\nabla\phi}{\|\nabla\phi\|}\right)\mathbf{p}\right]\right\} h\, d\mathbf{x}\,,\\
\gamma_1 &= \frac{1}{p}\left[\int_{\mathcal{M}_\epsilon} d^p(\mathbf{y})\delta(\phi)\delta(\psi)\|\nabla\phi\| d\mathbf{y}\right]^{\frac{1}{p}-1} > 0\,,\\
\gamma_2 &= \frac{1}{2}\left[\int_{\mathcal{M}_\epsilon} |\mathbf{n}(\mathbf{y})\cdot\mathbf{p}(\mathbf{y})|^2\,\delta(\phi)\delta(\psi)\|\nabla\phi\|\, d\mathbf{y}\right]^{-\frac{1}{2}} > 0\,,
\end{aligned}$$

and $\mathcal{P} = (I - \mathbf{n}\otimes\mathbf{n})$ is the projection operator onto the direction orthogonal to $\mathbf{n}$. Now, based on the gradient descent, we introduce an artificial time t and determine the steady state solution to the following time-dependent equation with some initial guess,

$$\frac{\partial\phi}{\partial t} = \delta(\phi)\nabla\cdot\left\{\left[\gamma_1\, d^p(\mathbf{x}) + \lambda\gamma_2\left(\mathbf{n}\cdot\mathbf{p}\right)^2\right]\mathbf{n}\,\delta(\psi)\right\} + \lambda\gamma_2\nabla\cdot\left[2(\mathbf{n}\cdot\mathbf{p})\mathcal{P}\mathbf{p}\,\delta(\phi)\delta(\psi)\right] . \tag{6}$$

Both terms on the right-hand side have factors $\delta(\phi)$ and $\delta(\psi)$ which concentrate the computations only near the curve Γ on the manifold $\mathcal{M}$. Following [48], we can further replace $\delta(\phi)$ by $\|\nabla\phi\|$, and $\delta(\psi)$ by $\|\nabla\psi\| = 1$. If we reinitialize and orthogonalize the level set functions such that $\|\nabla\phi\| = 1$ and $\nabla\phi\cdot\nabla\psi = 0$, we finally obtain

$$\begin{aligned}
\frac{\partial\phi}{\partial t} &= \nabla\cdot\left[\gamma_1\, d^p(\mathbf{x})\,\mathbf{n} + \lambda\gamma_2(\mathbf{p}+\mathcal{P}\mathbf{p})(\mathbf{p}\cdot\mathbf{n})\right]\\
&= \nabla\cdot\left[\gamma_1\, d^p(\mathbf{x})\,\nabla\phi + \lambda\gamma_2(\mathbf{p}+\mathcal{P}\mathbf{p})(\mathbf{p}\cdot\nabla\phi)\right] .
\end{aligned} \tag{7}$$

We then solve this partial differential equation up to the steady state. The zero level set of the solution gives an implicit representation of the reconstruction on the given manifold $\mathcal{M}$.

To end this section, we give the following properties and observations concerning the functional which help us to understand the performance of the algorithm.

Observation 3.1. The two terms in the energy are of different units. The first term has the unit of length and has a scale depending on the manifold, while the second term is an integral of a scalar function which has no unit. To balance these two terms, one can pick $\lambda = O(L)$ for the length scale L for the manifold.

Observation 3.2. Although the vector $\mathbf{p}$ estimated from the PCA is unique only up to the sign, the factor $(\mathbf{p}+\mathcal{P}\mathbf{p})(\mathbf{p}\cdot\nabla\phi)$ remains unchanged if we replace $\mathbf{p}$ by $-\mathbf{p}$. This implies that the sign of $\mathbf{p}$ will not affect our result.

Observation 3.3. Since $\|\mathbf{p}\|^2 = (\mathbf{n}\cdot\mathbf{p})^2 + \|\mathcal{P}\mathbf{p}\|^2 = (\mathbf{n}\cdot\mathbf{p})^2 + \mathbf{p}\cdot\mathcal{P}\mathbf{p}$, we have

$$\begin{aligned}\|\mathbf{p}+\mathcal{P}\mathbf{p}\|^2 &= \|\mathbf{p}\|^2 + \|\mathcal{P}\mathbf{p}\|^2 + 2(\mathbf{p}\cdot\mathcal{P}\mathbf{p})\\ &= 1 + \left[1-(\mathbf{n}\cdot\mathbf{p})^2\right] + 2\left[1-(\mathbf{n}\cdot\mathbf{p})^2\right]\\ &= 4 - 3(\mathbf{n}\cdot\mathbf{p})^2 \geq 1\,.\end{aligned}$$

Therefore, the term $\mathbf{p}+\mathcal{P}\mathbf{p}$ can never be the zero vector. This implies that the second term in the divergence operator cannot be zero, except when the normal $\mathbf{n} = \nabla\phi$ is orthogonal to the directional vector $\mathbf{p}$ estimated from the PCA. In the evolution of ϕ, the component in the functional corresponding to the principal direction always contributes to the functional unless Γ is tangent to $\mathbf{p}$.

4. Implementation details. In this section, we explain the implementation details. As mentioned before, the function $d(x)$ in the functional depends on the geodesic distance function $g(\mathbf{x})$. We will first introduce a fast sweeping algorithm to numerically compute $g(\mathbf{x})$. Then we will discuss the choice of the function $d(\mathbf{x})$. To improve the computational complexity, we will discuss an approximation to the typical PCA which can be efficiently determined. To improve the stability and accuracy of the overall algorithm, we follow the standard regularization procedure in the level set community using the level set reinitialization and orthogonalization. The numerical implementations of these two algorithms will be given at the end of this section.

4.1. The geodesic function $g(\mathbf{x})$. We determine the geodesic function $g(\mathbf{x})$ by solving the surface eikonal equation (4). In this work, we follow our recent embedding approach developed in [45], which extends and improves the algorithm in [32]. Since the solution of the surface eikonal equation on an implicit surface is the intrinsic weighted distance function, we approximate such a distance function by the Euclidean weighted distance function defined in a tubular neighborhood of the implicit surface. In other words, we replace the surface eikonal equation (4) by the typical eikonal equation $\|\nabla g\| = 1$ defined in the small neighborhood of $\mathcal{M}$, i.e., $\mathcal{M}_\epsilon$, which can be easily solved by the fast sweeping method developed based on the Lax–Friedrichs Hamiltonian or the Godunov Hamiltonians [20, 47].

The fundamental idea of the fast sweeping method is to design an upwind, monotone, and consistent discretization for the nonlinear term $\|\nabla g\|$. Furthermore, if there is a simple local solver for the discretized system, one can then obtain an efficient numerical algorithm for solving the nonlinear partial differential equation. The term *sweeping* comes from the fact that all methods in this class can be easily incorporated with the symmetric Gauss–Seidel iterations which may lead to an $O(N)$ algorithm with N the total number of mesh points in the computational domain. In the two dimensional space ($m = 2$), for example, the local solver for the Godunov Hamiltonian can be explicitly constructed, and it leads to the iterative formula

$$g_{i,j} \leftarrow \begin{cases} \min(g^x_{\min}, g^y_{\min}) + \Delta x & \text{if } |g^x_{\min} - g^y_{\min}| > \Delta x,\\ \frac{1}{2}\left[g^x_{\min} + g^y_{\min} + \sqrt{2\Delta x^2 - (g^x_{\min} - g^y_{\min})^2}\right] & \text{otherwise.}\end{cases} \tag{8}$$

The corresponding local solver for the LxF Hamiltonian is given by

$$\begin{aligned} g_{i,j} \leftarrow\ & \frac{g_{i+1,j} + g_{i-1,j} + g_{i,j+1} + g_{i,j-1}}{4}\\ & + \frac{\Delta x}{2}\left[1 - \sqrt{\left(\frac{g_{i+1,j} - g_{i-1,j}}{2\Delta x}\right)^2 + \left(\frac{g_{i,j+1} - g_{i,j-1}}{2\Delta x}\right)^2}\right]. \end{aligned} \tag{9}$$

To correctly impose the boundary condition on the boundary $\partial\mathcal{M}_\epsilon$ taking care of the causality, however, we have proposed to introduce an extra layer of computational tube

$$\partial\mathcal{M}_{\epsilon'} = \{\mathbf{x} \in \mathbb{R}^m : \epsilon < \psi(\mathbf{x}) < \epsilon'\}$$

and impose an extension equation in this computational domain. In two dimensions, for example, $\mathbf{v} \cdot \nabla g = (p, q) \cdot (g_x, g_y) = 0$, we can use the simple upwind differencing to discretize the equation and derive the update formula [26, 7, 27, 28]. At each point (x_i, y_j), we define $p^\pm = \frac{1}{2}(p_{i,j} \pm |p_{i,j}|)$ and, therefore, obtain the update formula given by

$$g_{i,j} \leftarrow \frac{p^+ g_{i-1,j} - p^- g_{i+1,j} + q^+ g_{i,j-1} - q^- g_{i,j+1}}{|p_{i,j}| + |q_{i,j}|}. \tag{10}$$

We summarize the overall fast sweeping method in Algorithm 1. For more details, we refer interested readers to [45].

Algorithm 1 A fast sweeping method for $|\nabla_\Sigma g| = 1$ [45].

Data: The source location $\mathbf{x}_s$, the mesh size Δx, the level set representation of the surface ψ_i, the inner tube radius ϵ, and the outer tube radius ϵ'
Result: g_i in the computational tube
Initialization: Set $g_i = 0$ if $\mathbf{x}_i$ is at the point source, otherwise $g_i = \infty$
while *not converged* **do**
  for *each of the* 2^m *sweeping directions* **do**
    if $|\psi_i| \le \epsilon$ *and* $g_i \neq 0$ **then**
      update g_i using (8) for the Godunov Hamiltonian or (9) for the LxF Hamiltonian
    else if $\epsilon < |\psi_i| \le \epsilon'$ **then**
      update g_i using (10) with $\mathbf{v} = \mathrm{sgn}(\psi)\nabla\psi/\|\nabla\psi\|$
    end
  end
end

4.2. The function $d(\mathbf{x})$. The data mismatch function $d(\mathbf{x})$ is crucial in the algorithm. Numerically, it might not be a good idea to directly use $d(\mathbf{x}) = g(\mathbf{x})$ since such a choice will give $d(\mathbf{x}_i) = 0$ for any data $\mathbf{x}_i \in \mathcal{S}$ and the evolution of ϕ will easily get stuck at these local minimum when it touches any of these data points. The situation will be worse if the dataset $\mathcal{S}$ contains noise because the interface reconstructed will be significantly polluted by the perturbation. A simple example is shown in Figure 1(a). We consider $\mathcal{M} = \mathbb{R}^2$, and therefore the geodesic function $d(\mathbf{x})$ is now reduced back to the usual Euclidean distance function. We assume data are given at the four corners of a square with extra points located on the side. The reconstruction with the choice of $d(\mathbf{x}) = g(\mathbf{x})$ is plotted using the blue curve. The level set function ϕ is stuck because of those outmost data points.

Following the idea of edge based segmentation algorithms [21, 5], we regularize $g(\mathbf{x})$ by a simple low pass filter. This can be done easily by using the box filter or simply convolving the geodesic distance function with a Gaussian, denoted by

$$d(\mathbf{x}) = G_\sigma \star g(\mathbf{x}) \tag{11}$$

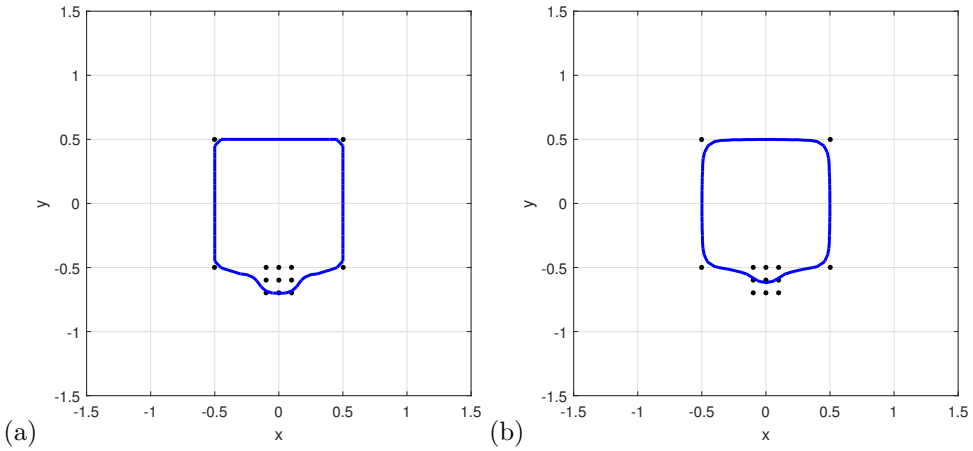


FIG. 1. (a) *Typical scenario when* $d(\mathbf{x}) = g(\mathbf{x})$ *with* $\lambda = 0$. (b) $d(\mathbf{x}) = G_\sigma \star g(\mathbf{x})$ *with* $\lambda = 0$. *The data points given in* $\mathcal{M} = \mathbb{R}^2$ *are plotted in black dots, while the reconstructed curve is plotted in blue.*

for some standard deviation σ. Since the geodesic distance function is defined in a small neighborhood of the manifold, i.e., $\mathcal{M}_\epsilon$, the box filter implies

$$d(\mathbf{x}) = \frac{\int_{\mathcal{B}_r(\mathbf{x})\cap\mathcal{M}_\epsilon} g(\mathbf{y})d\mathbf{y}}{\int_{\mathcal{B}_r(\mathbf{x})\cap\mathcal{M}_\epsilon} d\mathbf{y}},$$

where $\mathcal{B}_r(\mathbf{x})$ is a ball of radius r centered at the point $\mathbf{x} \in \mathcal{M}_\epsilon$ with $r = O(\epsilon)$. We plot in Figure 1(b) the reconstruction based on this smoothed geodesic distance function with the same dataset as in Figure 1(a). Even though the solution is now regularized with the corners significantly smoothened, such regularity would be important in developing a robust algorithm when dealing with noised data.

Having said that the smoothing is important for noisy data, the smoothing parameter σ cannot be arbitrarily large since introducing an excessive amount of regularization to the functional might lead to an oversmoothed and undesired solution. To see this, we consider the case where $g(\mathbf{x})$ is given by $|\|\mathbf{x}\| - r_0|$ representing infinitely many data points that are given on the circle of radius r_0 centered at the origin. We consider the function $G_\sigma \star g(\mathbf{x})$ for various magnitude of σ. The larger the value of σ, the larger the averaging window and therefore the solution $d(\mathbf{x})$ will be smoother. Figure 2 shows the function $d(r) = d(\|\mathbf{x}\|)$ as a function of r, the distance away from the origin. The solutions for various $\sigma > 0$ are plotted in red. As we can see, the minimum of $d(r)$ shifts toward the left as we increase σ. Although the overall energy (3) for a circular Γ of radius r^* and $\lambda = 0$ is given by $E(r^*) = [2\pi r^* d^p(r^*)]^{\frac{1}{p}}$, Figure 2 has clearly demonstrated that the minimizer to (3) might be significantly perturbed if the data mismatch function is oversmoothed.

Once we determine $d(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{M}_\epsilon$, we further extend it to those grid points adjacent to the computational tube by solving the extension equation $\mathbf{n}\cdot\nabla d = 0$ using the fast sweeping method as discussed in [26, 7, 27, 28, 45].

4.3. The estimation p(x) from the PCA. The standard way to determine the principal direction $\mathbf{p}$ consists of two steps. The first step is to project all data points onto a tangent plane at $\mathbf{x}$ on $\mathcal{M}$ using the log map. Then we can apply the typical PCA on the $\mathbb{R}^{m-1}$ plane. In our formulation, however, such an approach is not

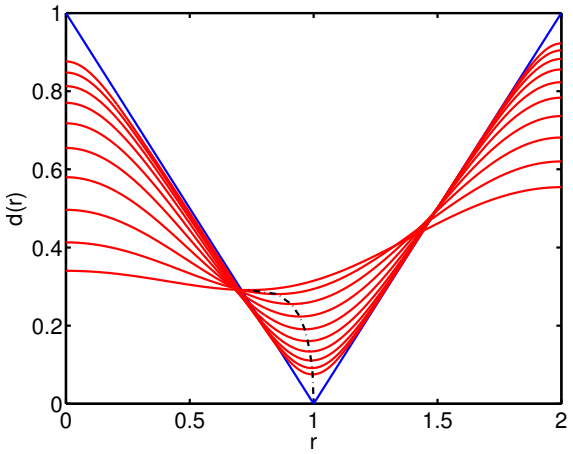


FIG. 2. *Regularization using* $d(\mathbf{x}) = G_\sigma \star g(\mathbf{x})$ *with* $g(\mathbf{x}) = |\|\mathbf{x}\| - r_0|$ *and* $r_0 = 1$. *The minimizers of* $d(r)$ *for different* σ *are plotted in the black dash-dotted line.*

directly applicable. The first concern is about the computational efficiency. Since we need $\mathbf{p}$ at every grid point in our computing tube $\mathcal{M}_\epsilon$, the computational complexity is therefore $O(N)$ when N is the number of data points in $\mathcal{S}$. Second, $\mathbf{x}$ is a grid point and is not necessarily on $\mathcal{M}$. In particular, the point $\mathbf{x}_i$ lives only on the $(m-1)$ dimensional manifold $\{\mathbf{x} \in \mathbb{R}^m : \psi(\mathbf{x}) = \psi(\mathbf{x}_i)\}$ but not necessarily in $\psi^{-1}(0)$.

Motivated by [36], we propose to compute the principal direction $\mathbf{p}(\mathbf{x})$ in the following way. At each grid point $\mathbf{x}$, we first collect all data points within a small neighborhood, denoted by $\mathcal{S}_\mathbf{x} = \{\mathbf{x}_i \in \mathcal{S} : \|\mathbf{x}-\mathbf{x}_i\|_2 < \delta\}$. Then, we simply define $\mathbf{p}(\mathbf{x})$ to be the normalized eigenvector corresponding to the largest magnitude eigenvalue of

$$\sum_{\mathbf{x}_i \in S_\mathbf{x}} (\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T . \tag{12}$$

Note that this approach performs the standard PCA in the Euclidean space by relaxing the nonlinear Riemannian constraint. It is *different* from the tangent PCA which involves a projection of all data points onto the tangent plane at $\mathbf{x}$ using the log map. For the ease of the discussion, we will denote the corresponding principal direction on the tangent plane by $\mathbf{p}_{\log}$. In practice, however, it might be numerically challenging to compute the log map on Riemannian manifolds. Instead, we simply replace the log map by the simple Euclidean orthogonal projection and denote the resulting principal direction by $\mathbf{p}_{\text{orth}}$. Indeed, we do not expect to have the same $\mathbf{p}$, $\mathbf{p}_{\log}$, and $\mathbf{p}_{\text{orth}}$. But we are going to show that under some mild conditions on the data, we have $\mathbf{p} = \mathbf{p}_{\text{orth}}$ which turns out to be a good approximation to $\mathbf{p}_{\log}$.

THEOREM 4.1. *Given* $\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathcal{M} \subset \mathbb{R}^3$ *for* $n > 0$ *and denoting* $\mathbf{y}_i = (\mathbf{x}_i - \mathbf{x})$, *let* $\mathbf{n}$ *be the unit normal of the tangent plane at* $\mathbf{x}$, *and* $\mathbf{t}_i$ *be the unit tangent component of* $\mathbf{y}_i$ *on the tangent plane. If* $\mathbf{y}_i \cdot \mathbf{t}_i > \sqrt{2}\ (\mathbf{y}_i \cdot \mathbf{n})$ *for all data points, we have* $\mathbf{p} = \mathbf{p}_{orth}$.

To show this theorem, we need the following two lemmas.

LEMMA 4.1. *The principal direction* $\mathbf{p}_{orth}$ *is the first or the second principal direction of the standard PCA in the Euclidean space.*

Proof. Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be the set of data points in a small neighborhood of $\mathbf{x}$ with $\mathbf{x}$ and $\mathbf{x}_i \in \mathcal{M} \subset \mathbb{R}^m$ for all $i = 1, \ldots, n$. Introducing $\mathbf{y}_i = \mathbf{x}_i - \mathbf{x}$ and

$$\mathbf{y}_i = (\mathbf{y}_i \cdot \mathbf{t}_i)\ \mathbf{t}_i + (\mathbf{y}_i \cdot \mathbf{n})\ \mathbf{n}\,,$$

where $\mathbf{t}_i$ is the unit tangent component of $\mathbf{y}_i$ on the tangent plane at $\mathbf{x}$ and $\mathbf{n}$ is the corresponding unit normal vector of the tangent plane, we compute the standard PCA of the Euclidean space by determining the principal eigenvector of

$$M_n = \sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i^T = \sum_{i=1}^{n} \left[(\mathbf{y}_i \cdot \mathbf{t}_i)^2\, \mathbf{t}_i \mathbf{t}_i^T + (\mathbf{y}_i \cdot \mathbf{n})^2\, \mathbf{n}\mathbf{n}^T \right] = E_n + F_n$$

with $E_n = \sum_{i=1}^{n} (\mathbf{y}_i \cdot \mathbf{t}_i)^2\, \mathbf{t}_i \mathbf{t}_i^T$ and

$$F_n = \sum_{i=1}^{n} (\mathbf{y}_i \cdot \mathbf{n})^2\, \mathbf{n}\mathbf{n}^T = \left[\sum_{i=1}^{n} (\mathbf{y}_i \cdot \mathbf{n})^2 \right] \mathbf{n}\mathbf{n}^T = \xi_n\, \mathbf{n}\mathbf{n}^T\,,$$

where $\xi_n = \sum_{i=1}^{n} (\mathbf{y}_i \cdot \mathbf{n})^2 \geq 0$. Note that $\mathbf{p}_{\text{orth}}$ is simply the principal eigenvector of E_n. The component F_n has only one nonzero eigenvalue given by ξ_n with the corresponding eigenvector $\mathbf{n}$.

Let $\mathbf{s}_j$ be the eigenvectors of E_n for $j = 1, \ldots, m$ with corresponding eigenvalues λ_j with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$, i.e., $E_n \mathbf{s}_j = \lambda_j \mathbf{s}_j$. Since $\mathbf{n}$ is orthogonal to all $\mathbf{t}_i$, it is clear that $\mathbf{s}_m = \mathbf{n}$ is an eigenvector of E_n with the corresponding eigenvalue $\lambda_m = 0$. This implies that

$$M_n \mathbf{n} = (E_n + F_n)\mathbf{n} = 0\mathbf{n} + F_n \mathbf{n} = \xi_n\, \mathbf{n}\,,$$

and so $\mathbf{n}$ is also an eigenvector of M_n with the corresponding eigenvalue given by ξ_n. For $\mathbf{s}_j \neq \mathbf{n}$, we have

$$M_n \mathbf{s}_j = E_n \mathbf{s}_i + F_n \mathbf{s}_j = E_n \mathbf{s}_i + 0\mathbf{s}_j = \lambda_j \mathbf{s}_j\,.$$

Therefore, the matrices M_n and E_n have the same set of eigen-pairs, except for the eigenvector $\mathbf{n}$, where the corresponding eigenvalue of E_n is 0 while that of M_n is ξ_n. Finally, if $\lambda_1 > \xi_n$, we have $\mathbf{p} = \mathbf{p}_{\text{orth}}$. Otherwise, $\mathbf{p}_{\text{orth}}$ is the second principal direction of the standard PCA in the Euclidean space. ∎

LEMMA 4.2. *Follow Lemma* 4.1 *and consider* $m = 3$. *If* $\mathbf{y}_i \cdot \mathbf{t}_i > \sqrt{2}\,(\mathbf{y}_i \cdot \mathbf{n})$ *for all data points* $\mathbf{x}_i$, *we have* $\lambda_1 > \xi_n$.

Proof. Recall that $\xi_n = \sum_{i=1}^{n} (\mathbf{y}_i \cdot \mathbf{n})^2 = \mathbf{n}^T M_n \mathbf{n}$. First, we will show that for any $n > 0$, there always exists two vectors $\mathbf{g}$ and $\mathbf{h}$ such that $(\mathbf{g} \cdot \mathbf{h}) = 0$ and

$$\mathbf{g}^T M_n \mathbf{g} + \mathbf{h}^T M_n \mathbf{h} > 2\xi_n \tag{13}$$

if $\mathbf{y}_i \cdot \mathbf{t}_i > \sqrt{2}\ (\mathbf{y}_i \cdot \mathbf{n})$ for all i.

We prove this by induction. When $n = 1$, we obtain

$$M_1 = (\mathbf{y}_1 \cdot \mathbf{t}_1)^2\, \mathbf{t}_1 \mathbf{t}_1^T + (\mathbf{y}_1 \cdot \mathbf{n})^2\, \mathbf{n}\mathbf{n}^T\,.$$

One can simply choose $\mathbf{g} = \mathbf{t}_1$ and $\mathbf{h}$ to be a unit vector perpendicular to both $\mathbf{g}$ and $\mathbf{n}$. In the three dimensional case, i.e., $m = 3$, we can use $\mathbf{h} = \frac{\mathbf{t}_1 \times \mathbf{n}}{\|\mathbf{t}_1 \times \mathbf{n}\|}$. Then

$$\mathbf{g}^T M_1 \mathbf{g} + \mathbf{h}^T M_1 \mathbf{h} = (\mathbf{y}_1 \cdot \mathbf{t}_1)^2 + 0 > \left[\sqrt{2}\ (\mathbf{y}_1 \cdot \mathbf{n}) \right]^2 = 2\xi_1\,.$$

Now, assuming that (13) holds when $n = k$ and adding one more data point into our data set so that the number of data is $k + 1$, we have

$$\xi_{k+1} = \sum_{i=1}^{k+1} (\mathbf{y}_i \cdot \mathbf{n})^2 \, \xi_k + (\mathbf{y}_{k+1} \cdot \mathbf{n})^2 \,,$$
$$M_{k+1} = \sum_{i=1}^{k+1} \mathbf{y}_i \mathbf{y}_i^T = M_k + (\mathbf{y}_{k+1} \cdot \mathbf{t}_{k+1})^2 \, \mathbf{t}_{k+1}\mathbf{t}_{k+1}^T + (\mathbf{y}_{k+1} \cdot \mathbf{n})^2 \, \mathbf{n}\mathbf{n}^T \,.$$

Since $\{\mathbf{g}, \mathbf{h}\}$ forms an orthonormal basis of the two dimensional tangent plane, we have $\|\mathbf{g}^T\mathbf{t}_{k+1}\|^2 + \|\mathbf{h}^T\mathbf{t}_{k+1}\|^2 = \|\mathbf{t}_{k+1}\|^2 = 1$ for any vector $\mathbf{t}_{k+1}$ on the tangent plane and so,

$$\begin{aligned}
&\mathbf{g}^T M_{k+1}\mathbf{g} + \mathbf{h}^T M_{k+1}\mathbf{h} \\
&\quad = \mathbf{g}^T M_k \mathbf{g} + \mathbf{h}^T M_k \mathbf{h} + (\mathbf{y}_{k+1} \cdot \mathbf{t}_{k+1})^2 \left(\mathbf{g}^T \mathbf{t}_{k+1}\mathbf{t}_{k+1}^T \mathbf{g} + \mathbf{h}^T \mathbf{t}_{k+1}\mathbf{t}_{k+1}^T \mathbf{h}\right) \\
&\quad = \mathbf{g}^T M_k \mathbf{g} + \mathbf{h}^T M_k \mathbf{h} + (\mathbf{y}_{k+1} \cdot \mathbf{t}_{k+1})^2 \left(\|\mathbf{g}^T \mathbf{t}_{k+1}\|^2 + \|\mathbf{h}^T \mathbf{t}_{k+1}\|^2\right) \\
&\quad = \mathbf{g}^T M_k \mathbf{g} + \mathbf{h}^T M_k \mathbf{h} + (\mathbf{y}_{k+1} \cdot \mathbf{t}_{k+1})^2 \\
&\quad > 2\xi_k + (\mathbf{y}_{k+1} \cdot \mathbf{t}_{k+1})^2 > 2\xi_k + \left[\sqrt{2} \, (\mathbf{y}_{k+1} \cdot \mathbf{n})\right]^2 = 2\xi_{k+1} \,.
\end{aligned}$$

To conclude, if we have $\mathbf{y}_i \cdot \mathbf{t}_i > \sqrt{2} \, (\mathbf{y}_i \cdot \mathbf{n})$ for all $1 \le i \le n$, then

$$\mathbf{g}^T M_n \mathbf{g} + \mathbf{h}^T M_n \mathbf{h} > 2\xi_n = 2\mathbf{n}^T M_n \mathbf{n} \,.$$

So it gives either $\mathbf{g}_n^T M_n \mathbf{g}_n > \mathbf{n}^T M_n \mathbf{n}$ or $\mathbf{h}_n^T M_n \mathbf{h}_n > \mathbf{n}^T M_n \mathbf{n}$, which implies that $\mathbf{n}$ cannot be the principal direction of M_n. ∎

Theorem 4.1 follows immediately by these two lemmas. In practice, however, there could be grid points $\mathbf{x}$ far away from the data set $\mathcal{S}$, i.e.,

$$\min_{\mathbf{x}_i \in \mathcal{S}} \|\mathbf{x} - \mathbf{x}_i\|_2 > \epsilon \,,$$

and so the set $\mathcal{S}_\mathbf{x}$ is in fact empty. This implies that $n = 0$ in Theorem 4.1, and so we cannot use it to find an approximation of $\mathbf{p}$. The fix to this problem is not unique. Since we do not actually expect that the zero level set of ϕ would stay clear to such a location, we can maximize the contribution from the fidelity term by setting $\mathbf{p}(\mathbf{x})$ using the unit normal vector of the level set surface given by $\mathbf{n}(\mathbf{x}) = \nabla\phi/\|\nabla\phi\|$. Such a choice, however, would lead to a nonlinear optimization problem. Even though one might linearize the energy by requiring $\mathbf{p}(\mathbf{x})$ to be $\nabla\phi^n/\|\nabla\phi^n\|$, where ϕ^n is the level set function at the previous time step at the gradient flow, we do not recommend this approach since the computational complexity of this step will be expensive. In this work instead, we propose to simply randomly assign a unit vector to $\mathbf{p}(\mathbf{x})$ if $\mathbf{x}$ is far from $\mathcal{S}$. Because adjacent vectors are independent from each other, perturbation of the zero level set will not alter the contribution from the fidelity term to the overall energy in general. This implies that the change in the energy will be automatically contributed mostly from the part with the data mismatch function.

4.4. Level set regularizations. In the previous derivation, we have assumed that the level set function $\phi(\mathbf{x})$ is a signed distance function such that $\|\nabla\phi\| = 1$ for all time $t > 0$ to simplify some computations. This property, however, cannot be satisfied automatically in the gradient flow. This implies that ϕ may develop

steep and flat gradients at or near the zero level set as it evolves according to (7), making the computed location of Γ and further computations inaccurate. Therefore, we use the following regularization procedure, which consists of reinitialization and orthogonalization. To restore the distance property for the level set functions, the usual way is to make ϕ a signed distance function without moving its zero level set appreciably. This can be done by the so-called reinitialization. There are also many well-developed numerical methods to efficiently obtain their numerical solutions. We refer all interested readers to [34] and thereafter for a more detailed discussion.

Following the typical level set approach [34], we first reinitialize the level set functions. We introduce an artificial time τ and solve the following reinitialization equation:

$$\phi_\tau + \text{sign}(\phi^n)(\|\nabla\phi\| - 1) = 0 \tag{14}$$

with the initial condition $\phi(\tau = 0)$ being the intermediate state given by ϕ^n at the nth iteration step for (7). Another regularization is the orthogonal extension from $\psi = 0$ so that two zero level set surfaces are perpendicular to each other. This can reduce the error in extracting the curve explicitly when we need to visualize the intersection of these surfaces. Mathematically, this can be achieved by solving the following orthogonal extension equation

$$\phi_\tau + \text{sign}(\phi^n)\nabla\psi \cdot \nabla\phi = 0\,, \tag{15}$$

which is simply an advection equation with a discontinuous velocity $\mathbf{u} = \text{sign}(\phi^n)\nabla\psi$. There are standard numerical implementations and packages available for solving these two nonlinear equations of the Hamilton–Jacobi type today.

4.5. Improving the computational efficiency. To improve the computational complexity of the algorithm, we also incorporate the following two approaches. The first one is to localize the computations near the zero level set in $\mathcal{M}_\epsilon$. In practice, we are interested only in the solution near the zero level set. There is no reason why we need to solve the level set equation in the whole computational domain $\mathcal{M}_\epsilon$ up to the steady state. In the following examples, we only solve these two regularization equations for one single $\Delta\tau$ step each and use these intermediate numerical solutions to replace the original level set function ϕ^n. To further reduce the computational cost, we also apply the local level set strategy as developed in [37], which can concentrate all computational power in the region near the zero level set within the small neighborhood $\mathcal{M}_\epsilon$.

The second way to improve the computational efficiency of the proposed algorithm is a two-stage strategy for the level set initialization. At the initial time when the zero level set is far away from the dataset, we have proposed in section 4.3 to assign a random vector to $\mathbf{p}(\mathbf{x})$. Because of the independence of any neighboring vectors, we do not expect that the fidelity term containing $\mathbf{p}(\mathbf{x})$ would have much influence on the level set evolution. To further speedup the computations, we propose the following *two-stage strategy*. Since the algorithm simply does not want any contribution from $\mathbf{p}(\mathbf{x})$ when the zero level set is far from the dataset, we start the whole algorithm by first turning the fidelity term off with $\lambda = 0$ and obtain the steady state solution. The zero level set of this first-stage solution is expected to be already close to the dataset. Then, the second stage is to use this steady state solution as the initial condition for the nonzero λ. Since the zero level set of this solution is now near the dateset $\mathcal{S}$, the contribution from $\mathbf{p}(\mathbf{x})$ will provide extra information to better fit the data on the manifold $\mathcal{M}$.

4.6. Numerical discretization. In this section, we discuss the numerical discretization of various equations in the proposed algorithm.

4.6.1. Level set equations. We first discuss the discretization of the equation (7). The equation can be expressed in the form

$$\frac{\partial \phi}{\partial t} = \nabla \cdot [a(\mathbf{x}) \nabla \phi] + \nabla \cdot [(\mathbf{p}(\mathbf{x}) \cdot \nabla \phi)\, \mathbf{b}(\mathbf{x})]\,,$$

where

$$\begin{aligned} a(\mathbf{x}) &= \gamma_1\, d^p(\mathbf{x})\,, \\ \mathbf{b}(\mathbf{x}) &= \lambda \gamma_2 [\mathbf{p}(\mathbf{x}) + \mathcal{P}\mathbf{p}(\mathbf{x})] \\ &= (b^x(\mathbf{x}), b^y(\mathbf{x}), b^z(\mathbf{x}))^T\,, \\ \mathbf{p}(\mathbf{x}) &= (p^x(\mathbf{x}), p^y(\mathbf{x}), p^z(\mathbf{x}))^T\,. \end{aligned}$$

The diffusion term on the right-hand side of the equation can be easily discretized using the symmetric central difference given by

$$\begin{aligned} &\frac{1}{\Delta x^2}\left[a_{i-\frac{1}{2},j,k}\phi_{i-1,j,k} - \left(a_{i-\frac{1}{2},j,k} + a_{i+\frac{1}{2},j,k}\right)\phi_{i,j,k} + a_{i+\frac{1}{2},j,k}\phi_{i+1,j,k}\right] \\ &\quad + \frac{1}{\Delta y^2}\left[a_{i,j-\frac{1}{2},k}\phi_{i,j-1,k} - \left(a_{i,j-\frac{1}{2},k} + a_{i,j+\frac{1}{2},k}\right)\phi_{i,j,k} + a_{i,j+\frac{1}{2},k}\phi_{i,j+1,k}\right] \\ &\quad + \frac{1}{\Delta z^2}\left[a_{i,j,k-\frac{1}{2}}\phi_{i,j,k-1} - \left(a_{i,j,k-\frac{1}{2}} + a_{i,j,k+\frac{1}{2}}\right)\phi_{i,j,k} + a_{i,j,k+\frac{1}{2}}\phi_{i,j,k+1}\right]\,, \end{aligned}$$

where

$$\begin{aligned} a_{i\pm\frac{1}{2},j,k} &= \frac{1}{2}\left[a(\mathbf{x}_{i,j,k}) + a(\mathbf{x}_{i\pm1,j,k})\right]\,, \\ a_{i,j\pm\frac{1}{2},k} &= \frac{1}{2}\left[a(\mathbf{x}_{i,j,k}) + a(\mathbf{x}_{i,j\pm1,k})\right]\,, \\ a_{i,j,k\pm\frac{1}{2}} &= \frac{1}{2}\left[a(\mathbf{x}_{i,j,k}) + a(\mathbf{x}_{i,j,k\pm1})\right]\,. \end{aligned}$$

The second term on the right-hand side is a little tedious since it consists of nine mixed derivatives of ϕ including $[b^x p^x \phi_x]_x$, $[b^x p^y \phi_y]_x$, $[b^x p^z \phi_z]_x$, $[b^y p^x \phi_x]_y$, $[b^y p^y \phi_y]_y$, $[b^y p^z \phi_z]_y$, $[b^z p^x \phi_x]_z$, $[b^z p^y \phi_y]_z$ and $[b^z p^z \phi_z]_z$. Numerically, we construct the projection $\mathcal{P}\mathbf{p}_{i,j,j}$ in $\mathbf{b}(\mathbf{x}_{i,j,k})$

$$\mathcal{P}\mathbf{p}_{i,j,k} = \mathcal{P}\mathbf{p}(\mathbf{x}_{i,j,k}) = \mathbf{p}_{i,j,k} - (\mathbf{n}_{i,j,k} \cdot \mathbf{p}_{i,j,k})\, \mathbf{n}_{i,j,k}$$

with $\mathbf{n}_{i,j,k} = \nabla \phi_{i,j,k}$ computed using central difference. Then, we approximate those mixed derivatives of ϕ using central difference. To simplify the following notation, we introduce $c^{x,x} = b^x p^x$, $c^{x,y} = b^x p^y$, etc. Now, we have

$$\begin{aligned} &[c^{x,x}\phi_x]_x \\ &\quad = \frac{1}{\Delta x^2}\left[c^{x,x}_{i-\frac{1}{2},j,k}\phi_{i-1,j,k} - \left(c^{x,x}_{i-\frac{1}{2},j,k} + c^{x,x}_{i+\frac{1}{2},j,k}\right)\phi_{i,j,k} + c^{x,x}_{i+\frac{1}{2},j,k}\phi_{i+1,j,k}\right] \\ &[c^{x,y}\phi_y]_x \\ &\quad = \frac{1}{4\Delta x \Delta y}\left[c^{x,y}_{i+1,j,k}(\phi_{i+1,j+1,k} - \phi_{i+1,j-1,k}) - c^{x,y}_{i-1,j,k}(\phi_{i-1,j+1,k} - \phi_{i-1,j-1,k})\right]\,, \end{aligned}$$

where $c^{x,y}_{i\pm\frac{1}{2},j,k} = (c^{x,y}_{i,j,k} + c^{x,y}_{i\pm1,j,k})/2$.

Finally, the temporal derivative is approximated by the forward Euler method with the stability condition

$$\Delta t < \frac{1}{2}\frac{\Delta x^2}{\max(\|a(\mathbf{x})\|, \|\mathbf{b}(\mathbf{x})\|)}\,.$$

4.6.2. Level set regularizations. The reinitialization equation (14) and the orthogonal extension equation (15) are both Hamilton–Jacobi equations. There are well-developed algorithms for obtaining highly accurate numerical solutions. In this work, we apply the TVDRK [39] and the WENO [31, 18] for the temporal and the spatial derivatives, respectively. The Hamiltonian is approximated by the Lax–Friedrichs Hamiltonian. We will just briefly state the discretization procedure here but refer readers to the above references and thereafter for some detail discussions. For simplicity, we consider only the one dimensional Hamilton–Jacobi equation given by

$$\frac{\partial \phi}{\partial \tau} + H\left(\frac{\partial \phi}{\partial x}\right) = 0\,.$$

High dimensional generalization is straightforward. First, we discuss the spatial discretization of the equation. We approximate the Hamiltonian by the Lax–Friedrichs Hamiltonian given by

$$\hat{H}^{\mathrm{LxF}}\left(\left.\frac{\partial \phi}{\partial x}\right|_{x=x_i}\right) = H\left(\frac{p_i^+ + p_i^-}{2}\right) - \sigma_x\left(\frac{p_i^+ - p_i^-}{2}\right),$$

where the derivatives $p^{\pm}$ are approximated by the WENO strategy. The viscosity parameter σ_x is chosen so that for any x_i we have the monotonicity requirements given by

$$\sigma_x \geq \max_{\phi,\phi_x}\left|\frac{\partial \hat{H}^{\mathrm{LxF}}(\phi_x)}{\partial \phi_x}\right|.$$

Taking the third order WENO (i.e., WENO3) as an example, we have

$$p_i^- = (1-\omega_-)\left(\frac{\phi_{i+1}-\phi_{i-1}}{2\Delta x}\right) + \omega_-\left(\frac{3\phi_i - 4\phi_{i-1} + \phi_{i-2}}{2\Delta x}\right),$$
$$p_i^+ = (1-\omega_+)\left(\frac{\phi_{i+1}-\phi_{i-1}}{2\Delta x}\right) + \omega_+\left(\frac{-3\phi_i + 4\phi_{i+1} - \phi_{i+2}}{2\Delta x}\right)$$

with $\omega_\pm = (1+2\gamma_\pm^2)^{-1}$ and

$$\gamma_- = \frac{\delta^2 + (\phi_i - 2\phi_{i-1} + \phi_{i-2})^2}{\delta^2 + (\phi_{i+1} - 2\phi_i + \phi_{i-1})^2} \text{ and } \gamma_+ = \frac{\delta^2 + (\phi_i - 2\phi_{i+1} + \phi_{i+2})^2}{\delta^2 + (\phi_{i+1} - 2\phi_i + \phi_{i-1})^2}$$

for some small positive constant δ to avoid division by zero. Concerning the temporal derivative, we apply the method of line strategy. In particular, at each grid point, we update the solution using

$$\hat{\phi}_i^{m+1} = \phi^m - \Delta\tau\,\hat{H}^{\mathrm{LxF}}\left(\left.\frac{\partial \phi^m}{\partial x}\right|_{x=x_i}\right),$$
$$\hat{\phi}_i^{m+2} = \hat{\phi}^{m+1} - \Delta\tau\,\hat{H}^{\mathrm{LxF}}\left(\left.\frac{\partial \hat{\phi}^{m+1}}{\partial x}\right|_{x=x_i}\right),$$
$$\phi_i^{m+1} = \frac{1}{2}\left(\phi_i^m + \hat{\phi}_i^{m+2}\right).$$

Because of the CFL stability condition, the time marching step is chosen to be $\Delta\tau = O(\Delta x)$.

4.7. The overall algorithm. The proposed method is summerized in Algorithm 2. Now, we discuss the computational complexity of the overall algorithm. Let N_x be the number of the grid points in each physical dimension and N be the total number of grid points in the small neighborhood of the manifold $\mathcal{M}$ denoted by $\mathcal{M}_\epsilon$. We first discuss the computational cost of the implicit representation of $\mathcal{M}$ used in the proposed method and that by a typical explicit representation using triangulations. Since $\mathcal{M}$ is a codimensional one surface in $\mathbb{R}^3$, we have $N = O(N_x^2)$. Therefore the order of these two costs are in fact the same which is in fact optimal. Now, it takes $O(N)$ operations to determine $g(\mathbf{x})$, $d(\mathbf{x})$, and also $\mathbf{p}(\mathbf{x})$ because these functions are defined in a pointwise fashion. For each iteration step, it also takes $O(N)$ operations to update and regularize the level set function ϕ if we do not implement the local level set approach but to update ϕ for all $\mathbf{x} \in \mathcal{M}_\epsilon$. If we further concentrate the computational power near the zero level set in $\mathcal{M}_\epsilon$, the computational complexity of each iteration can be further reduced to $O(N_x)$.

Algorithm 2 The proposed algorithm with the two-stage strategy.

Data: The data set $\mathbf{x}_s$, the mesh size $(\Delta x, \Delta y, \Delta z)$, the level set representation of the manifold ψ_i, the weight λ for the data fidelity
Result: Γ represented by the intersection of zero level set of ϕ and ψ
Initialization: Set ϕ such that Γ encloses all data points
Compute $g(\mathbf{x})$ by Algorithm 1;
Determine $d(\mathbf{x})$ by smoothing $g(\mathbf{x})$ using (11);
Find $\mathbf{p}(\mathbf{x})$ by (12);
for $\lambda' = 0$ *and then set* $\lambda' = \lambda$ **do**
  while $E(\phi; \lambda')$ *in* (5) *does not converge* **do**
    Update ϕ using (7) according to section 4.6.1;
    Regularize ϕ using (14) and (15) according to section 4.6.2;
  end
end

5. Reconstruction using an open curve. In the previous discussions, we have assumed that $\Gamma = \{\mathbf{x} \in \mathbb{R}^m : \psi(\mathbf{x}) = \phi(\mathbf{x}) = 0\}$ is a closed codimensional-one manifold of $\mathcal{M}$. When we try to determine an open curve on a two dimensional $\mathcal{M}$, on the other hand, we cannot simply implicitly represent Γ by $\{\mathbf{x} \in \mathbb{R}^m : \psi(\mathbf{x}) = \phi(\mathbf{x}) = 0\}$. Instead, we follow [40] and introduce a third level set function ρ to implicitly represent the end points of Γ by $\partial\Gamma = \{\mathbf{x} \in \mathbb{R}^m : \psi(\mathbf{x}) = \phi(\mathbf{x}) = \rho(\mathbf{x}) = 0\}$. In particular, we define the low dimension representation of the data set by $\Gamma = \{\mathbf{x} \in \mathbb{R}^m : \psi(\mathbf{x}) = \phi(\mathbf{x}) = 0 \text{ and } \rho(\mathbf{x}) < 0\}$.

The choice of ρ for the same curve is clearly not unique. Assuming that we are given the locations of $\partial\Gamma$, we propose the following algorithm to initialize the level set functions ϕ and ρ if the manifold $\mathcal{M}$ is the unit sphere $\mathbb{S}^2$ with the origin denoted by O. We denote the given two end points of Γ by A and B. If A and B are not on the opposite poles of the sphere, we find the midpoint along the geodesic between A and B on $\mathbb{S}^2$ and we denote that point by C_0. We let P be the plane containing these points A, B, and C_0. Clearly the intersection of P and $\mathbb{S}^2$ gives the unique great circle passing through both points A and B. Now, we construct the plane Q which is perpendicular to P and passes through C_0 and the origin O. The intersection of Q

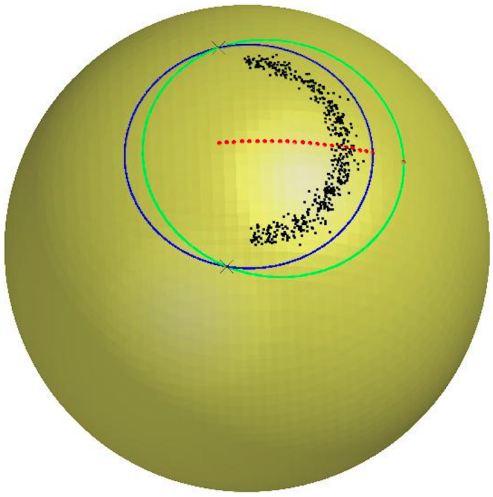


FIG. 3. *Reconstructing the data set using an open curve. The given data set is given in the black dots. The black crosses is the given two end points. The red points are traces of d we tested. The green curve represents the intersection of the zero level set of $\rho(\mathbf{x})$ and $\mathcal{M}$ where we identify regions where the intersection of the zero level set function of ϕ and $\mathcal{M}$, the blue curve, is updated.*

and $\mathbb{S}^2$ will give another great circle on $\mathbb{S}^2$. We denote this great circle by $\mathcal{C}$ which can be parametrized by $C(\theta)$ with $C(0) = C_0$. A typical situation is shown in Figure 3. We plot the given data set on $\mathbb{S}^2$ by black dots. The end points $\partial\Gamma$ are plot by black cross. The locations of various $C(\theta)$ are plotted in red dots. For each particular θ, we can determine unique signed distance function ϕ_θ containing A, B. and $C(\theta)$, in the form of

$$\phi_\theta(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_\theta\| - r_\theta \, .$$

Finally, we gradually increase (or decrease) the value of θ from 0 until we find a value θ^* such that $\phi_{\theta^*}(\mathbf{x}_i) < 0$ for all given data points $\mathbf{x}_i$. This implies that the zero level set of ϕ_{θ^*} now encloses all data points in $\mathbb{S}^2$ and, therefore, the intersection of $\{\phi_{\theta^*} = 0\}$ and $\mathbb{S}^2$ gives a good initial condition of ϕ. We are plotting this intersection by the blue curve in Figure 3. Once we have ϕ constructed from θ^*, we further increase (or decrease) the value of θ by ϵ and we can use it to define $\rho(\mathbf{x}) = \phi_{\theta^* + \epsilon \mathrm{sign}(\theta^*)}(\mathbf{x})$. The zero level set of ρ on $\mathbb{S}^2$ is plotted by green solid line in Figure 3. The black dots is the given data set. The black crosses is the given two end points. The red points are a trace of d we tested. The green circle is the intersection of the zero level set of $\rho(\mathbf{x})$ and $\mathcal{M}$ such that all data points on the black curve is in the interior. The intersection of the zero level set function of ϕ and $\mathcal{M}$ is plotted in blue.

The overall algorithm for this case is almost the same as the one we developed above, except for the way we define Γ and and that we have an extra constraint where we update the level set function. The overall algorithm is summarized in Algorithm 3.

6. Numerical examples and discussions. In this section, we consider various examples to demonstrate the effectiveness of the proposed algorithm. Unless otherwise specified, we have chosen $\lambda = 0.01$ in all of our numerical experiments. In the following examples, we first analytically determine a target curve on the underlying manifold. These target solutions are plotted in red solid lines. Then the data points are randomly drawn from an exact target curves. In the case when we would like to

Algorithm 3 The proposed algorithm for an open curve with the two-stage strategy.

Data: The data set $\mathbf{x}_s$, the mesh size $(\Delta x, \Delta y, \Delta z)$, the level set representation of the manifold ψ_i, the weight λ for the data fidelity
Result: Γ represented by the intersection of zero level set of ϕ and ψ
Initialization: Set ϕ such that Γ encloses all data points, and ρ according to section 5
Compute $g(\mathbf{x})$ by Algorithm 1;
Determine $d(\mathbf{x})$ by smoothing $g(\mathbf{x})$ using (11);
Find $\mathbf{p}(\mathbf{x})$ by (12);
for $\lambda' = 0$ *and then set* $\lambda' = \lambda$ **do**
  while $E(\phi; \lambda')$ *in* (5) *does not converge* **do**
    if $\rho(\mathbf{x}_i) < 0$ **then**
      Update ϕ using (7) according to section 4.6.1;
    end
    Regularize ϕ using (14) and (15) according to section 4.6.2;
  end
end

test the robustness of the proposed algorithm, we add random Gaussian noise with the standard deviation equal to $\frac{h}{2}$ to these data with a constraint that the perturbed locations are still staying on the surface. These data sets are plotted using black dots, while the level set $\{\phi = 0\}$ is shown in blue solid line.

6.1. Closed curves on a sphere and a torus. Figure 4(a)–(f) show the reconstruction using the proposed algorithm without and with the fidelity term using the estimation by the PCA. The variational formulation can robustly determine a curve Γ which fits the data reasonably well even when there is noise in the data, as shown in Figure 4(c). In this example, the fidelity term from the PCA does not contribute much new information for the reconstruction. The iteration stops almost immediately when we switch from $\lambda = 0$ to a nonzero λ in the two-stage strategy. Therefore, for simple cases like these we are testing on, we simply start with a nonzero λ immediately. In Figure 5, we show the change in the energy in the iterations for all these cases. It is clearly that the mismatch energy is gradually reduced to its steady state as we evolve Γ according to the proposed method. For this example, it takes approximately 500 iterations to get to these final solutions.

The algorithm can be easily extended to manifolds other than a sphere. In Figure 6, we consider the reconstruction of a circular curve on a torus. The result by our algorithm can naturally estimate the target curve even when there are significant amount of noise in the given data.

One interesting example is to reconstruct a nonconvex object using the current approach. As the zero level set shrinks, we would like to know if the total length of the level curve would be able to increase itself to better fit the data at some point of the evolution. Figure 7 shows a scenario where the zero level set of $\phi(\mathbf{x})$ touches a six-folded star shape at its vertices. After this particular time, the curve will further shrink a little until the total length has to increase if we want to reach the *global* minimum of the functional.

To simplify the discussion, we approximate one-sixth of the above curve by a semicircle of radius 1, as shown in Figure 8(a). In this analysis, we determine the resulting energy of various segments and try to conclude that the weight based on the distance function $d(\mathbf{x})$ indeed allows an elongation of the curve to better fit the data.

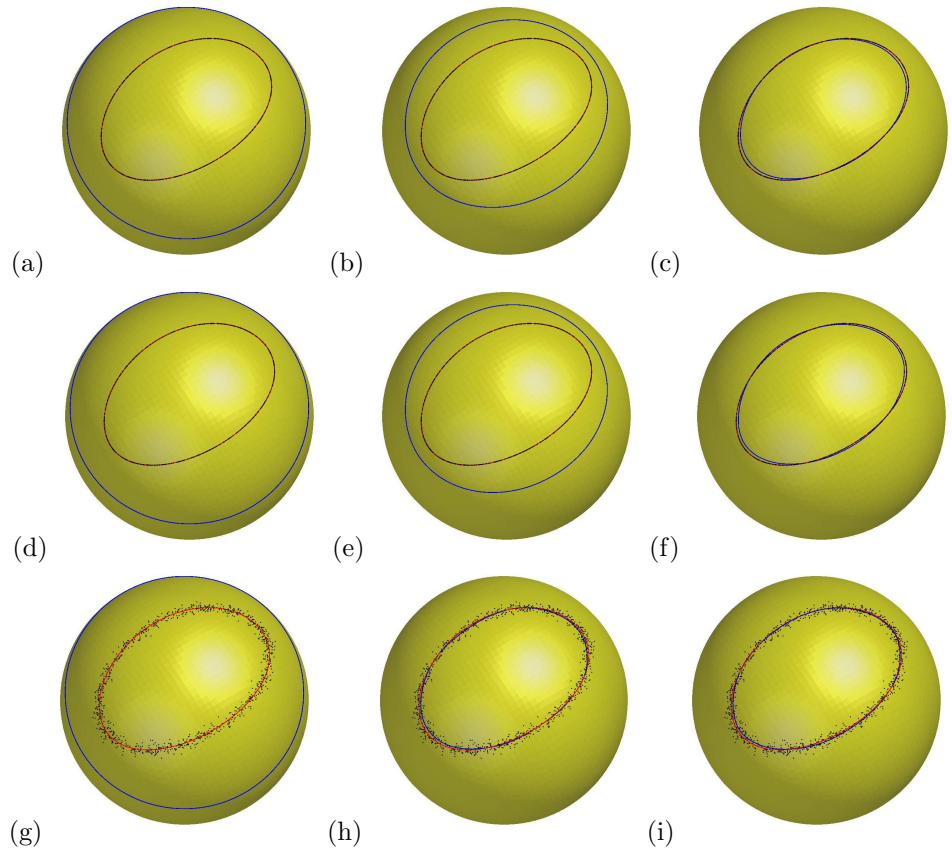


FIG. 4. *(Example 6.1) Ellipsoidal curves on a sphere.* (a)–(c) *Reconstruction using* $\lambda = 0$. (a) *The initial condition,* (b) *an intermediate solution, and* (c) *the final solution.* (d)–(f) *Reconstruction using* $\lambda = 0.01$. (d) *The initial condition,* (e) *an intermediate solution, and* (f) *the final solution.* (g) *The dataset with noise. Our reconstruction results using* (h) $\lambda = 0$ *and* (i) $\lambda = 0.01$.

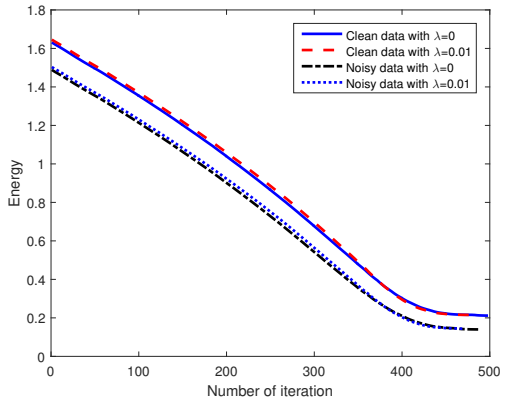
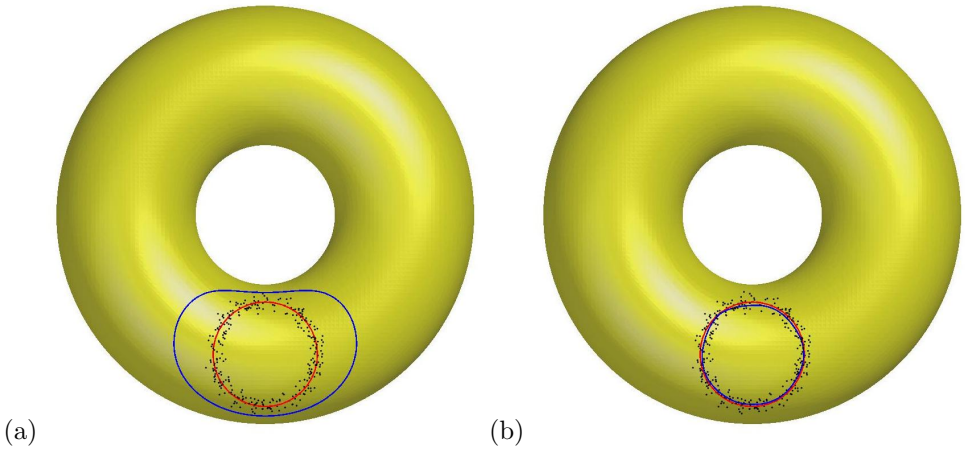


FIG. 5. *(Example 6.1) Reconstructing an ellipsoidal curve on a sphere. The change in the energy for various tests for the clean data and noisy data with* $\lambda = 0$ *and* $\lambda = 0.01$.



(a) (b)

FIG. 6. *(Example 6.1) A closed target curve on a torus.* (a) *Our initial condition.* (b) *Result by our variational formula.*

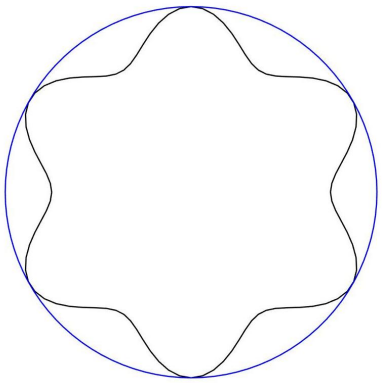
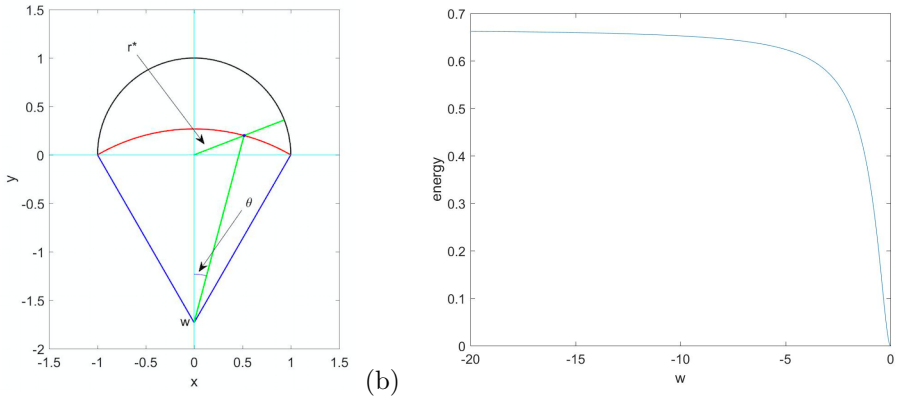
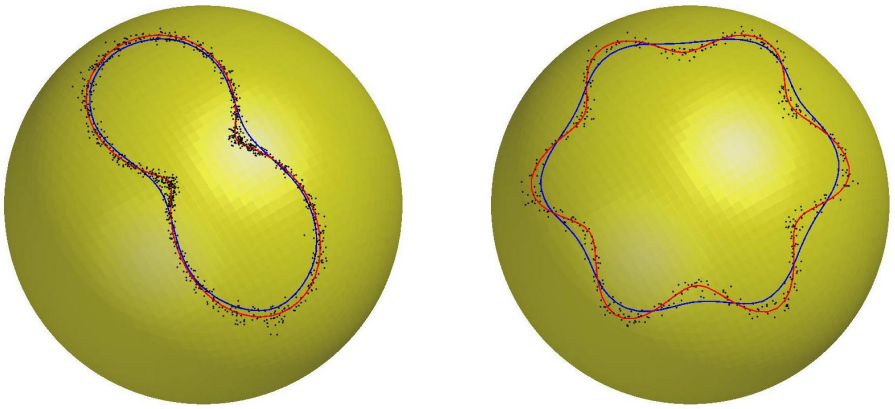


FIG. 7. *(Example 6.1) Reconstructing a nonconvex curve. The zero level set of ϕ is given in blue and is touching the target at the vertices.*



(a) (b)

FIG. 8. *(Example 6.1)* (a) *The black semicircle with radius* 1 *is the data set. The red arc is part of a circle centered at* $(0, w)$ *with* $w \leq 0$. *And assume it is the evolution of* Γ *with end points being the same as the semicircle and being fixed.* (b) *Energy of the red arc for different* w.



(a) (b)

FIG. 9. *(Example* 6.1*)* (a) *A two-folded and* (b) *a six-folded curves on a sphere. Our reconstructions are drawn in blue solid lines.*

We consider that the part of the smaller arc of a circle centered at $(0, w)$ touches the semicircle at $(\pm 1, 0)$ for some $w < 0$, denoted by the red segment in Figure 8(a). As w increases from $-\infty$ to 0, Γ evolves from the diameter of the semicircle to the semicircle itself. For each point $\mathbf{z}$ on this arc, we define θ to be the angle it made with the positive y-axis and r^* to the its distance from the origin. The shortest distance from $\mathbf{z}$ to the black semicircle is therefore given by $d(\mathbf{z}) = 1 - r^*$ with

$$(r^*)^2 = 1 + 2w^2 - 2|w|\sqrt{1+w^2}\cos\theta\,.$$

This implies that the corresponding energy associated to this arc is

$$E(w) = 2\sqrt{1+w^2}\int_0^{\arctan\frac{1}{|w|}} (1-r^*)^2 d\theta\,.$$

Figure 8(b) shows the energy $E(w)$ as we vary w. While the length of the arc increases from 2 to π as w changes from $-\infty$ to 0, the energy decreases monotonically to 0 reaching the *global* minimizer of the energy.

Two simple tests of the algorithm on nonconvex shapes are shown in Figure 9. We first randomly draw some data points from a two-folded and a six-folded curves, as shown in (a) and (b), respectively. Gaussian noise is added to these points while constraining that the final location is still on $\mathbb{S}^2$. Our algorithm can still reach the target curve reasonably well, as shown by the blue solid curve.

Because of the implicit representation of the reconstruction, the proposed method does not require any a priori information on the topology of the underlying dataset. For example, in Figure 10, we are given some noisy data on a sphere generated from two circular curves. Once again, we start with an initial zero level set surrounding all data points, as shown in Figure 10(a). As the level set evolves according to the proposed algorithm, it automatically splits into two disjoint curves without any user intervention as demonstrated in (b) and (c). The test example is a challenging one for explicit methods like the original principle flow method since the user has to be able to segment the dataset into two disjoint groups and then identify an initial point for each of these groups.

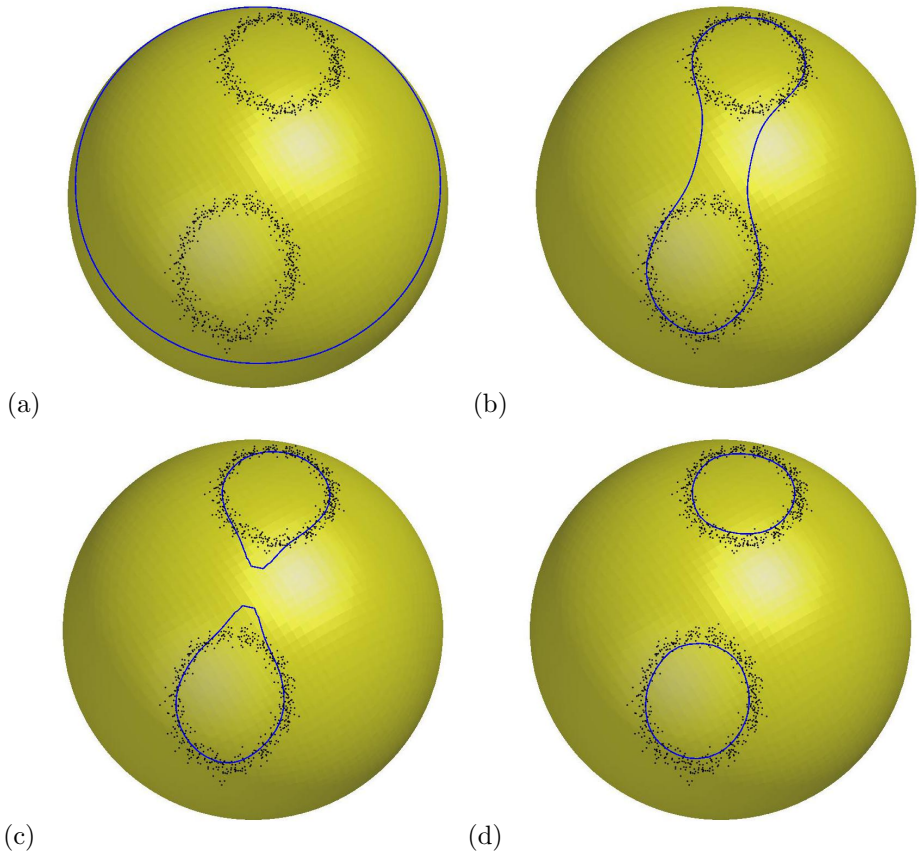


FIG. 10. *(Example* 6.1*) Reconstructing two circular curves on a sphere with* $\lambda = 0.01$. (a) *The initial condition,* (b)–(c) *two intermediate solutions showing the topological change in the level set evolution and* (d) *the final solution.*

6.2. Open curves on a sphere. In this example, we consider three cases where the target structures are represented by open curves given by a C-shape curve, a sine function, and a noised C-shape curve. The result for the first two cases are shown in Figure 11. We can see that our variational formula can naturally handle these open curves. In the case of the C-shape curve, the initial condition does not actually prefer the final solution since the curve Γ has to increase its total length. Nevertheless our computed solution represents very well the given data. For the case with the noised C-shape, we have also compared our result with that by the principal flow method developed in [36]. The results are shown in Figure 12. We have used two different ways to initialize the principal flow method. The first one is based on the Fréchet mean of the dataset which gives a very rough initial starting point of the tracing algorithm. Because of this, the solution from the principal flow does not give a good reconstruction of the data, as shown in Figure 12(c). To incorporate more information to the reconstruction, we also initialize the principal flow method using the midpoint of the underlying clean C-shape curve. The reconstruction result is shown in Figure 12(d). The method can now reconstruct the underlying manifold pretty well. In real applications, however, the midpoint to the *unknown* underlying curve is unavailable. Compared to the original principle flow method, the proposed

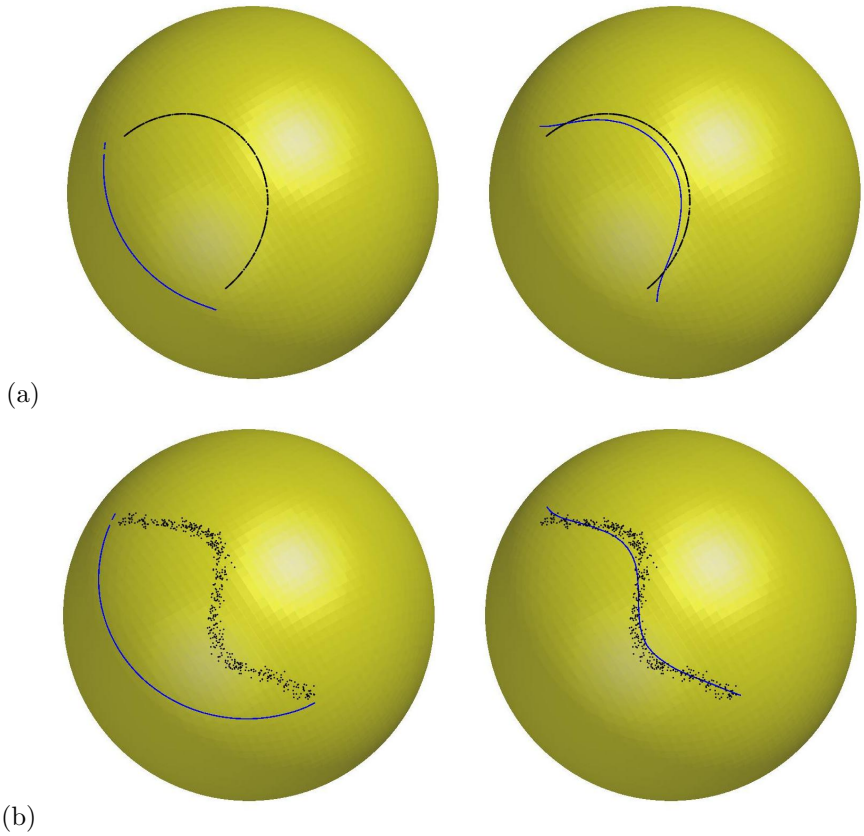


FIG. 11. *(Example* 6.2*) Open curves on a sphere. We plot the given data points on the sphere using black dots. Our solutions* Γ *are plotted in blue. (Left) The initial condition and* (b) *the reconstructions.*

method can automatically drive the evolution starting from an initial level set function enclosing the data to a reasonably good reconstruction.

6.3. Examples with outlier. To further test the performance of our proposed algorithm under noise, we manually add some outliers to the noisy data sets. We consider a circular target curve and also an open curve on $\mathbb{S}^2$. We plot the evolutions of our reconstruction under the datasets with and without outliers in Figure 13.

In these examples, we found that the steady state solutions do not heavily depend on the choice of λ. For a zero λ, the proposed method can already capture the expected trend in the data on the manifold. When followed by a nonzero λ in the two-stage strategy, the method stops almost immediately in a few extra iterations. The corresponding energy evolutions are shown in Figure 14. We observe that both red dashed lines take more iterations to reach the steady state. It implies that more iterations are required to get over the outliners in the data set. But more importantly, the existence of outliers in the given data does not affect our final reconstruction. It seems that the steady state solution is the same as the one obtained by the corresponding dataset without the outliers, as demonstrated in Figure 13, and that the final energies are comparable in Figure 14.

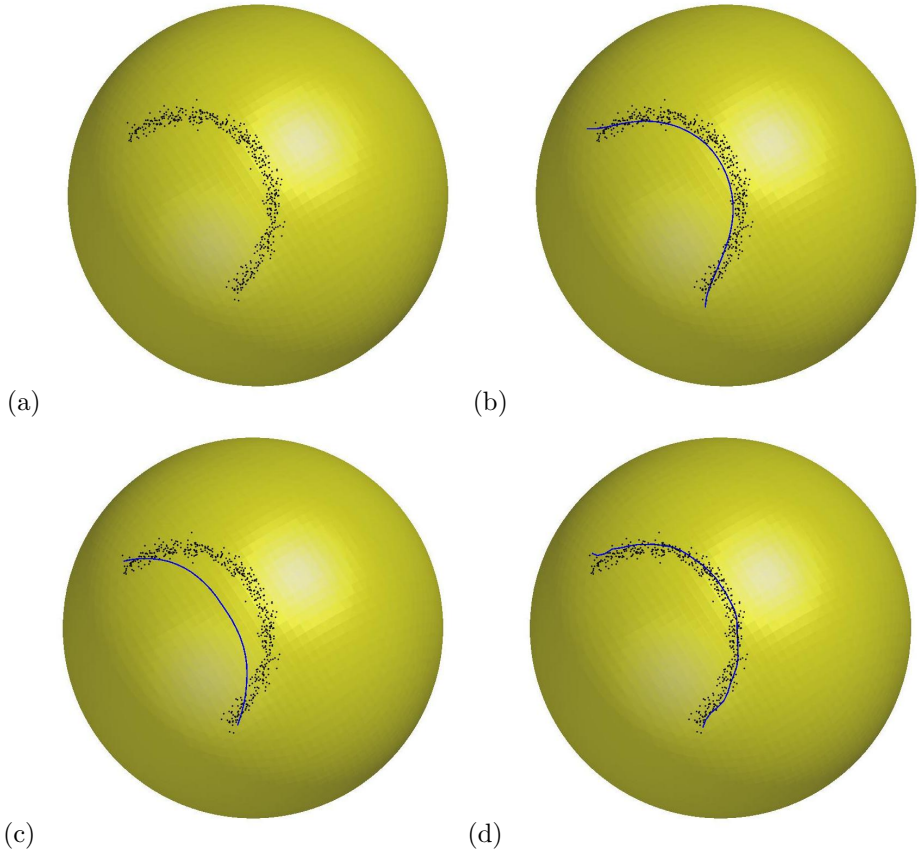


FIG. 12. *(Example 6.2)* (a) *Data sampled from a C-shape curve on a sphere with noise.* (b) *The result by our proposed approach.* (c) *The result by the principal flow method with the starting point being the Fréchet mean.* (d) *The result by the principal flow method with the starting point being the midpoint of the underlying clean C-shape curve.*

6.4. Examples with incomplete information. It is in general not reasonable to assume that the given data can perfectly resolve the underlying geometry on the manifold. In this problem, we assume that the given data can only sample part of the target shape. In particular, we use only partial data from a square on a sphere with missing data at all four corners. The reconstruction from our algorithm for various λ's are shown in Figure 15. Without the matching from the PCA, i.e., $\lambda = 0$, our solution is not able to recover the corner since the algorithm takes into account only the location of the data points on the manifold. However, as we increase the value of λ, from 0 in (a) to $\lambda = 1$ in (d), the contribution from the PCA gives a more realistic reconstruction and helps the curve Γ to predict a better trend in the data. Figure 16 shows the change in the energy with different λ's. In Figure 16(a), we show the energy in the first stage of the iterations when we have $\lambda = 0$. It takes around 540 iterations to reach the steady state solution. In the second stage of the algorithm, we then use this steady state solution as the initial condition for different nonzero λ. This explains why there are jumps in the energies as shown in Figure 16(b).

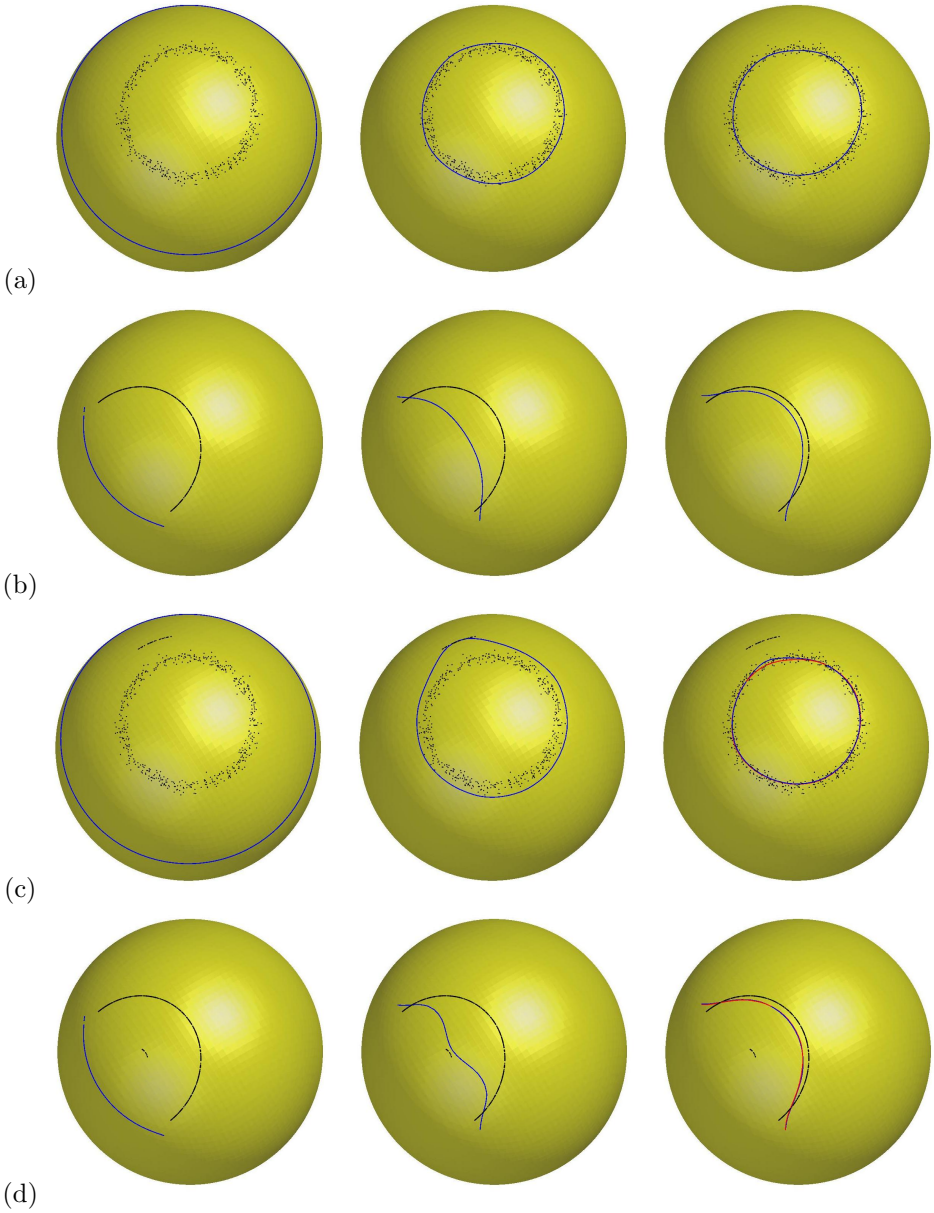


FIG. 13. *(Example 6.3) Data points with outlier are plotted in black.* (a)–(b) *Evolutions of the zero level set for data sets without outliers.* (c)–(d) *Evolutions of the zero level set for the corresponding data sets with outliers. The red curves are final results from* (a) *and* (b)*, respectively.*

6.5. Examples based on earthquake data. In this example, we test our algorithm using the earthquake data from the U.S. geological survey. We consider two datasets containing the epicenter of the earthquakes with magnitude larger than or equal to 4 that occurred between December 29, 2014 and January 1, 2016 in the Russia–Alaska region and in the Australia region, as shown in Figure 17. Our initial conditions and results for these two data sets are shown in Figure 18(a) and (b), respectively.

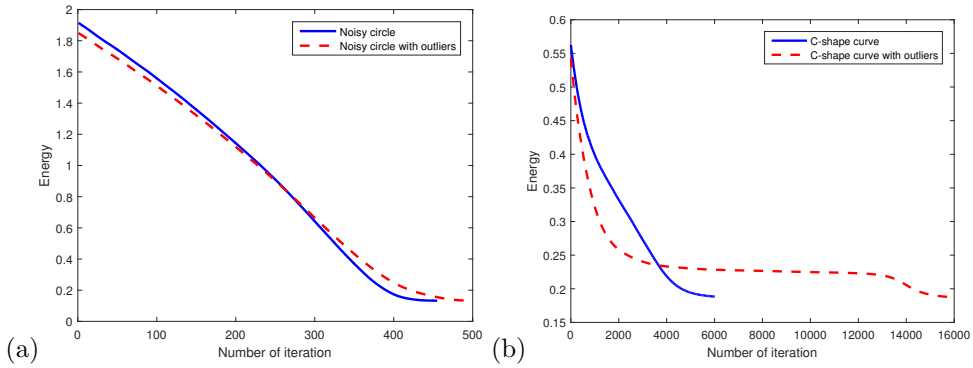


FIG. 14. *(Example* 6.3*) Changes in the energy versus the iteration number.* (a) *Energies corresponding to the test examples in Figure* 13(a) *and* (c). (b) *Energies corresponding to the test examples in Figure* 13(b) *and* (d).

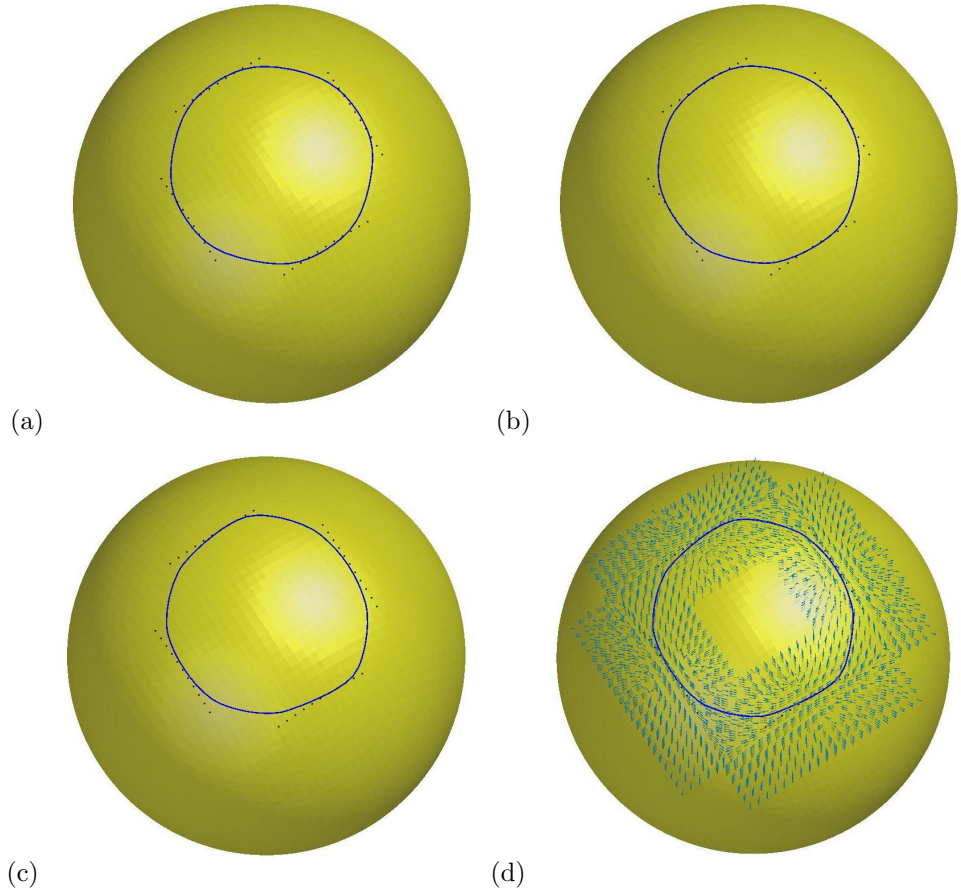
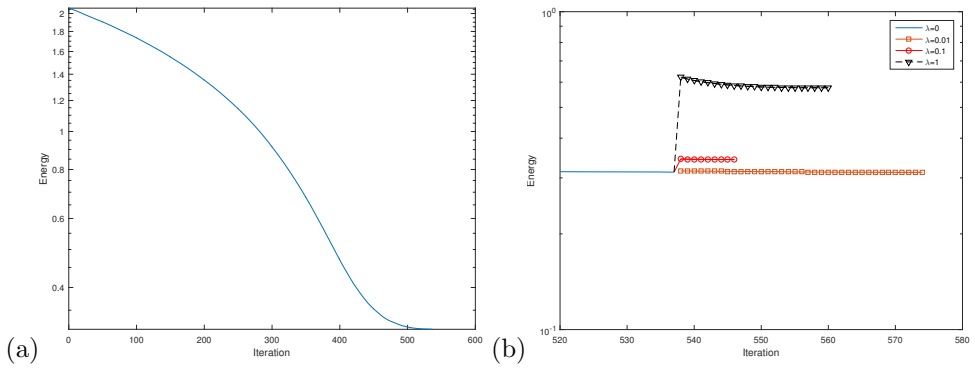
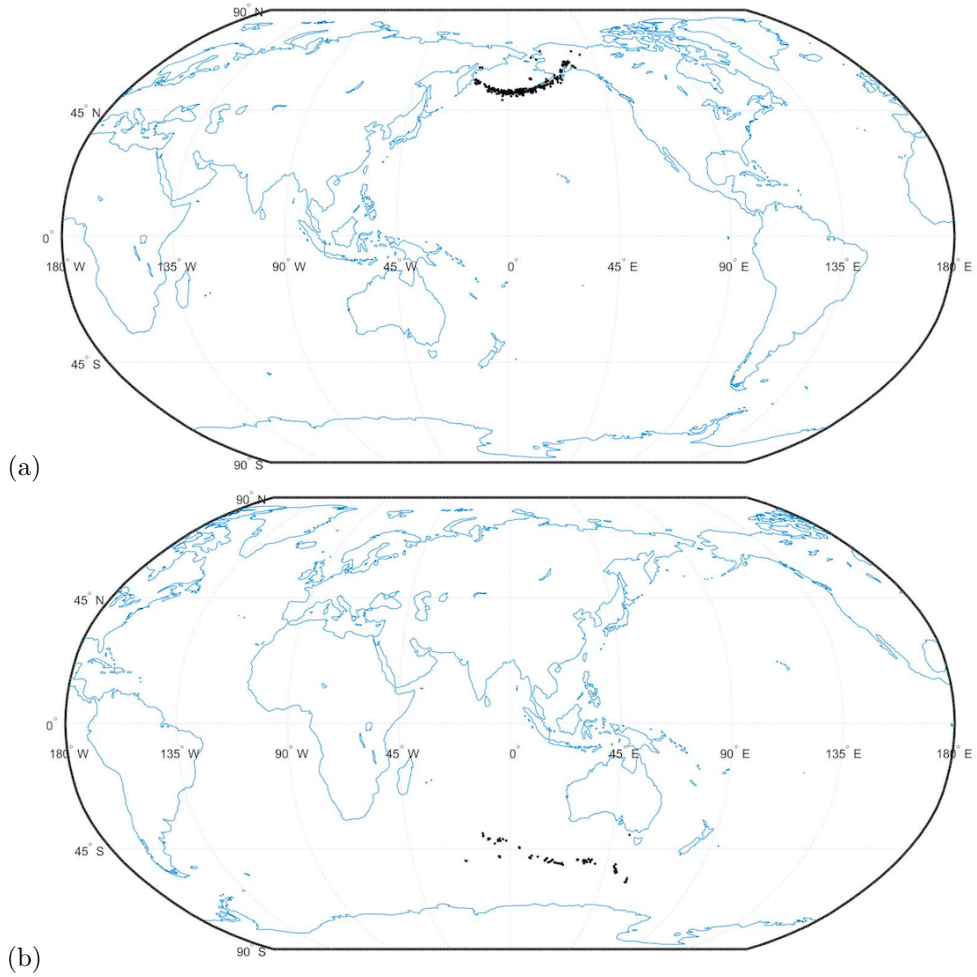


FIG. 15. *(Example* 6.4*) An example when we miss data points at some place on the original curve. Given data points are plotted in black dots. Our reconstruction based on* (a) $\lambda = 0$, (b) $\lambda = 0.01$, (c) $\lambda = 0.1$, *and* (d) $\lambda = 1$. *In* (d), *we have also plotted the PCA vectors near the data points.*



(a) (b)

FIG. 16. *(Example 6.4) The change in the energy with* (a) $\lambda = 0$ *followed by* (b) $\lambda = 0.01$, $\lambda = 0.1$, *and* $\lambda = 1$, *respectively.*



(a)

(b)

FIG. 17. *(Example 6.5) The location of the epicenters of earthquakes with magnitude larger than or equal to* 4 *and occurred between December* 29, 2014 *and January* 1, 2016. (a) *Epicenters of the earthquake in the Russia–Alaska region.* (b) *Epicenters of the earthquake in the Australia region.*

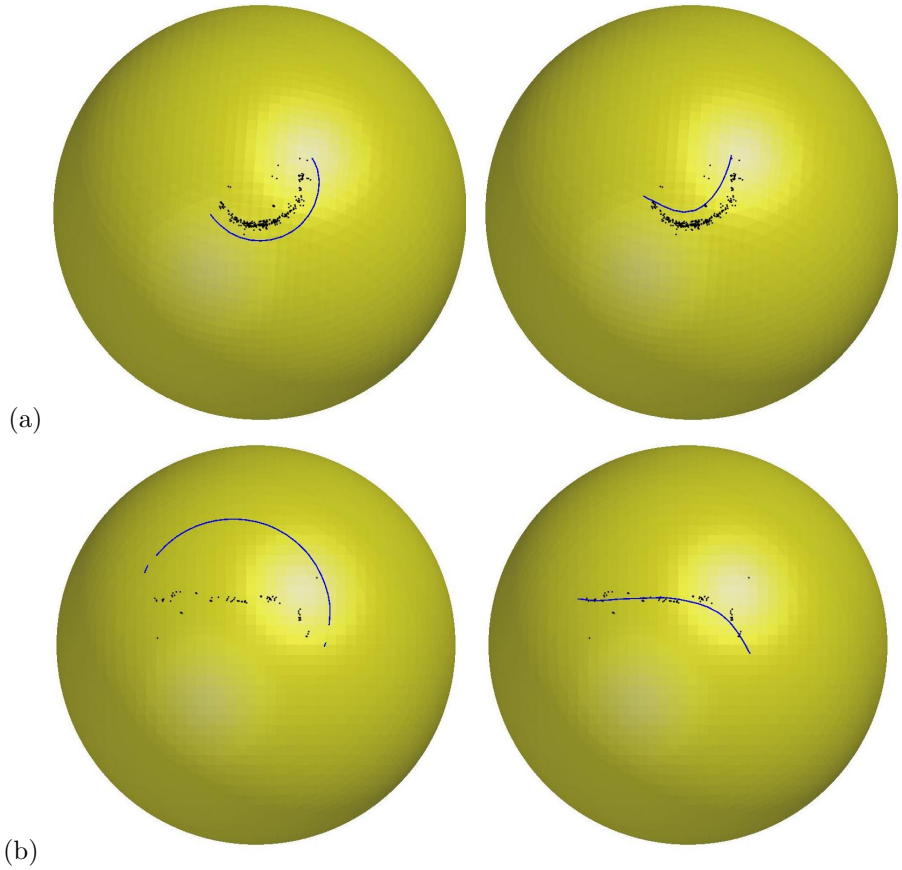


FIG. 18. *(Example* 6.5*)* (a) *Our initial condition and result for earthquakes in the Russia–Alaska region.* (b) *Initial condition and result in the Australia region.*

7. Conclusion. The paper proposed a variational formulation for dimension reduction on Riemannian manifolds. The algorithm is developed based on the level set method to obtain a fully implicit formulation so that the method does not require any *a prior* knowledge on the topology of the structure we are reconstructing. We have tested the algorithm on various numerical examples with different manifolds. Even with given measurements with noise or outliers, the proposed algorithm gives good and robust reconstructions.

There are several possible extensions of our proposed algorithm. Even though it might need a careful study of the data structure such as [33, 8], the algorithm can be extended to handle even higher dimensional problems. Furthermore, since the method is a variational approach and does not require any labeled response from the user, such unsupervised algorithm has the potential to be extended to a manifold learning where the embedding is not necessarily known.

REFERENCES

[1] N. AMENTA AND M. BERN, *Surface reconstruction by Voronoi filtering*, in Proceedings of the 14th ACM Symposium on Computational Geometry, 1998.

[2] M. BERTALMIO, L.-T. CHENG, S. OSHER, AND G. SAPIRO, *Variational problems and partial differential equations on implicit surfaces*, J. Comput. Phys., 174 (2001), pp. 759–780.

[3] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, *Fast global minimization of the active contour/snake model*, J. Math. Imaging Vision, 28 (2007), pp. 151–167.
[4] P. Burchard, L.-T. Cheng, B. Merriman, and S. Osher, *Motion of curves in three spatial dimensions using a level set approach*, J. Comput. Phys., 170 (2001), pp. 720–741.
[5] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, Int. J. Comput. Vis., 22 (1997), pp. 61–79.
[6] T.F. Chan, S. Esedoḡlu, and M. Nikolova, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math., 66 (2006), pp. 1632–1648.
[7] W. Chen, C.S. Chou, and C.Y. Kao, *Lax–Friedrichs fast sweeping methods for steady state problems for hyperbolic conservation laws*, J. Comput. Phys., 234 (2013), pp. 452–471.
[8] L.-T. Cheng, *Efficient level set methods for constructing wavefronts in three spatial dimensions*, J. Comput. Phys., 226 (2007), pp. 2250–2270.
[9] L.-T. Cheng, P. Burchard, B. Merriman, and S. Osher, *Motion of curves constrained on surfaces using a level-set approach*, J. Comput. Phys., 175 (2002), pp. 604–644.
[10] H. Edelsbrunner, *Shapre reconstruction with Delaunay complex*, in Proceedings of LATIN '98: Theoretical Informatics, Vol. 1380, Springer-Verlag, New York, 1998.
[11] P.T. Fletcher and S. Joshi, *Riemannian Geometry for the Statistical Analysis of Diffusion Tensor Data*, Signal Processing, 87 (2007), pp. 250–262.
[12] P.T. Fletcher, C. Lu, S.M. Pizer, and S. Joshi, *Principal geodesic analysis for the study of nonlinear statistics of shape*, IEEE Trans. Med. Imag., 23 (2004), pp. 995–1005.
[13] A. Goh and R. Vidal, *Clustering and dimensionality reduction on Riemannian manifolds*, in Proceedings of CVPR, 2008.
[14] C.R. Goodall, *Procrustes methods in the statistical analysis of shape*, J. Roy. Statist. Soc. Ser. B, 53 (1991), pp. 285–339.
[15] T. Hastie and W. Stuetzle, *Principal curves*, J. Amer. Statist. Assoc., 84 (1989), pp. 502–516.
[16] S. Hauberg, *Principal curves on Riemannian manifolds*, IEEE Trans. Pattern Anal. Mach. Intell., 38 (2016), pp. 1915–1921.
[17] S. Huckemann and H. Ziezold, *Principal component analysis for Riemannian manifolds, with an application to triangular shape spaces*, Adv. Appl. Probab., 38 (2006), pp. 299–319.
[18] G.S. Jiang and D. Peng, *Weighted ENO schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2126–2143.
[19] S. Jung, I.L. Dryden, and J.S. Marron, *Analysis of principal nested spheres*, Biometrika, 99 (2012), pp. 551–568.
[20] C.Y. Kao, S.J. Osher, and J. Qian, *Lax–Friedrichs sweeping schemes for static Hamilton–Jacobi equations*, J. Comput. Phys., 196 (2004), pp. 367–391.
[21] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, Int. J. Comput. Vis., 1 (1998), pp. 321–331.
[22] K. Kenobi, I.L. Dryden, and H. Le, *Shape curves and geodesic modelling*, Biometrika, 97 (2010), pp. 567–584.
[23] A. Kume, I.L. Dryden, and H. Le, *Shape-space smoothing splines for planar landmark data*, Biometrika, 94 (2007), pp. 513–528.
[24] S.-M. Lee, A.L. Abbott, and P.A. Araman, *Dimensionality reduction and clustering on statistical manifolds*, in Proceedings of CVPR, 2007.
[25] S. Leung and S. Osher, *Fast global minimization of the active contour model with TV-inpainting and two-phase denoising*, in Proceedings of the 3rd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision, 2005, pp. 149–160.
[26] S. Leung and J. Qian, *An adjoint state method for 3D transmission traveltime tomography using first arrival*, Commun. Math. Sci., 4 (2006), pp. 249–266.
[27] W.B. Li and S. Leung, *A fast local level set adjoint state method for first arrival transmission traveltime tomography with discontinuous slowness*, Geophys. J. Int., 195 (2013), pp. 582–596.
[28] W.B. Li, S. Leung, and J. Qian, *A level-set adjoint-state method for crosswell transmission-reflection traveltime tomography*, Geophys. J. Int., 199 (2014), pp. 348–367.
[29] J. Liang, F. Park, and H. Zhao, *Robust and efficient implicit surface reconstruction for point clouds based on convexified image segmentation*, J. Sci. Comput., 54 (2013), pp. 577–602.
[30] T. Lin, H. Zha, and S.U. Lee, *Riemannian manifold learning for nonlinear dimensionality reduction*, in Proceedings of ECCV, 2006.
[31] X.D. Liu, S.J. Osher, and T. Chan, *Weighted essentially nonoscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.

[32] F. Memoli and G. Sapiro, *Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces*, J. Comput. Phys., 173 (2001), pp. 730–764.
[33] C. Min, *Local level set methods in high dimension and codimension*, J. Comput. Phys., 200 (2004), pp. 368–382.
[34] S.J. Osher and R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2003.
[35] S.J. Osher and J.A. Sethian, *Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.
[36] V.M. Panaretoes, T. Pham, and Z. Yao, *Principal flows*, J. Amer. Statist. Assoc., 109 (2014), pp. 424–436.
[37] D. Peng, B. Merriman, S. Osher, H.K. Zhao, and M. Kang, *A PDE-based fast local level set method*, J. Comput. Phys., 155 (1999), pp. 410–438.
[38] J.A. Sethian, *Level Set Methods*, 2nd ed., Cambridge University Press, Cambridge, 1999.
[39] C.W. Shu and S.J. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[40] P. Smereka, *Spiral crystal growth*, Phys. D, 138 (2000), pp. 282–301.
[41] J. Su, I.L. Dryden, E. Klassen, H. Le, and A. Srivastava, *Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds*, Image Vis. Comput., 30 (2012), pp. 428–442.
[42] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, *Geometric surface processing via anisotropic diffusion of normals*, in Proceedings of IEEE Visualization, 2002, pp. 125–132.
[43] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, *Geometric surface processing via normal maps*, ACM Trans. Graphics, 22 (2003), pp. 1012–1033.
[44] L.J.P. Van der Maaten, E.O. Postma, and H.J. Van den Herik, *Dimensionality reduction: A comparative review*, J. Mach. Learn. Res., 10 (2009), pp. 66–71.
[45] T. Wong and S. Leung, *A fast sweeping method for eikonal equations on implicit surfaces*, J. Sci. Comput., 67 (2016), pp. 837–859.
[46] Z. Yao and T. Pham, *Principal Sub-Manifolds*, manuscript, 2016.
[47] H.K. Zhao, *Fast sweeping method for eikonal equations*, Math. Comp., 74 (2005), pp. 603–627.
[48] H.-K. Zhao, T. Chan, B. Merriman, and S.J. Osher, *A variational level set approach for multiphase motion*, J. Comput. Phys., 127 (1996), pp. 179–195.
[49] H.-K. Zhao, S. Osher, and R. Fedkiw, *Fast surface reconstruction using the level set method*, in Proceedings of the IEEE Workshop on Variational and Level Set Methods, 2001, pp. 194–202.
[50] H.K. Zhao, S. Osher, B. Merriman, and M. Kang, *Implicit and non-parametric shape reconstruction from unorganized points using variational level set method*, Comput. Vis. Image Underst., 80 (2000), pp. 295–319.