SOLVING PARTIAL DIFFERENTIAL EQUATIONS ON MANIFOLDS FROM INCOMPLETE INTER-POINT DISTANCE

RONGJIE LAI * AND JIA LI †

Abstract. Solutions of partial differential equations (PDEs) on manifolds have provided important applications in different fields in science and engineering. Existing methods are majorly based on discretization of manifolds as implicit functions, triangle meshes, or point clouds, where the manifold structure is approximated by either zero level set of an implicit function or a set of points. In many applications, manifolds might be only provided as an inter-point distance matrix with possible missing values. This paper discusses a framework to discretize PDEs on manifolds represented as incomplete inter-point distance information. Without conducting a time-consuming global coordinates reconstruction, we propose a more efficient strategy by discretizing differential operators only based on point-wisely local reconstruction. Our local reconstruction model is based on the recent advances of low-rank matrix completion theory, where only a very small random portion of distance information is required. This method enables us to conduct analyses of incomplete distance data using solutions of special designed PDEs such as the Laplace-Beltrami (LB) eigen-system. As an application, we demonstrate a new way of manifold reconstruction from an incomplete distance by stitching patches using the spectrum of the LB operator. Intensive numerical experiments demonstrate the effectiveness of the proposed methods.

Key words. Manifolds, Laplace-Beltrami eigenproblem, Eikonal equation, Low-rank matrix completion.

AMS subject classifications. 65D18, 65D25, 65N25

1. Introduction. With the rapid development of advanced data acquisition technology, processing and analyzing data sampled on 3D shapes or even higher dimensional geometric objects becomes ubiquitous tasks such as those used in a 3D camera, medical imaging, protein structuring, social network analysis and many others [21, 2, 58, 50, 17, 6, 44, 22, 9]. PDE and variational PDE based methods have made great success to handle problems in signal and image processing which can be viewed as data on Euclidean domains. It is natural to consider PDE based methods to analyze and process signals on a general manifold and to understand geometric structures hidden in the data. Besides the classical implicit methods [38, 8, 7], finite difference methods [39, 52, 36, 62], finite element methods [41, 27, 20] and parameterization methods [49, 48, 57, 35] for solving differential equations on surfaces in $\mathbb{R}^3$, there has been increasing interests of solving PDEs on general d-dimensional manifold in $\mathbb{R}^p$ and their applications to data analysis. For instance, a diffusion geometry framework is developed to investigate the geometric structure of data based on solving Laplace-Beltrami (LB) eigenproblem using integral kernel methods [3, 4, 16]. More recently, a moving least square method and a local mesh method are considered to intrinsically solve different types of PDEs on manifolds represented as point clouds and its applications to the geometric understanding of point cloud data [29, 33, 34, 31].

All the aforementioned methods of solving PDEs on a general manifold $\mathcal{M} \subset \mathbb{R}^p$ are commonly considered $\mathcal{M}$ is sampled as a set of points $\{\boldsymbol{x}_i \in \mathbb{R}^p\}_{i=1}^n$, referred as a *point cloud*, and discretization of differential operators or approximation of integral equations are relied on available coordinates information of $\{\boldsymbol{x}_i\}_{i=1}^n$. However, there are many applications whose input information has no point coordinates but only an incomplete inter-point distance $(d(\boldsymbol{x}_i, \boldsymbol{x}_j))$. Examples include incomplete distance

information from sensor network localization [22, 9], protein structuring from NMR spectroscopy [17, 6] and global position from local distance configuration of cities [47]. A well-known distance geometry problem [17, 37] is to find the global configuration of data based on the incomplete distance information. The objective in this paper is different from the canonical distance geometry problem. We would like to develop numerical methods for solving PDEs on manifolds represented as incomplete inter-point distance. One natural way to approach this problem is first to apply a global reconstruction algorithm to obtain a point cloud representation of the input manifold, then established numerical methods for PDEs on point clouds can be directly applied. However, the global coordinate reconstruction might be very time consuming as it involves with semi-positive definite programming whose size depends on the number of points and could be very large in practice [9].

In this paper, we propose a different strategy to solve PDEs on manifolds represented as incomplete inter-point distance without conducting global coordinates reconstruction. Our idea is based on two intrinsic properties of differential operators on manifolds. Namely, the definition of a differential operator is only point-wisely depending on local information of the manifold and is invariant under different choices of local coordinates. This motivates us to only consider to conduct point-wisely local coordinate reconstruction for the associated local neighborhood, then we can point-wisely approximate differential operators based on the reconstructed local coordinates. After that, numerical solver for differential equations can be constructed.

Inspired by the classical multi-dimensional scaling [26], the full distance matrix is one-to-one corresponding to the Gram matrix which can be further used to determine coordinates by its eigen-decomposition. More importantly, the Gram matrix can be essentially viewed as an inner product matrix after certain coordinates shift. As long as the matrix size, namely the size of a given local neighborhood, is larger than the embedding dimension of the local neighborhood, then the corresponding Gram matrix has to be a low-rank matrix. This suggests us to use low rank as a prior knowledge to reconstruct the Gram matrix based on constraints of available distance information. Adapted from the recent advances of the low-rank matrix completion theory [13], we consider a nuclear norm regularized convex optimization problem to reconstruct local coordinates based on available distance information. Once local coordinates can be obtained, we apply the intrinsic methods of approximating differential operators on point clouds developed in [29, 33, 34] to discretize the desired differential equation. These intrinsic methods can be used to discretize different types of PDEs including parabolic, elliptic and hyperbolic PDEs. Our method can be viewed as natural extensions of these two methods to a new data structure where no coordinate information but only partial inter-point distance information of point clouds is provided. The outline of our strategy can be as summarized as follows:

Step 0 For the i-th point, chose its K-nearest neighborhood (KNN) $N(i)$ based on the given incomplete distance or from the prior information if its available.

Step 1 Applying the matrix completion method discussed in Section 2 to reconstruct local coordinates for KNN of the i-th point.

Step 2 Applying the MLS method or the local mesh method in [29, 33, 34] to approximate the desired differential operators at the i-th point. This provides the i-th row of the discretized matrix representation of the desired differential equation.

An immediate advantage of this approach is to save computation time by avoiding the global coordinate reconstruction, which can reduce the complexity quadratically to linearly scaling with the total number of points. More details about this

method will be discussed in Section 3 and will be verified in our numerical experiments. Furthermore, this approach enables us to conduct geometric understanding of data without global coordinates reconstruction. Examples include global pattern extraction, comparison and classification as many existing methods conducted using results of differential equations [46, 30, 31]. As a byproduct of global information from PDEs, we also propose a new method for reconstructing manifolds by stitching its local patches. This new method is much more efficient than the way of direct global reconstruction using matrix completion method. Moreover, it can also overcome possible reconstruction failure using global matrix completion due to the coherent missing information.

The rest of this paper is organized as follows. In section 2, we propose a low-rank matrix completion model to reconstruct local coordinates and design an algorithm to solve the proposed convex optimization problem based on operator splitting and the alternating direction method. After that, section 3 is devoted to discuss point-wisely approximating differential operators based on the intrinsic methods proposed in [29, 33, 34]. We also discuss our two model PDEs, the Laplace-Beltrami (LB) eigenvalue problem and the Eiknoal equation for solve geodesic distance on manifolds. As an application of global information using solutions of LB eigenvalue problem, we propose a new method for reconstructing manifolds by stitching local patches in the LB frequency space in section 4. All numerical experiments about the effectiveness of our local coordinates reconstruction model, the accuracy and robustness of the PDEs solvers, and demonstration of manifold reconstruction are discussed in section 5. Finally, we conclude our work in section 6.

2. Local coordinates reconstruction via matrix completion. In this section, we first review the concept of the classical multidimensional scaling (MDS). Inspired by the classical MDS, we propose a method of reconstructing local coordinates from the given incomplete inter-point distance matrix using the low-rank matrix completion [13]. A numerical algorithm based on the augmented Lagrangian is also designed to solve the proposed convex optimization problem. As we described before, our strategy is to conduct local coordinates reconstruction for the KNN $N(i)$ of the i-th point. For convenience, we write $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_\ell\}$ as ℓ points for $N(i)$ as the following model does not rely on a typical choice of $N(i)$.

2.1. Classical multidimensional scaling. The classical MDS is to find a configuration of a set of points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_\ell\}$ in $\mathbb{R}^p$ from the given distance matrix $D = (d_{ij}^2)_{\ell \times \ell}$ [1] such that the Euclidean distance matrix $\left(\|\boldsymbol{x_i} - \boldsymbol{x_j}\|_2^2 \right)_{ij}$ among these points is as close as possible to the given distance matrix D [26, 10].

DEFINITION 1. *A distance matrix $D = (D_{ij})_{\ell \times \ell}$ is called Euclidean if there exist some points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\ell \in \mathbb{R}^p$ such that $D_{ij} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)$. Let's write $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_\ell)^\top$ as an $\ell \times p$ matrix and denote the centering matrix by $H = I_l - \frac{1}{l}\boldsymbol{1}\boldsymbol{1}^\top$. A Gram matrix $B = (B_{ij})$ is defined as $B = -\frac{1}{2}HDH$, equivalently, $D_{ij} = B_{ii} + B_{jj} - B_{ij} - B_{ji}$.*

Note that the Euclidean distance matrix is translation invariant, thus, the matrix H essentially translates the point set centering at the origin. It is well-known that D is Euclidean if and only if the Gram matrix B is positive semidefinite. Furthermore, since $B = HXX^\top H$, it is clear that $\text{rank}(B) = \text{rank}(XX^\top) = \text{rank}(X) = p$. In our case, we consider the number of points is much larger the dimension of points. This

---

[1]For convenience, each entry of D is specified by the square of distance.

suggests the low-rank property of B as its prior knowledge. If D is Euclidean, and thus B is positive semidefinite, then the eigen-decomposition of $B = U\Lambda U^\top$ provides the coordinate reconstruction $X = U_p \Lambda_p^{1/2}$, where Λ_p is a diagonal matrix formed by the largest p eigenvalues of X and U_p are the corresponding eigenfunctions.

2.2. Local coordinates reconstruction via positive-semidefinite matrix completion. Different from the MDS described as the above, one of the major challenges in our assumption is that only a small portion of D is available. Generally speaking, it is impossible to recover the whole matrix only from its small portion of information. We here assume the information D is randomly missing. Different models of semi-definite programming have been considered [1, 9] in this scenario. Here, we consider a different model inspired by the low-rank prior information of the Gram matrix and the recent advances from the low-rank matrix completion theory [13]. In other words, we are seeking for a symmetric positive-semidefinite low-rank matrix B satisfying $B_{ii} + B_{jj} - B_{ij} - B_{ji} = D_{ij}$ for (i, j) in the available portion. Mathematically, let's write $\Omega \subset \{(i, j) \mid 1 \le j < i < \ell\}$ as the available index set and write $\mathbb{S} = \{B = (b_{ij}) \in \mathbb{R}^{\ell \times \ell} \mid B = B^T\}$. We further write the available part of D as a vector $D_\Omega \in \mathbb{R}^{|\Omega|}$ and define a restriction operator $\mathcal{R}_\Omega : \mathbb{S} \to \mathbb{R}^{|\Omega|}$ by $\mathcal{R}_\Omega(B)_{ij} = B_{ii} + B_{jj} - B_{ij} - B_{ji}$ with $(i, j) \in \Omega$. Our problem becomes to find a positive semidefinite matrix $B \in \mathbb{S}$ satisfying $\mathcal{R}_\Omega(B) = D_\Omega$. It is clear that this problem is underdetermined and there are infinitely many solutions, most of which are far from the ground truth of B and thus not applicable. Regularization method is a common way to handle this type of situation. Based on the prior knowledge of B is a low rank matrix without knowing the exact rank p, we here consider the following optimization problem.

$$(1) \qquad \min_{B \in \mathbb{S}} \operatorname{Rank}(B), \quad \text{s.t.} \quad B \succeq 0 \quad \& \quad \mathcal{R}_\Omega(B) = D_\Omega$$

Inspired by the matrix completion theory [13], we relax the above NP-hard problem by replacing $\operatorname{Rank}(B)$ as the nuclear norm $\|B\|_*$. In addition, the input distance matrix is invariant under translation of the reconstructed coordinates $\{\boldsymbol{x}_i\}$. This translation ambiguity can be fixed by requiring $\sum_i \boldsymbol{x}_i = \boldsymbol{0}$. This condition is equivalent to row sum of B is zero, which can also be obtained from the definition of the Gram matrix B. Therefore, we propose the following matrix completion model:

$$(2) \qquad \min_{B \in \mathbb{S}} \|B\|_*, \quad \text{s.t.} \quad B \succeq 0, \quad \mathcal{R}_\Omega(B) = D_\Omega \quad \& \quad B\boldsymbol{1} = 0$$

where $\boldsymbol{1}$ denotes a column vector with all elements constant 1. Once the above Gram matrix is obtained, coordinates can be reconstructed by the eigen-decomposition of B as the method used in the classical MDS described in section 2.1. Simultaneously, the eigen-decomposition of the reconstructed B clearly shows the exact dimension p by counting the cardinality of non-zero singular value. We would like to remark that the constraints and the restriction operator are different from the setting in [13] and it is highly nontrivial to check the restricted isometry property [40], therefore, theoretical analysis conducted in [13, 40] can not be directly applied in our problem. By considering a dual basis method, we can also theoretically show the exact completion can be achieved under certain coherence condition. More details about the theoretical analysis of the model will appear in our ongoing work [28]. Additionally, in a related work [59], the authors propose a semidefinite embedding (SDE) model as a kernel learning method for nonlinear dimension reduction, where no random missing but only KNN information of D is considered, and the rank minimization is not

considered but maximization of $Trace(B)$ is proposed for maximizing the variance in feature space.

2.3. Numerical Algorithm. For a convenience of designing a numerical algorithm of model (2), we first introduce some notations. Note that the set of symmetric matrix $\mathbb{S}$ can be viewed as $\mathbb{R}^{\ell(\ell+1)/2}$ due to a natural isomorphism $\zeta : \mathbb{S} \to \mathbb{R}^{\ell(\ell+1)/2}, B \mapsto \hat{B} = (b_{11}, \cdots, b_{\ell 1}, b_{22}, \cdots, b_{\ell 2}, \cdots, b_{\ell \ell})^T$. We denote $\iota = \zeta^{-1}$ and further merge the two linear constraints by defining the following linear operator:

$$(3) \qquad \mathcal{A} : \mathbb{R}^{l(l+1)/2} \to \mathbb{R}^{\Omega} \times \mathbb{R}^{\ell}, \qquad \hat{B} \mapsto \left(\mathcal{R}_{\Omega} \circ \iota(\hat{B}), \iota(\hat{B})\mathbf{1} \right)$$

and extend the vector D_{Ω} as $\tilde{D}_{\Omega} = (D_{\Omega}, \vec{0})$. Note that $\|B\|_*$ is the same as $\mathrm{Trace}(B)$ as B is positive semi-definite. Therefore, the proposed model (2) can be written as

$$(4) \qquad \min_{\hat{B} \in \mathbb{R}^{k(k+1)/2}} \mathrm{Trace}(\iota(\hat{B})), \quad \text{s.t.} \quad \mathcal{A}\hat{B} = \tilde{D}_{\Omega}, \quad \iota(\hat{B}) \succeq 0$$

The model (4) is a semi-definite programming (SDP) problem. It is well-known that the SDP problem can be solved by interior-points method [53, 56] as a canonical choice. However, when the number of constraint is with the order $\mathcal{O}(\ell^2)$ as in (4), the computational time can be with the complexity $\mathcal{O}(\ell^6)$ [60]. Thus, we use the alternating direction method [60] to save both the time and memory consumption. We first introduce an auxiliary variable $\hat{C} = \hat{B}$ and write (4) as:

$$(5) \qquad \min_{\hat{B}, \hat{C} \in \mathbb{R}^{n(n+1)/2}} \mathrm{Trace}(\iota(\hat{B})), \quad \text{s.t.} \quad \mathcal{A}\hat{B} = \tilde{D}_{\Omega}, \quad \iota(\hat{C}) \succeq 0, \quad \hat{B} = \hat{C}$$

Then we introduce an augmented Lagrangian of model (5) as follows:

$$(6) \ \mathcal{L}_{\mu_1, \mu_2}(\hat{B}, \hat{C}; H_1, H_2) = \mathrm{Trace}(\iota(\hat{B})) + \frac{\mu_1}{2}\|\mathcal{A}\hat{B} - \tilde{D}_{\Omega} + H_1\|_2^2 + \frac{\mu_2}{2}\|\hat{B} - \hat{C} + H_2\|_2^2.$$

where H_1 and H_2 are two dual variables. The variable splitting enables us to alternatively optimize $\hat{B}$ and $\hat{C}$. As (5) is convex, the saddle point of $\mathcal{L}_{\mu_1, \mu_2}(\hat{B}, \hat{C}; H_1, H_2)$ is the solution which can be obtained by the following iterative procedure:

$$(7) \quad \begin{cases} \hat{B}^{k+1} = \arg\min_{\hat{B}} \mathrm{Trace}(\iota(\hat{B})) + \frac{\mu_1}{2}\|\mathcal{A}\hat{B} - \tilde{D}_{\Omega} + H_1\|_2^2 + \frac{\mu_2}{2}\|\hat{B} - \hat{C} + H_2\|_2^2 \\ \hat{C}^{k+1} = \arg\min_{\hat{C}} \frac{\mu_2}{2}\|\hat{B}^{k+1} - \hat{C} + H_2^k\|^2, \quad \iota(\hat{C}) \succeq 0, \\ H_1^{k+1} = H_1^k + (\mathcal{A}\hat{B} - \tilde{D}_{\Omega}), \\ H_2^{k+1} = H_2^k + \hat{B} - \hat{C}, \end{cases}$$

Note that the model (5) is convex, therefore convergence of the above method can be guaranteed [54, 60].

The first subproblem is convex and differentiable. From its first order optimality condition, $\hat{B}^{k+1}$ solves the following linear system:

$$(8) \qquad (\mu_1 \mathcal{A}^* \mathcal{A} + \mu_2)\hat{B}^{k+1} = \mu_1 \mathcal{A}^*(\tilde{D}_{\Omega} - H_1^k) + \mu_2(\hat{C}^k - H_2^k) - \zeta(I_n).$$

where $\mathcal{A}^*$ is the adjoint operator of $\mathcal{A}$ defined as follows. It is clear that for any given $\tilde{U} = (U_{\Omega}, \vec{q}) \in \mathbb{R}^{\Omega} \times \mathbb{R}^{\ell}$ with $U_{\Omega} = \{u_{ij}\}_{(i,j) \in \Omega}$, the conjugate of $\mathcal{A}$ is given by $\mathcal{A}^* \tilde{U} = \zeta(U) + \zeta(Q)$. Here we define symmetric matrices U and Q by:

$$(9) \qquad U_{ij} = \begin{cases} \displaystyle\sum_{(i,k) \in \Omega} u_{ik} + \sum_{(k,i) \in \Omega} u_{ki}, & \text{if } i = j \\ -2u_{ij}, & \text{if } i > j \end{cases} \quad \text{and} \quad Q_{ij} = \begin{cases} q_i, & \text{if } i = j \\ q_i + q_j, & \text{if } i > j \end{cases}$$

and $U_{ij} = 0$ for any $(i,j) \notin \Omega$. As $\mu_1 \mathcal{A}^* \mathcal{A} + \mu_2 I_\ell$ is a symmetric and positive definite matrix, there are numerous numerical solvers to approach the solution of the above problem. Here, we typically chose the conjugate gradient method to solve (8).

The second subproblem of (7) is to find the nearest point of $\hat{B}^{k+1} + H_2^k$ in the positive semi-definite cone. Its closed-form solution can be simply realized by truncating all the information in the negative eigenvalue part. More precisely, for a symmetric matrix X with eigen-decomposition $X = V\Lambda V^\top$, we define an eigenvalue hard thresholding (EVHT) operator as follows:

$$(10) \qquad \mathcal{T}_E(X) = V\Lambda_+ V^\top, \quad \Lambda_+ = \text{diag}(\max\{\Lambda(i,i), 0\}_{i=1}^n).$$

With this definition, the second step of (7) can be calculated as follows:

$$(11) \qquad \hat{C}^{k+1} = \zeta \circ \mathcal{T}_E \circ \iota(\hat{B}^{k+1} + H_2^k).$$

We summarize the above iterative method in Algorithm 1. After finding $\hat{B}$ by Algorithm 1, eigen-decomposition of $\iota(\hat{B})$ can ultimately reconstruct the coordinates of points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_\ell\}$ from the incomplete distance D_Ω.

Algorithm 1 Augmented Lagrangian method to solve (4)

Initialization. Set $C^0 = 0, H_1^0 = 0, H_2^0 = 0, E(0) = 2e+10, E(1) = 1e+10, \epsilon = 1e-6$. Set $\mu_1 = 10, \mu_2 = 5$.
while ($|E(k) - E(k-1)|/E(k) \geq \epsilon$) **do**
    1. Solve $(\mu_1 \mathcal{A}^* \mathcal{A} + \mu_2)\hat{B}^{k+1} = \mu_1 \mathcal{A}^*(\tilde{D}_\Omega - H_1^k) + \mu_2(\hat{C}^k - H_2^k) - \zeta(I_n)$.
    2. $\hat{C}^{k+1} = \zeta \circ \mathcal{T}_E \circ \iota(\hat{B}^{k+1} + H_2^k)$.
    3. $H_1^{k+1} = H_1^k + \mathcal{A}\hat{B}^{k+1} - \tilde{D}_\Omega$.
    4. $H_2^{k+1} = H_2^k + \hat{B}^{k+1} - \hat{C}^{k+1}$.
    5. Let $E(k+1) = \text{Trace}(\iota(\hat{B}^{k+1})) + \frac{\mu_1}{2}\|\mathcal{A}\hat{B}^{k+1} - \tilde{D}_\Omega\|_2^2 + \frac{\mu_2}{2}\|\hat{B}^{k+1} - \hat{C}^{k+1}\|_2^2$.
end while

Remark 1. *In Algorithm 1, Step 2 is usually time consuming as the full eigen-decomposition has time complexity $\mathcal{O}(\ell^3)$. Because ultimately the solution of B and C should be low rank, it is safe to only carry out the largest m eigenfunctions by setting m is relatively big enough comparing with the rank of B but also small in terms of the matrix size. For instance, in our later numerical experiments for 3D point clouds, we use the MATLAB function "eigs" to find the largest 20 eigenvalues and truncate all the other parts. In this way, the time consumption of Step 2 can be apparently decreased, and numerical results suggest no difference between this simplification and calculating full eigen-decomposition.*

Remark 2. *In addition, we would like to remark that the distance matrix is also a low rank matrix provided by $Rank(D) \leq Rank(B) + 2$ due to the relationship between B and D satisfying $D = -2B + diag(B)\mathbf{1}^\top + \mathbf{1}\,diag(B)^\top$, where $diag: \mathbb{R}^{\ell \times \ell} \to \mathbb{R}^{\ell \times 1}$ is the linear operator to project diagonal component of B to a column vector. Therefore, we can also directly reconstruct D based on its available information. This leads to a low rank minimization model as follows:*

$$(12) \qquad \min_{\tilde{D} \in \mathbb{S}} \|\tilde{D}\|_* \quad s.t. \quad \tilde{D}_{ij} = D_{ij}, \quad (i,j) \in \Omega$$

The above model is exactly the same as the general matrix completion theory discussed in [13], which also indicates that the minimum desired information of D for successful

reconstruction is proportion to the matrix rank. Note that the rank of D is higher than the rank of matrix B. Plus, the less flexibility of B due to the positive semi-definite condition can also help to reduce the requirement of valid distance information for successful reconstruction. Therefore, we would expect that model (12) *requires a higher rate of valid distance than the proposed model* (2). *In fact, our numerical experiments in Section* 5 *also verify this observation.*

3. Solving PDEs on Manifolds from Incomplete Distance. In this section, after a brief review of the moving least square (MLS) method [33, 34] and the local mesh (LM) method [29] for solving PDEs on manifolds represented as point clouds, we demonstrate our proposed methods of solving an elliptic eigenvalue problem and a nonlinear hyperbolic equation on manifolds represented as incomplete inter-point distance information.

3.1. Approximating differential operators from reconstructed local coordinates. For traditional implicit or triangle mesh representation of surfaces in $\mathbb{R}^3$, implicit methods, parameterization methods, finite difference methods and finite element methods [38, 39, 8, 52, 7, 36, 49, 48, 57, 35, 11, 62, 41, 27, 20] have been proposed to solve PDEs on surfaces. However, it is not straightforward to use these methods in our case as the incomplete distance data structure has no global mesh structure and the ground manifold could be with dimension higher than two and in a high co-dimensional ambient space. As the definitions of differential operators on manifolds are coordinate invariant, we can approximate differential operators point-wisely based on its local coordinate reconstruction obtained from the matrix completion method proposed in Section 2. Namely, each given point, its reconstructed local neighborhood can be viewed as a point cloud, therefore, we apply moving least square (MSL) method [33, 34] or local mesh (LM) method [29] to approximate the differential operators. These methods can achieve high order accuracy and enjoy more flexibility. They can be applied to manifolds with arbitrary dimensions and codimensions. To make the paper self-sufficient, we briefly discuss these two methods in this section.

Denote the index set of the KNN of the i-th point as $N(i)$ and write $\mathcal{X}(i) = \{\boldsymbol{x}_k \in \mathbb{R}^p \mid k \in N(i)\}$ as the reconstructed Euclidean coordinates of the neighborhood $N(i)$. Based on $\mathcal{X}(i)$, its tangent space and normal space can be determined by standard principle component analysis (PCA) [23], which is provided by eigensystem of the covariance matrix P_i of $N(i)$ defined on:

$$(13) \qquad P_i = \sum_{k \in N(i)} (\boldsymbol{x}_k - c_i)^T (\boldsymbol{x}_k - c_i).$$

where $c_i = \frac{1}{|N(i)|} \sum_{k \in N(i)} \boldsymbol{x}_k$ is the centroid of $\mathcal{X}(i)$. If the intrinsic dimension of the manifold is d, then the jump of P_i's eigenvalues guides the splitting $\mathbb{R}^p = \mathcal{T}_i \oplus \mathcal{N}_i$. Here $\mathcal{T}_i$ represents the tangent space spanned by $\{e_i^1, \cdots, e_i^d\}$ corresponding to the d largest eigenvectors of P_i, and $\mathcal{N}_i$ represents the normal space spanned by the rest of the eigenvectors of P_i. As $\{e_i^1, \cdots, e_i^p\}$ forms an orthonormal basis near the point $\boldsymbol{x}_i$, a new *virtual coordinates* of $\boldsymbol{x}_k$ can be obtained by $(\boldsymbol{u}_k, \boldsymbol{v}_k) = (\{\langle \boldsymbol{x}_k, e_i^\alpha \rangle\}_{\alpha=1}^d, \{\langle \boldsymbol{x}_k, e_i^\beta \rangle\}_{\beta=d+1}^p)$. Therefore, the manifold structure near $\boldsymbol{x}_i$ can be approximated by a degree 2 polynomial map $\boldsymbol{Z}_i : \mathbb{R}^d \to \mathbb{R}^{p-d}, \boldsymbol{Z}_i(\boldsymbol{u}) = (\sum_{|\boldsymbol{\xi}| \le 2} \boldsymbol{c}_{\boldsymbol{\xi}}^1 \boldsymbol{u}^{\boldsymbol{\xi}}, \cdots, \sum_{|\boldsymbol{\xi}| \le 2} \boldsymbol{c}_{\boldsymbol{\xi}}^{p-d} \boldsymbol{u}^{\boldsymbol{\xi}})$ whose coefficients

are given by the following moving least square problem:

$$(14) \qquad \min_{\boldsymbol{c}_{\boldsymbol{\xi}}^1, \cdots, \boldsymbol{c}_{\boldsymbol{\xi}}^{p-d}} \sum_{k \in N(i)} w(\|\boldsymbol{x}_k - \boldsymbol{x}_i\|) \|\boldsymbol{Z}_i(\boldsymbol{u}_k) - \boldsymbol{v}_k\|^2,$$

where the standard multi-index notation is used for $\boldsymbol{u^\xi} = u_1^{\xi_1} \cdots u_d^{\xi_d}$ and $w(\cdot)$ is a weight function whose typical choice can be $w(d) = \exp(-\frac{d^2}{h^2})$ and $h = \max_{k \in N(i)} \|\boldsymbol{x}_k - \boldsymbol{x}_i\|$. Similarly, any function f or vector filed $\boldsymbol{V}$ defined on $\mathcal{X}(i)$ can also be interpolated as a degree 2 polynomial $f(\boldsymbol{u})$ or $\boldsymbol{V}(\boldsymbol{u})$. Overall, we obtain a local polynomial interpolation of $\mathcal{X}(i)$ as $\boldsymbol{Z}_i(\boldsymbol{u})$ and a local polynomial interpolation of function defined on $\mathcal{X}(i)$ as $f(\boldsymbol{u})$. Therefore, the metric tensor $G = (g_{st})$ near $\boldsymbol{x}_i$ can be obtained by $g_{st} = \delta_{st} + \langle \frac{\partial \boldsymbol{Z}_i}{\partial u_s}, \frac{\partial \boldsymbol{Z}_i}{\partial u_t} \rangle$. This enables us to approximate intrinsic differential operators at $\boldsymbol{x}_i$ such as:

$$\nabla_{\mathcal{M}} f = g^{st} \frac{\partial f}{\partial u_t} \partial_{u_s}, \quad \text{div}_{\mathcal{M}} \boldsymbol{V} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u_s} \left(\sqrt{g} \, V^s \right), \quad \Delta_{\mathcal{M}} f = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u_s} \left(\sqrt{g} g^{st} \frac{\partial f}{\partial u_t} \right).$$

where $g = det(G)$, $(g^{st}) = G^{-1}$ and the Einstein summation is used. Other differential operators can also be approximated using the similar method. After that, a finite difference type of method can be applied to solve differential equations on the given data. We refer [33, 34] for more details about this approach.

The local mesh method proposed in [29] is another way to approximate the above differential operators from the virtual local coordinates. Namely, at i-th point, a local connectivity can be constructed for the projection image of $\mathcal{X}(i)$ on $\mathcal{T}_i$ through the standard Delaunay triangulation. Thus, this connectivity can be directly inherited on $\mathcal{X}(i)$. If we write the simplex of the first ring of $\boldsymbol{x}_i$ by $\mathcal{R}(i)$, the differential operators at each point can be approximated by weighted average as follows:

$$\nabla_M f(\boldsymbol{x}_i) = \frac{\sum_{S \in \mathcal{R}(i)} |S| \nabla_S f(\boldsymbol{x}_i)}{\sum_{S \in \mathcal{R}(i)} |S|}, \quad \text{div}_M \overrightarrow{V}(p_i) = \frac{\sum_{S \in \mathcal{R}(i)} |S| \text{div}_S \overrightarrow{V}(\boldsymbol{x}_i)}{\sum_{S \in \mathcal{R}(i)} |S|}$$

Moreover, a finite element type of method can be applied to estimate the mass matrix and stiffness matrix. Therefore, differential equations like Laplace-Beltrami eigenvalues problems can be solved. We refer [29] for more detailed discussion about construction of mass matrix and stiffness matrix.

We remark that the above constructions are conducted point-wisely. As long as a differential operator is well-defined on a manifold, namely, its definition does not depend on the choice of local coordinate, then the above procedure can consistently approximate the desired differential operators. A different strategy that can also be considered is to reconstruct coordinates first using the proposed Gram matrix completion algorithm, then approximate differential operators based on the global reconstructed coordinates. We would like to point out that the proposed methods of approximating differential operators based on local reconstruction enjoys advantages of computation efficiency and memory consumption. In fact, the most time-consuming part of the coordinate reconstruction Algorithm 1 is the eigenvalue hard thresholding. Consider a incomplete distance data with n nodes, the complexity of each eigenvalue thresholding step is $\mathscr{O}(n^2 m)$ for global reconstruction if only the largest m eigenvalues are computed. In our local reconstruction strategy, we only reconstruct coordinates of ℓ nearest points near $\boldsymbol{x}_i$ with complexity $\mathscr{O}(n\ell^2 m)$. When n is very large (such as 16002 used in our experiments) and ℓ is small (≤ 30 in our experiments), the

local reconstruction strategy is apparently much more efficient than global reconstruction strategy. Our numerical experiments discussed in Section 5 also support this point (see Table 2). Moreover, the local reconstruction method supports parallel computation which can further reduce time consumption. Additionally, the memory consumption of global reconstruction is $\mathcal{O}(n^2)$ and the matrix is not sparse, which often crashes the program due to insufficient memory for large size data. However, our local reconstruction strategy only requires the memory $\mathcal{O}(\ell^2)$ which avoids the problem of exceeding memory limit.

3.2. Solving differential equations based on incomplete distance. The first PDE we consider is an elliptic eigenvalue problem of the Laplace-Beltrami (LB) operator $\Delta_{\mathcal{M}}$. The LB operator is self-adjoint and elliptic, so its spectrum is discrete. We denote the eigenvalues of $-\Delta_{\mathcal{M}}$ as $0 = \lambda_0 < \lambda_1 < \lambda_2 < \cdots$ and the corresponding eigenfunctions as $\phi_0, \phi_1, \phi_2, \cdots$ satisfying the following equations [15]:

$$(15) \qquad \Delta_{\mathcal{M}}\phi_k = -\lambda_k \phi_k, \quad k = 0, 1, 2, \cdots.$$

The set of LB eigenfunctions $\{\phi_k\}_{k=0}^{\infty}$ forms an orthonormal basis of the space of L^2 functions on $\mathcal{M}$. The set $\{\lambda_k, \phi_k\}_{k=0}^{\infty}$ is called LB eigensystem of $\mathcal{M}$. Due to the intrinsic definition of the LB operator $\Delta_{\mathcal{M}}$, the induced LB eigensystem $\{\lambda_k, \phi_k\}_{k=0}^{\infty}$ is also completely intrinsic to the manifold geometry and provides an intrinsic and systematic characterization of the underlying manifold geometry [5]. Recently, there have been increasing interests in using the LB eigen-geometry for 3D shape analysis as well as point clouds analysis [41, 32, 55, 46, 24, 51, 30, 12, 33, 31]. Therefore, it would be also important to consider solve the LB eigensystem for manifolds represented as incomplete inter-point distance, then all existing methods of conducting data analysis using the LB eigensystem can also be adapted to the incomplete distance data structure. Our numerical solvers are based on the numerical solvers for on manifolds represented as point clouds. First, we apply the matrix completion model for point-wise local coordinate reconstruction. Either MLS or LM can be then applied to obtain the discretized matrix form of the equation, where the only step we need to conduct is to use the reconstructed local coordinates for the LB operator discretization as we discussed in the section 3.1. As the LB operator is invariant under different choices of the local coordinates, the rigid motion ambiguity from the local coordinate reconstruction will not introduce inconsistency across different coordinate patches. This guarantees that our method can provide satisfactory numerical results as we illustrate in the numerical section. We would like to remark that it is relatively straightforward to have the local consistency of the differential operator as long as the available distance information satisfying certain incoherence condition as we discussed in our theoretical validation work [28]. However, the stability of the discretization is still open and will be explored in our future work.

The second equation we consider is the Eikonal equation, a special type of nonlinear hyperbolic PDE on manifolds. This equation is used to characterize the geodesic distance, an intrinsic measurement between two points, on a manifold. The Eikonal equation for the distance map d to a given set Γ on $\mathcal{M}$ can be stated as follows:

$$(16) \qquad \begin{cases} |\nabla_{\mathcal{M}} d(x)| = 1 \\ d(x) = 0, \ x \in \Gamma \subset \mathcal{M} \end{cases}$$

when Γ only includes a single point p, the distance map can be denoted as d_p. In practice, provided points coordinates and mesh structure, model (16) can be solved

by fast marching and fast sweeping methods [45, 25, 63]. For manifolds represented as point clouds, a local mesh method based on fast marching is proposed in [29]. As long as the local coordinate reconstruction is obtained using the proposed matrix completion model, then a local Delaunay mesh structure can be constructed. After that, we repeat the local mesh method to conduct the fast marching as this approach only depends on the distance information of the first ring structure of the local mesh, which is again rigid motion invariant and will not be affected due to the rigid motion ambiguity from the local coordinate reconstruction.

4. Applications on Manifold Reconstruction and Dimension Reduction. As we mentioned before, solutions of differential equations on manifolds can provide global information for understanding data structure. Thus, PDEs can be viewed as "bridges" linking between local information and global information. This shares the same sprit as "think globally, fit locally" discussed in [43] although no PDEs are involved there. Therefore, without global reconstruction, some global analysis of point clouds such as pattern recognition, comparisons and classification can be further considered for data represented as incomplete distance. In this section, we illustrate results on applications of manifold reconstruction and dimension reduction based on solutions of the Laplace-Beltrami eigenvalue problem on incomplete distance data.

4.1. Manifold reconstruction from distance via patch stitching using LB eigenfunctions. The patch stitching problem is to reconstruct the coordinate of a point cloud $P \in \mathbb{R}^{n \times d}$ from coordinates of L subsets (referred as patches) $\{\Omega_j\}_{j=1}^{L}$, where Ω_j denotes the index set for patch j. In practice, for each patch Ω_j, the local coordinates $Q_j \in \mathbb{R}^{n_j \times d}$, which are provided as input information or are reconstructed from distance information, have the same pair-wise distance and geometry as the restricted global coordinate P_{Ω_j} with possible rigid motion difference. In other words, $P_{\Omega_j} = Q_j R_j + \mathbf{1}_j b_j$, where $R_j \in O(d) = \{R \in \mathbb{R}^{d \times d} \mid R^\top R = I_d\}$ is an orthogonal matrix for rotation and reflection, $\mathbf{1}_j \in \mathbb{R}^{n_j \times 1}$ is a vector with all element equal to 1, and $b_j \in \mathbb{R}^{1 \times d}$ is the translation vector. A straightforward approach to solve $\{R_j\}$ and $\{b_j\}$ is to minimize the quadratic loss:

$$(17) \qquad \min_{P, \{R_j \in O(d),\ b_j \in \mathbb{R}^d\}_{j=1}^{L}} \sum_{j=1}^{L} \|P_{\Omega_j} - Q_j R_j - \mathbf{1}_j b_j\|_2^2,$$

Such model is a non-convex problem since $\{R_j\}$ are required to optimize over the non-convex domain of orthogonal transforms. In [18], the authors proposed a three-stage patch stitching method from the local coordinate by synchronizing the reflection, rotation and translation successively. However, the quality of synchronization is strongly relies on the proportion of the overlapping index. More recently, a spectral relaxation method is proposed to solve the above problem in [14]. This method essentially relax the set of orthogonality constraints and perform well if neighborhood patches has reasonable enough overlapping points.

Here, we propose an alternative method to reconstruct the global coordinate from local coordinate, or merely incomplete distance information of each patch. Based on solution of LB eigen-problem incomplete distance information discussed in Section 3.2, we consider the LB eigen-system as a "bridge" to connect the information of different patches. This idea of this new method is essentially align each local patch in the spectral domain instead of the original Euclidean coordinate domain. As the global information has been captured from the LB eigen-system, the overlapping of different

patches is not directly required here. This approach is inspired by approximating coordinates using LB eigenfunctions. For a given point cloud $P \in \mathbb{R}^{n \times d}$, it can be approximated by the linear combination of the LB eigenfunctions as $P \approx \Phi \alpha$. Here, $\Phi = (\phi_1, \cdots, \phi_N) \in \mathbb{R}^{n \times N}$ represents the first N LB eigenfunctions $\{\phi_i\}_{i=1}^N$ and $\alpha = (\alpha_1, \cdots, \alpha_d) \in \mathbb{R}^{N \times d}$ represents coefficients of coordinates as column vectors. Since the mass matrix is generally unknown without the prior knowledge of the structure information, the coefficients matrix α can be obtained in the following least square sense:

$$(18) \qquad \min_{\alpha} \|P - \sum_{i=1}^N \phi_i \alpha_i\|_F^2 = \min_{\alpha} \|P - \Phi \alpha\|_F^2,$$

Consider the relationship between restricted global coordinate $\{P_{\Omega_j}\}$ and the input local coordinate $\{Q_j\}$, we already have that for any $1 \le j \le L$, $P_{\Omega_j} = Q_j R_j + \mathbf{1} b_j$. Therefore the norm $\|P - \Phi \alpha\|_2^2$ in local patch Q_j can be equivalently written as $\|P_{\Omega_j} - \Phi_{\Omega_j} \alpha\|_2^2 = \|Q_j R_j + \mathbf{1}_j b_j - \Phi_{\Omega_j} \alpha\|_2^2 = \|Q_j + \mathbf{1}_j b_j R_j^\top - \Phi_{\Omega_j} \alpha R_j^\top\|_2^2$. Since $b_j R_j^\top$ is still a translation vector in $\mathbb{R}^d$, the norm can be simplified as $\|Q_j - \Phi_{\Omega_j} \alpha R_j - \mathbf{1}_j b_j\|_2^2$ without loss of generality. As a result, with the given Φ and all the local coordinate $\{Q_j\}$, we propose the following model for stitching point clouds $\{Q_j\}_{j=1}^L$:

$$(19) \qquad \min_{\alpha \in \mathbb{R}^{N \times d}, \{R_j \in O(d),\ b_j \in \mathbb{R}^d\}_{j=1}^L} \mathcal{E}(\alpha, R_j, b_j) = \frac{1}{2} \sum_{j=1}^L \|Q_j - \Phi_{\Omega_j} \alpha R_j - \mathbf{1}_j b_j\|_F^2,$$

Once $\{R_j\}$ and $\{b_j\}$ are obtained, we can find all the adjusted coordinates $P_{\Omega_j} = Q_j R_j + \mathbf{1}_j b_j$, which leads to the global coordinate P ultimately.

Different from the synchronization method and the spectral relaxation method considered in [18, 14], we design the following method to solve the nonconvex problem (19) by updating α, $\{R_j\}$ and $\{b_j\}$ iteratively:

$$(20) \qquad \begin{cases} \alpha^{k+1} = \alpha^k - \big(\sum_{j=1}^L \Phi_{\Omega_j}^\top \Phi_{\Omega_j}\big)^{-1} \nabla_{\alpha} \mathcal{E}(\alpha^k, R_j^k, b_j^k), \\ R_j^{k+1} = \mathcal{K}(R_j^k, \nabla_{R_j} \mathcal{E}(\alpha^{k+1}, R_j^k, b_j^k)), \qquad \forall 1 \le j \le L, \\ b_j^{k+1} = b_j^k - \frac{1}{\mathrm{Card}(\Omega_j)} \nabla_{b_j} \mathcal{E}(\alpha^{k+1}, R_j^{k+1}, b_j^k), \qquad \forall 1 \le j \le L. \end{cases}$$

where we update α and $\{b_j\}$ using Newton's method and it is straightforward to check

$$(21) \qquad \begin{cases} \nabla_{\alpha} \mathcal{E}(\alpha^k, R_j^k, b_j^k) = -\sum_{j=1}^L \Phi_{\Omega_j}^\top (Q_j - \Phi_{\Omega_j} \alpha^k R_j^k - \mathbf{1}_j b_j^k)(R_j^k)^\top, \\ \nabla_{R_j} \mathcal{E}(\alpha^{k+1}, R_j^k, b_j^k) = -(\alpha^k)^\top \Phi_{\Omega_j}^\top (Q_j - \Phi_{\Omega_j} \alpha^k R_j^k - \mathbf{1}_j b_j^k), \\ \nabla_{b_j} \mathcal{E}(\alpha^{k+1}, R_j^{k+1}, b_j^k) = -\mathbf{1}_j^\top (Q_j - \Phi_{\Omega_j} \alpha^k R_j^k - \mathbf{1}_j b_j^k). \end{cases}$$

We next describe a method of updating the orthogonality constrained variable $\{R_j\}$. The operator $\mathcal{K}$ proposed in [61] is designed for searching the gradient descent direction on the Stiefel manifold, the geometric description of the set of orthogonal matrices. To realize the operator $\mathcal{K}(\nabla R_j^k)$, we first define the skew-symmetric operator as

$$(22) \qquad \begin{aligned} G_j^k &= \nabla_{R_j} \mathcal{E}(\alpha^{k+1}, R_j^k, b_j^k) \\ S_j^k &= G_j^k (R_j^k)^\top - R_j^k (G_j^k)^\top, \quad \forall 1 \le j \le L, \end{aligned}$$

then the new trial point R_j^{k+1} satisfing $R_j^{k+1}(R_j^{k+1})^\top = I$ can be generated by

$$(23) \qquad R_j^{k+1} = \mathcal{K}(R_j^k, G_j^k) = (1 + \frac{\delta}{2}S_j^k)^{-1}(1 - \frac{\delta}{2}S_j^k)R_j^k,$$

where the step size δ can be obtained by setting fixed value or line search methods. More detailed discussion about the orthogonality preserving property and convergence of this approach can be found in [61]. With the above explanation of (20), we summarize an algorithm of solving (19) as Algorithm 2.

Algorithm 2 Gradient Descent method to solve (19)

Initialization: Set the initial values such that $\boldsymbol{\alpha}^0 = 0, b_j^0 = 0, R_1^0 = I_{d\times d}$. By roughly estimating the orthogonal transform matrix $R_{i,j}$ between neighbourhood patches Ω_i and Ω_j with overlapping points, one can set the initial guess of $R_j^0 = R_i^0 R_{i,j}$. Using the broad first search (BFS) scheme, the rotation matrix of all connected patches can be estimated. If a patch Ω_i is isolated to any visited patches in the BFS algorithm, set $R_i^0 = I_{d\times d}$.

while $\frac{|E(k)-E(k-1)|}{E(k)} > \epsilon$ **do**

   1. $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k + (\sum\limits_{j=1}^{L} \Phi_{\Omega_j}^\top \Phi_{\Omega_j})^{-1}(\sum\limits_{j=1}^{L} \Phi_{\Omega_j}^\top(Q_j - \Phi_{\Omega_j}\boldsymbol{\alpha}^k R_j^k - \mathbf{1}_j b_j^k)(R_j^k)^\top).$

   2. $R_j^{k+1} = (1 + \frac{\delta}{2}S_j^k)^{-1}(R_j^k - \frac{\delta}{2}S_j^k R_j^k)$ where $G_j^k = -(\boldsymbol{\alpha}^k)^\top \Phi_{\Omega_j}^\top(Q_j - \Phi_{\Omega_j}\boldsymbol{\alpha}^k R_j^k - \mathbf{1}_j b_j^k)$, $S_j^k = G_j^k(R_j^k)^\top - R_j^k(G_j^k)^\top$ $\forall 1 \le j \le L$ and δ is the step size which can be solved by line search method.

   3. $b_j^{k+1} = b_j^k + \frac{1}{\text{Card}(\Omega_j)}\mathbf{1}_j^\top(Q_j - \Phi_{\Omega_j}\boldsymbol{\alpha}^k R_j^k - \mathbf{1}_j b_j^k),\ \forall 1 \le j \le L,$

   4. Calculate $E(k+1) = \mathcal{E}(\boldsymbol{\alpha}^{k+1}, R_j^{k+1}, b_j^{k+1})$ for the stopping criteria.

end while

   5. Apply $P_{\Omega_j} = Q_j R_j + \mathbf{1}_j b_j$ and combine all the P_{Ω_j} to find the P.

Based on the above method, we therefore propose a global coordinate reconstruction algorithm from merely random missing distance in each local patches. Given the local patches $\{\Omega_j\}, 1 \le j \le L$ with patch size not larger than K, we assume the local distance information include $r\%$ of the local distance of each patches. The full reconstrution procedure can be summarized as follows:

Step 1 Using the method discussed in Section 3.2 with local mesh reconstruction method to compute the first N Laplace-Beltrami eigenfunctions Φ.

Step 2 Using Algorithm 1 to generate the virtual local coordinate $\{Q_j\}, 1 \le j \le L$ for each patch.

Step 3 Using Algorithm 2 to find the global coordinate P from Φ and $\{Q_j\}$.

5. Numerical Results. In this section, numerical tests are presented to illustrate the proposed methods for solving PDEs on manifold represented as incomplete inter-point distance information and the application to manifold stitching. First, we demonstrate the effectiveness of the proposed matrix completion method for coordinate reconstruction. We also illustrate the phase transition curve of successful reconstruction. Second, we test our method of computing the LB eigen-problem and illustrate that the proposed method of solving PDEs through local coordinate reconstruction is much more efficient than solving PDEs from global coordinate reconstruction. In addition, we also test the problem of solving Eiknoal equations and illustrate

the accuracy improvement of this approach to the canonical Dijkstra method. As applications, we also demonstrate preliminary results on point clouds reconstruction using global information from LB eigenfunctions and its potential application to dimension reduction problem. All numerical experiments are implemented by MATLAB in a PC with a 32G RAM and 16 dual-core 2.7 GHz CPUs.

5.1. Coordinates reconstruction from incomplete distances. The local coordinate reconstruction method based on matrix completion is a crucial part of solving differential equations on a manifold represented as incomplete inter-point distance data. To demonstrate the effectiveness of the method proposed in Section 2, we first test this method for global coordinate reconstruction, although our numerical method for solving differential equations only conduct coordinate reconstruction for a certain neighborhood of each point. Starting with a full distance matrix $D \in \mathbb{R}^{n \times n}$, the actual number of freedom in D is $(n^2 - n)/2$ as the diagonal elements of D are 0 and D is symmetric. We randomly chose $\gamma(n^2 - n)/2, (\gamma \in [0, 1])$ number of entries of D as the input incomplete distance matrix and apply Algorithm 1 to find a Gram matrix B. After that, global coordinates can be consequently generated from B by its eigen-decomposition as we described in Section 2.1.
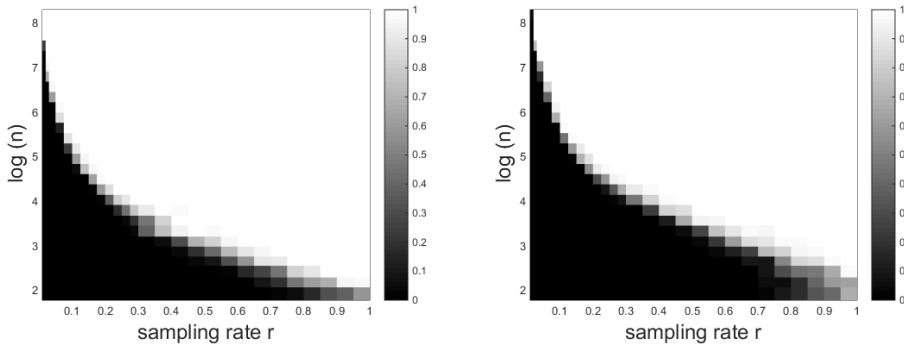

FIG. 1. *Phase transition of successful reconstruction rate ρ out of* 50 *tests of reconstructing a uniformly sampled unit sphere. Left: successful reconstruction rate of the proposed model* (2) *based on Gram matrix. Right: successful reconstruction rate based on distance completion model* (12).

Our first numerical test is conducted for checking successful reconstruction rate of reconstructing the input incomplete distance matrices. We test our model for different size matrices with fixed rank. By uniformly choosing n points on the unit sphere, we obtained the corresponding pairwise Euclidean distance matrix D_T and the associated Gram matrix B_T as ground truth. After that, we randomly choose $\gamma n(n-1)/2$ entries of D_T as available information and apply our reconstruction algorithm. We mean that the reconstruction is successful if the relative error between B_T and the reconstructed Gram matrix under the Frobenius norm is less than 10^{-3}. Given n and γ, we run 50 tests and record the rate of successful reconstruction as $\rho = \dfrac{\# \text{ successful reconstruction}}{50}$. Based on the same input, Figure 1 plots the phase transition of ρ using Gram matrix completion model (2) (the left image in Figure 1) and distance completion model (12) (the right image in Figure 1) in terms of the sampling rate γ and the logarithm of total number of points $\log(n)$. Both images clearly show that for a fixed rank problem, successful reconstruction needs less portion of distance information as number of points increases. As we expect, the Gram

13

matrix completion model (2) has a larger domain of successful reconstruction than the model (12) using distance matrix completion. This is because the rank of Gram matrix is less than the rank of distance matrix and an additional semi-positive definite constraint is also impose for Gram matrix which further decrease the number of freedom. Therefore, we use model (2) as our major local coordinate reconstruction tool to approximate differential operators in the rest of numerical experiments.

Data \ γ		1%	2%	3%	5%	10%	20%
S^2	E_B	7.157E-1	1.376E-3	4.791E-4	2.474E-4	1.342E-5	4.262E-5
	ρ	0%	92%	100%	100%	100%	100%
Cow	E_B	4.9427E-5	3.980E-4	1.837E-4	5.319E-5	1.4072E-5	2.155E-5
	ρ	100%	100%	100%	100%	100%	100%
Swiss roll	E_B	2.722E-4	2.894E-4	1.633E-4	5.054E-5	1.704E-5	1.114E-5
	ρ	100%	100%	100%	100%	100%	100%

TABLE 1

Rate of The successful reconstruction ρ and the average relative error E_B of the Gram matrix out of 50 tests by the proposed model (2) from distances with information availability rate γ.
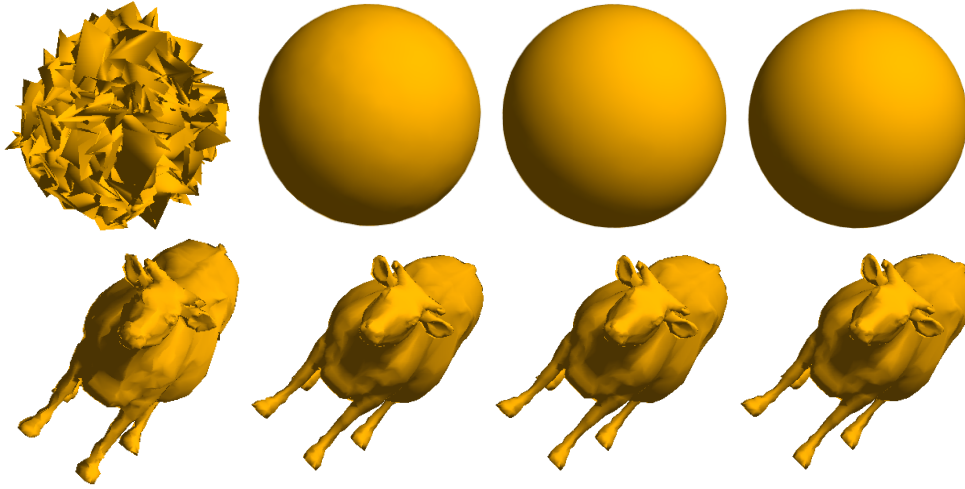

FIG. 2. *Globally reconstructed coordinates of a unit sphere (1002 points) and a cow (2601 points) using proposed model (2). Images from left top to right bottom are reconstructed from distance with of 1%, 2%, 3%, and 5% available informatoin, respectively.*

We further test distance matrices from different data including point clouds sampled from the unit sphere(1002 points), a cow surface (2602 points), and a Swiss roll surface (2048 points) using our Gram matrix completion model. Table 1 reports the relative error of the Gram matrix and rates of successful reconstruction out of 50 tests for each γ. Figure 2 shows that with 1% distance information, the reconstructed coordinates cannot formulate an outline of the unit sphere while the same rate of distance information can successfully reconstruct the cow shape. This is compatible with our observation showed in Figure 1, which indicates less percentage information is need for successfully reconstructing matrices with larger size. In addition, Table 1 and Figure 2 also demonstrate both quantitatively and qualitatively that the proposed reconstruction model can provide very good coordinate reconstruction once the available distance information is sufficient.
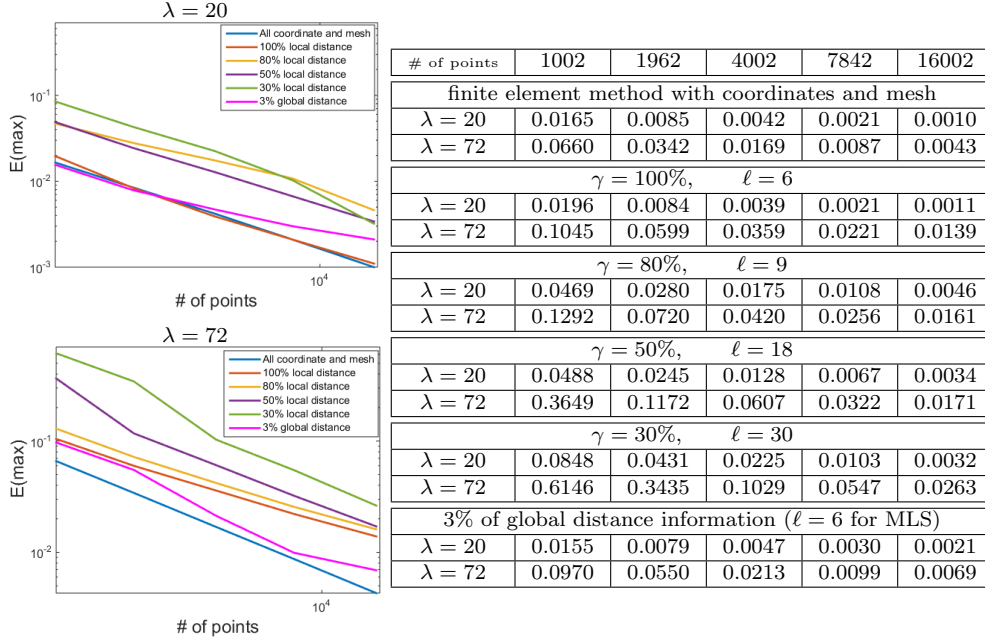
# of points	1002	1962	4002	7842	16002
finite element method with coordinates and mesh					
$\lambda = 20$	0.0165	0.0085	0.0042	0.0021	0.0010
$\lambda = 72$	0.0660	0.0342	0.0169	0.0087	0.0043
$\gamma = 100\%,\quad \ell = 6$					
$\lambda = 20$	0.0196	0.0084	0.0039	0.0021	0.0011
$\lambda = 72$	0.1045	0.0599	0.0359	0.0221	0.0139
$\gamma = 80\%,\quad \ell = 9$					
$\lambda = 20$	0.0469	0.0280	0.0175	0.0108	0.0046
$\lambda = 72$	0.1292	0.0720	0.0420	0.0256	0.0161
$\gamma = 50\%,\quad \ell = 18$					
$\lambda = 20$	0.0488	0.0245	0.0128	0.0067	0.0034
$\lambda = 72$	0.3649	0.1172	0.0607	0.0322	0.0171
$\gamma = 30\%,\quad \ell = 30$					
$\lambda = 20$	0.0848	0.0431	0.0225	0.0103	0.0032
$\lambda = 72$	0.6146	0.3435	0.1029	0.0547	0.0263
3% of global distance information ($\ell = 6$ for MLS)					
$\lambda = 20$	0.0155	0.0079	0.0047	0.0030	0.0021
$\lambda = 72$	0.0970	0.0550	0.0213	0.0099	0.0069

FIG. 3. *Computation errors for LB eigenvalues $\lambda = 20, 72$ on the unit sphere based on incomplete distance $D_{\gamma,\ell}$ with KNN information. Left: convergence curves. Right: Relative errors for input distance matrices with different sample size.*

5.2. Solve LB eigenvalue problem from distance data with missing values. Our second numerical experiment illustrates the proposed method of solving the LB eigenvalue problem on manifolds represented as incomplete inter-point distance information. Considering a point cloud $\{p_1, \cdots, p_n\}$ uniformly sampled on a given manifold $\mathcal{M} \subset \mathbb{R}^p$, we compute the associated distance matrix D provided by pairwise Euclidean distance for this point cloud. Our numerical experiments illustrate the proposed method for computing the LB eigenvalue problem only based on the incomplete distance. More precisely, we first uniformly randomly choose a subset $\Omega_\gamma \subset \{(i,j) \mid 1 \leq j < i \leq n\}$, such that $|\Omega_\gamma| = \lceil \gamma n(n-1)/2 \rceil$. Then incompletion distance information is provided as

$$D_{\gamma,\ell} = \left\{ D(i,j) \mid j \in N_\ell(i) \ \& \ (i,j) \in \Omega_\gamma, \quad i = 1, \cdots, n \right\}$$

where $N_\ell(i)$ denotes the index set of the nearest ℓ points to p_i. Thus, $D_{\gamma,\ell}$ roughly contains $\gamma\ell/n$ portion of D, which contains a very small portion of D if $\ell \ll n$ as we showed in our experiments. In addition, we measure computation accuracy using the normalized error $E_{\max,k} = \max_i \frac{|\tilde{\lambda}_{k,i} - \lambda_k|}{|\lambda_k|}$, where $\tilde{\lambda}_{k,i}$ are numerical approximation from our method, and i represents the multiplicity.

We first demonstrate our method on different data sampled from the unit sphere. It is known that the k-th LB eigenvalue of the unit sphere is given by $\lambda_k = k(k+1)$ with multiplicity $2k+1$. This allowss us to illustrate numerical accuracy of our methods on the unit sphere case. In the first test, we assume the K-nearest neighborhood (KNN) is provided and sample distance matrices for point clouds on the unit sphere by setting different values of $\gamma = 100\%, 80\%, 50\%, 30\%$ and with different KNN size $\ell = 6, 9, 18, 30$, respectively. Given a KNN size ℓ, the settings of local sampling rate

15

should above the border line as shown in Figure 1 to guarantee the exact local coordinate reconstruction. Conversely, given a fixed sampling rate of local distance, we need to set ℓ large enough to guarantee the successful reconstruction. With these input data, we apply the proposed strategy for approximating the LB operator only from local coordinates reconstruction and compute the first 100 LB eigenvalues. We typically chose approximation results of LB eigenvalues $\lambda = 20$ and $\lambda = 72$ to conduct accuracy analysis by illustrating the numerical error $E_{\max,n}$ in the right table of Figure 3. In addition, we list results from global coordinate reconstruction based on 3% of distance information. As a reference, we also list computation results from classical finite element method from all the exact coordinates of points and the mesh information, which can be experimentally regarded as the best computation result based on the whole distance information. It is clear that our method provides satisfactory approximation of the LB eigenvalue problem on the unit sphere. Convergence of our method is illustrated by the error curves showed in the left two plots in Figure 3. Moreover, we also observe the second order convergence numerically as $E_{\max,k}$ is approximately reduced by half if the number of total point doubled. This is compatible with convergence behavior of the moving least square method for solving the LB eigenvalue problem on point clouds discussed in [33, 34].

number of points				
1002	1962	4002	7842	16002
$\gamma = 100\%,$		$\ell = 6$		
0.26	0.51	1.01	2.03	4.05
$\gamma = 80\%,$		$\ell = 9$		
2.28	5.60	11.17	22.28	45.02
$\gamma = 50\%,$		$\ell = 18$		
4.03	8.09	16.14	32.44	64.71
$\gamma = 30\%,$		$\ell = 30$		
15.13	30.19	60.42	120.95	241.63
global reconstruction using 3% distance ($\ell = 6$ for MLS)				
2.09	9.86	40.13	154.40	597.06

TABLE 2

Comparisons of time consumption (minutes) of solving the LB eigenvalue problem based on local/global reconstruction methods.

We also report total time consumption in Table 2 for computing the first 100 LB eigenvalues with different setting of γ and ℓ in this test. Note that the major time-consuming part is local coordinates reconstruction based on incomplete distance information, whose complexity is dependent on the local size parameter ℓ and the total number of points n. Theoretically, our local coordinate reconstruction method generally has the time complexity $\mathcal{O}(\min(\ell^2 nm, \ell^3 n))$, while the global coordinate reconstruction has time complexity $\mathcal{O}(n^2 m)$. Here, m is denoted as the maximum number of eigenvalues used in the eigenvalue hard thresholding step of algorithm 1, which is typically chosen as 20 in our experiments. For large data sets satisfying $\ell \ll n$, local reconstruction method can save a huge amount of time. This advantage is also illustrated in Table 2, where local reconstruction method for solving LB eigenvalue problem is about 150 times faster than global reconstruction method if we choose $n = 16002$ and $\ell = 6$.

In practice, the input condition might be even weaker than our previous experiment as the exact KNN information may not be directly obtained from incomplete local distances. Therefore, we also test our method for reconstructing local coordinates only relying on KNN information from incomplete distance information. In

this scenario, we use the same setting as the previous test but require a larger rate of available distance information to have successfully local coordinate reconstruction. Our numerical results reported in Figure 4 also demonstrate nearly second order convergence of LB eigenvalue problem on the unit sphere. We further test the robustness of our method to distance with Gaussian perturbation. In this setting, we assume the input data is a incomplete distance $D_{80\%,30}$ corrupted by different levels of Gaussian noise with standard deviation $\sigma = 2\%d_{\max}, 5\%d_{\max}, 10\%d_{\max}, 15\%d_{\max}$ respectively, where $d_{\max} = \max_{i,j}\{D(i,j)\}$. The left table in Figure 5 reports numerical accuracy for different levels of Gaussian noise. It is clear that our method still provides a reasonable good approximation of LB eigenvalues. Moreover, we also plot the corresponding LB eigenfunctions in Figure 6, where LB eigenfunctions are color-coded on the unit sphere by setting red for positive values and blue for negative values. This figure illustrates consistent distribution patterns of LB eigenfunctions for noisy distance case. In addition, we also test our method of computing LB eigenvalue problems for incomplete distance information from point clouds sampled from more complicated manifolds such as armadillo (16519 points) and kitten (2884 points) surfaces. Figure 7 reports serval LB eigenfunctions on armadillo and kitten surfaces based on incomplete distance matrix $D_{50\%,30}$.
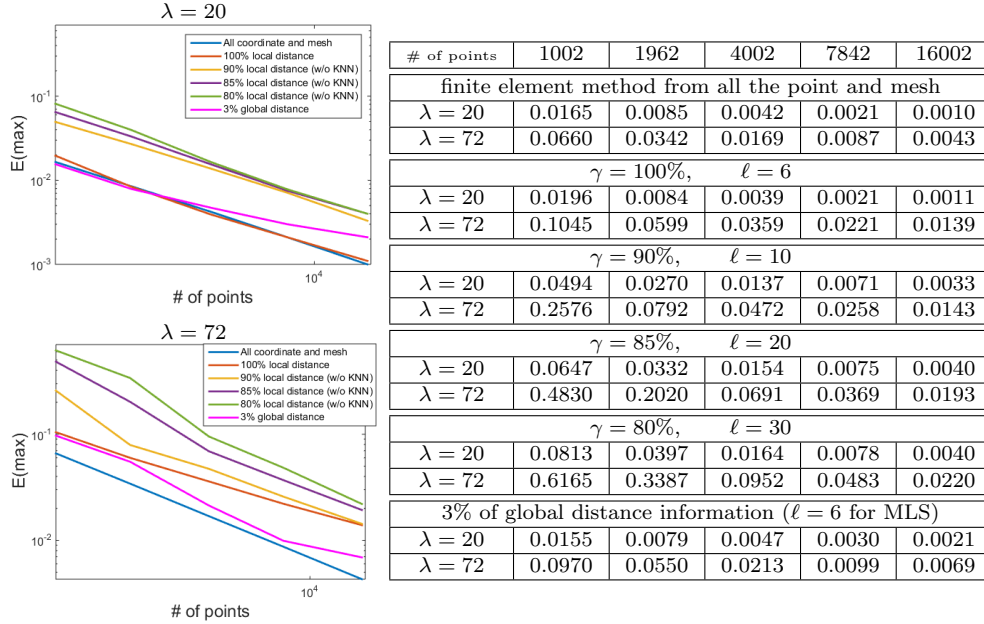

# of points	1002	1962	4002	7842	16002
finite element method from all the point and mesh					
$\lambda = 20$	0.0165	0.0085	0.0042	0.0021	0.0010
$\lambda = 72$	0.0660	0.0342	0.0169	0.0087	0.0043
$\gamma = 100\%, \quad \ell = 6$					
$\lambda = 20$	0.0196	0.0084	0.0039	0.0021	0.0011
$\lambda = 72$	0.1045	0.0599	0.0359	0.0221	0.0139
$\gamma = 90\%, \quad \ell = 10$					
$\lambda = 20$	0.0494	0.0270	0.0137	0.0071	0.0033
$\lambda = 72$	0.2576	0.0792	0.0472	0.0258	0.0143
$\gamma = 85\%, \quad \ell = 20$					
$\lambda = 20$	0.0647	0.0332	0.0154	0.0075	0.0040
$\lambda = 72$	0.4830	0.2020	0.0691	0.0369	0.0193
$\gamma = 80\%, \quad \ell = 30$					
$\lambda = 20$	0.0813	0.0397	0.0164	0.0078	0.0040
$\lambda = 72$	0.6165	0.3387	0.0952	0.0483	0.0220
3% of global distance information ($\ell = 6$ for MLS)					
$\lambda = 20$	0.0155	0.0079	0.0047	0.0030	0.0021
$\lambda = 72$	0.0970	0.0550	0.0213	0.0099	0.0069

FIG. 4. *Computation errors for LB eigenvalues* $\lambda = 20, 72$ *on the unit sphere based on incomplete distance* $D_{\gamma,\ell}$ *without knowing exact KNN information. Left: convergence curves. Right: Relative errors for input distance matrices with different sample size.*
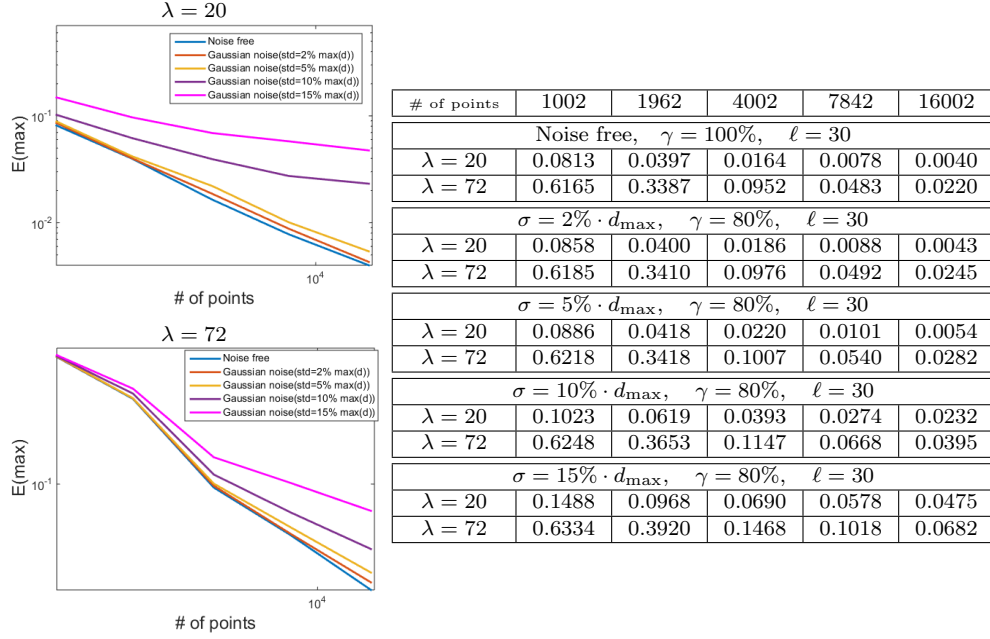
| # of points	1002	1962	4002	7842	16002
Noise free, $\quad \gamma = 100\%, \quad \ell = 30$					
$\lambda = 20$	0.0813	0.0397	0.0164	0.0078	0.0040
$\lambda = 72$	0.6165	0.3387	0.0952	0.0483	0.0220
$\sigma = 2\% \cdot d_{\max}, \quad \gamma = 80\%, \quad \ell = 30$					
$\lambda = 20$	0.0858	0.0400	0.0186	0.0088	0.0043
$\lambda = 72$	0.6185	0.3410	0.0976	0.0492	0.0245
$\sigma = 5\% \cdot d_{\max}, \quad \gamma = 80\%, \quad \ell = 30$					
$\lambda = 20$	0.0886	0.0418	0.0220	0.0101	0.0054
$\lambda = 72$	0.6218	0.3418	0.1007	0.0540	0.0282
$\sigma = 10\% \cdot d_{\max}, \quad \gamma = 80\%, \quad \ell = 30$					
$\lambda = 20$	0.1023	0.0619	0.0393	0.0274	0.0232
$\lambda = 72$	0.6248	0.3653	0.1147	0.0668	0.0395
$\sigma = 15\% \cdot d_{\max}, \quad \gamma = 80\%, \quad \ell = 30$					
$\lambda = 20$	0.1488	0.0968	0.0690	0.0578	0.0475
$\lambda = 72$	0.6334	0.3920	0.1468	0.1018	0.0682

FIG. 5. *Computation errors for LB eigenvalues $\lambda = 20, 72$ on the unit sphere based on incomplete corrupted distance $D_{\gamma, \ell} + N(0, \sigma^2)$ without KNN information. Left: convergence curves. Right: Relative errors for input distance matrices with different sample size.*
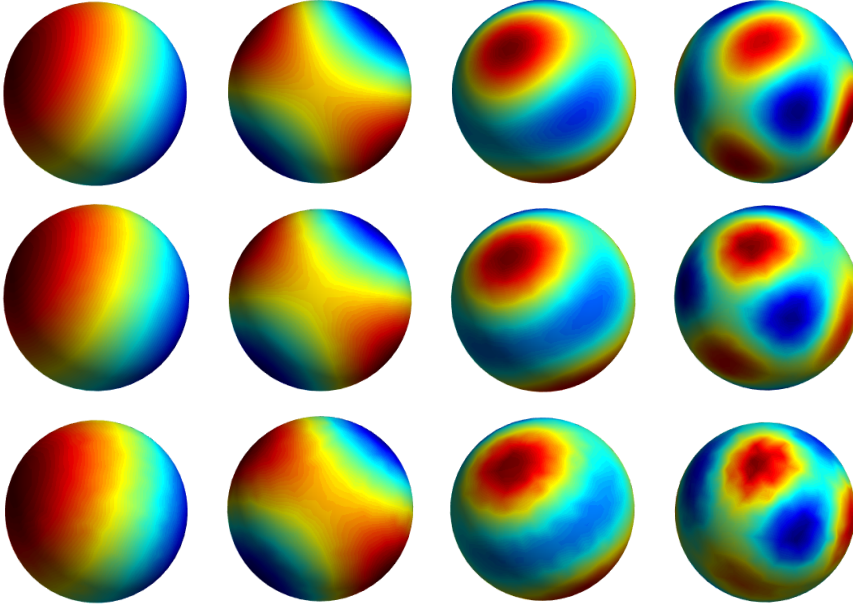
FIG. 6. *LB eigenfunctions are color-coded on the unit sphere (1002 points) based on incomplete distance $D_{80\%,30}$ corrupted by Gaussian noise of different levels. Images from top to bottom represent the LB eigenfunctions from noise free distance, distance with Gaussian noise of $\sigma = 5\% \cdot d_{\max}$, distance with Gaussian noise of $\sigma = 10\% \cdot d_{\max}$, respectively. Images from left to right represent the LB eigenfunctions corresponding to $\lambda = 2, 6, 12, 20$, respectively.*
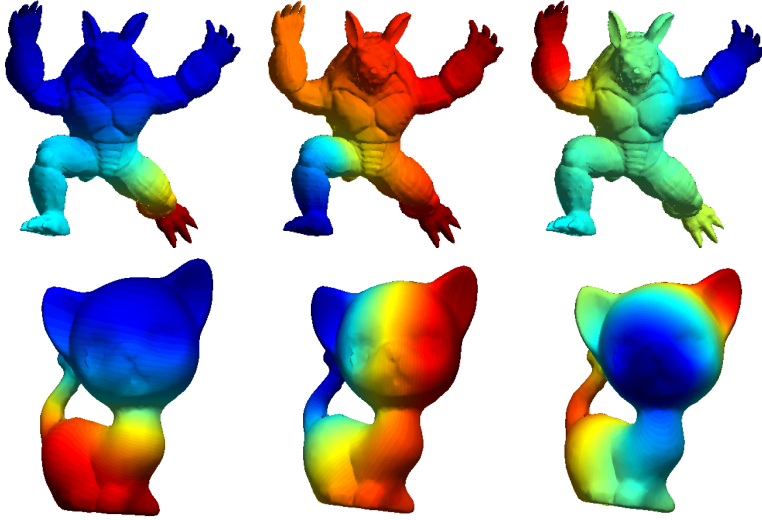
18

FIG. 7. *LB eigenfunctions for the armadillo (16519 points) and the kitten (2884 points) surfaces based on incomplete distance $D_{50\%,30}$. Top: the first three eigenfunctions are color-coded on the armadillo surface. Bottom: the first three eigenfunctions are color-coded on the kitten surfaces.*

As we discussed before, our methods can also be applied on manifolds with dimension more than 2 and co-dimension more than 1. Our third experiment tests the proposed method for computing LB eigenvalue problem on a 2 dimensional torus $T^2 = \{(\cos\theta_1, \sin\theta_1, \cos\theta_2, \sin\theta_2) \in \mathbb{R}^4 \mid \theta_1, \theta_2 \in [0, 2\pi)\}$ in $\mathbb{R}^4$ and a 3 dimension torus $T^3 = \{(\cos\theta_1, \sin\theta_1, \cos\theta_2, \sin\theta_2, \cos\theta_3, \sin\theta_3) \in \mathbb{R}^6 \mid \theta_1, \theta_2, \theta_3 \in [0, 2\pi)\}$ in $\mathbb{R}^6$. Solutions of the corresponding LB eigenvalue problem for both objects have closed forms, which enable us to conduct computation accuracy comparisons. For T^2, we choose $D_{80\%,30}$ from the corresponding distance matrix obtained from different sample size on T^2. Similarly setting is considered for T^3 but using $D_{85\%,30}$ with slightly large portion of information as the rank for each local Gram matrix for T^3 is higher than the one for T^2. This requires more information of the distance matrix from the matrix completion theory. Nevertheless, we only use less than 1% of the distance information for our experiments as $\ell = 30, n \geq 2500$. We remark that our computation does not assume the exact KNN information is available. This is the reason that we need higher portion of local information for accurate local coordinate reconstruction. In addition, we do not assume the dimension information is available in our computation. As we demonstrated in the first experiment, the value of γ can be even lower if the exact KNN information is provided. We compute the first 100 LB eigenvalues for T^2 and T^3 using incomplete distance matrices with different sample size . Figure 8 reports the convergence and relative errors of our method. Both cases demonstrate approximately second order convergence.

5.3. Solve Eikonal equation from incomplete distance. In this subsection, we test our method for solving a special hyperbolic equation, the Eikonal equation, on manifolds represented as incomplete distance information. In our experiments, assuming the KNN information is given as prior knowledge, we considered the incomplete local distance of uniform/non-uniform sampled point cloud of a unit sphere, in which the number of points vary from 1002 to 16002. In particular, we assume the point cloud always contains two points with coordinate $(0, 0, 1)$ and $(0, 0, -1)$, namely,
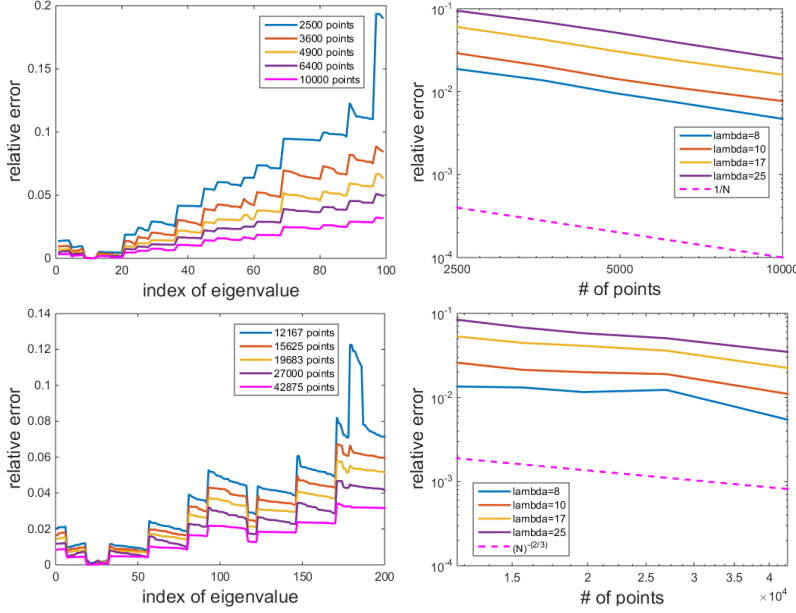
FIG. 8. *Top: The first 100 LB eigenvalues for T^2 given by $D_{80\%,30}$ with different sample size $n = 2500, 3600, 4900, 6400, 10000$. Top left: Relative errors. Top right: Convergence curves. Bottom: The first 200 LB eigenvalues for T^3 given by $D_{85\%,30}$ with different sample size $n = 12167, 15625, 19683, 27000, 42875$. Bottom left: Relative errors. Bottom right: Convergence curves.*

the north pole and the south pole (See Figure 9). To evaluate the quality of the reconstruction of distance map, we compare the relative error of the geodesic distance from the north pole to south pole (referred as E_{se}), and from the north pole to all other points (referred as E_{av}). The table in Figure 9 shows that our approach based on the fast marching method is much more accurate than the Dijkstra's method [19]. Moreover, relative errors of our method decrease as the number of points increase. Furthermore, for the uniform sample cases, the relative error of geodesic distance from north pole to south pole has first order convergence with respect to the density of points using our method, while the Dijkstra's method does not have the same convergence property.
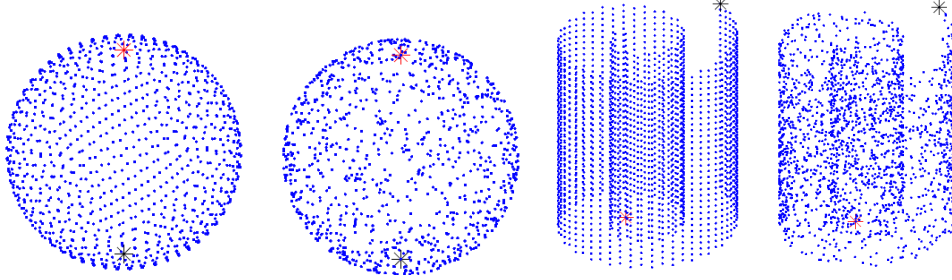
Similarly, we also compute the distance function on a Swiss roll:

$$S = \{(t + 0.1)\cos(t), (t + 0.1)\sin(t), 8\pi s) \mid t \in [0, 4\pi], s \in [0, 1]\}$$

based on incomplete local distance from uniform/non-uniform sampling, in which two diagonal points are fixed and their position can be seen in Figure 9. Using $D_{60\%,20}$ from different sample sizes vary from 1002 to 16002, the bottom Table in Figure 9 shows that our method can approximate the distance function with much less error than the Dijkstra's method, and approximate first convergence can also be observed from the case of the uniform sampled Swiss roll.

5.4. Global coordinate reconstruction from using patch stitching. In this experiment, we test our proposed global reconstruction model as an application of using global information from solving the LB eigenproblem based on incomplete distance. We assume the input manifold $\mathcal{M} \subset \mathbb{R}^3$ sampled as a point cloud and has been separated as partially overlapped L patches, namely, $\mathcal{M} = \bigcup_{j=1}^{L} \Omega_j$. We gen-

methods	sample size	1002	1962	4002	7842	16002
		Uniform sampling on S^2				
Dijkstra	E_{av}	0.0348	0.0285	0.0248	0.0232	0.0224
	E_{se}	0.008615	0.008606	0.008296	0.010642	0.011501
Our method	E_{av}	0.0113	0.0095	0.0080	0.0064	0.0022
	E_{se}	0.008100	0.005890	0.004110	0.002877	0.002158
		Non-Uniform sampling on S^2				
Dijkstra	E_{av}	0.0363	0.0344	0.0319	0.0305	0.0294
	E_{se}	0.011209	0.016090	0.018380	0.016391	0.019953
Our method	E_{av}	0.0200	0.0163	0.0141	0.0124	0.0088
	E_{se}	0.012016	0.008792	0.003742	0.001736	0.002765
		Uniform sampling on a Swiss roll				
Dijkstra	E_{av}	0.0119	0.0156	0.0198	0.0200	0.0203
	E_{se}	0.013104	0.021242	0.024560	0.024311	0.026004
Our method	E_{av}	0.0065	0.0044	0.0052	0.0033	0.0022
	E_{se}	0.003127	0.001637	0.001130	0.000783	0.000620
		Non-Uniform sampling on a Swiss roll				
Dijkstra	E_{av}	0.0400	0.0105	0.0114	0.0151	0.0180
	E_{se}	0.016612	0.015779	0.014573	0.016587	0.018649
Our method	E_{av}	0.0138	0.0053	0.0047	0.0029	0.0073
	E_{se}	0.004754	0.005189	0.003087	0.005171	0.007246

FIG. 9. *Top left two images: Incomplete distance data $D_{60\%,20}$ from 1002 points sampled uniformly and non-uniformly on the unit sphere with fixed north pole (red star) and south pole (blue star) for calculating geodesic distance. Top right two images: $D_{60\%,20}$ from 2048 points sampled uniformly and non-uniformly on the swiss role with a fixed starting point (red star) and an ending point (blue star) for calculating geodesic distance. Bottom table: Relative error of geodesic distances E_{av} (averaging from starting to all the points) and E_{se} (from the starting point to the ending point).*

erate distance matrices $\{D_j\}_{j=1}^L$ from pairwise Euclidean distance on each patch and randomly choose 50% information of each distance matrix. Based on these incomplete $\{D_j\}_{j=1}^L$, we first compute the first 100 LB eigenfunctions of $\mathcal{M}$ using the proposed method discussed in Section 3. After that, we reconstruct coordinates of each Ω_j from its incomplete distance using our matrix completion method. Finally, global reconstruction of $\mathcal{M}$ is obtained by the manifold stitching model discussed in Section 4, which can handle the coordinate inconsistency from each local reconstruction. In Figure 10, we report numerical results for reconstructing an armadillo surface (with 16519 points, 28 patches) and a kitten surface (with 2884 points, 30 patches). The two left images in Figure 10 illustrate coordinate reconstruction for each patch, the two middle images show the corresponding global reconstruction and the two right energy curves indicate the convergence of the proposed global reconstruction algorithm. It is clear to see that the proposed method can successfully reconstruct a manifold from a set of incomplete distance matrices from separated patches, while the global

data methods	Armadillo	Kitten	Swiss roll
Global reconstruction	35321.84	1315.50	622.65
Stitching	760.18	138.76	199.75

Time comparisons (seconds) between global reconstruction and the stitching model.

matrix completion model can not handle this case as the missing distance information is quite coherent. Moreover, compared to direct global coordinate reconstruction, the method of local distance reconstruction and patch stitching saves more computational time. To validate this claim, we measured that the 50% of local distance $\{D_j\}_{j=1}^L$ actually contains 3.59% of full distance D for armadillo surface and 3.45% for kitten surface. Then we also implement the direct global coordinate reconstruction for these two surfaces using the same rate of random missing global distance D. Table 3 shows that the computation time is much smaller using the stitching scheme especially for the armadillo surface with a large number of points.
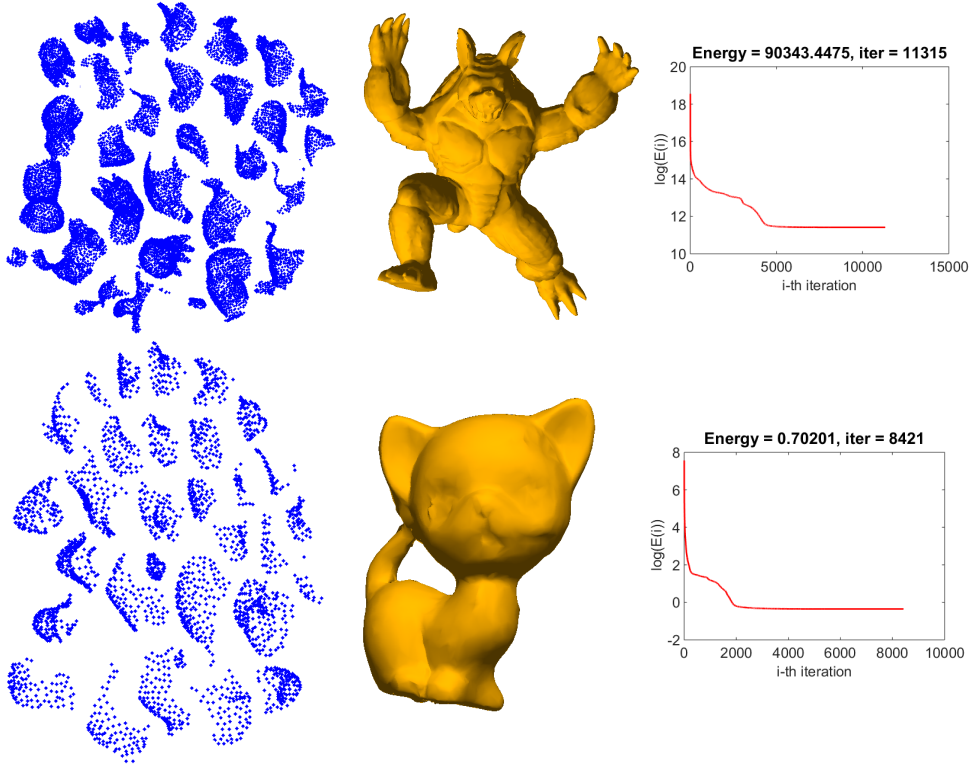
FIG. 10. *Top: Local (left) and global (middle) coordinate reconstruction of the armadillo surface (16519 points, 28 patches) and the corresponding convergence curve of the global reconstruction objective value $\log \mathcal{E}$. Bottom: Local (left) and global (middle) coordinate reconstruction of the kitten surface(2884 points, 30 patches) and the corresponding convergence curve of the global reconstruction objective value $\log \mathcal{E}$.*

We further conduct computation based on an incomplete geodesic distance matrix provided by a set of uniformly sampled points on the Swiss roll S used in the

previous example. It is straightforward to check that the geodesic distance between any two points $((t_1 + 0.1)\cos(t_1), (t_1 + 0.1)\sin(t_1), 8\pi s_1)$ and $((t_2 + 0.1)\cos(t_2), (t_2 + 0.1)\sin(t_2), 8\pi s_2)$ is given by $\sqrt{(8\pi(s_1 - s_2))^2 + (\mathcal{G}(t_2) - \mathcal{G}(t_1))^2}$, where $\mathcal{G}(t) = \frac{1}{2}[(t + 0.1)\sqrt{1 + (t + 0.1)^2} + \log(|\sqrt{1 + (t + 0.1)^2} + (t + 0.1)|)]$. In fact, this Swiss roll S is isometrical to a 2D flat domain $[\mathcal{G}(0), \mathcal{G}(4\pi)] \times [0, 8\pi]$. We uniformly sample 2048 points on S and construct a distance matrix D using pairwise geodesic distance. We test two ways of reconstruct the Swiss roll based on incomplete information of D. Firstly, similar as patch stitching simulations, we assume the Swiss roll is separated as 23 patches with partial overlap. Then we chose only 50% of local geodesic distance $\{D_j\}_{j=1}^{23}$, which is essentially 3.61% coherent sampling of totally geodesic distance D. The bottom row of figure 11 shows that the global stitched coordinate is a 2D rectangle, which is identical to the theoretical ground truth of the dimension reduced Swiss roll. Secondly, we assume 3.61% random sampled global geodesic distance is available and apply our coordinate reconstruction algorithm to have the global coordinate reconstruction for the Swiss roll. Top right image in figure 11 shows that direct global coordinate reconstruction from incoherent random missing distance is also applicable. Table 3 shows that with same sampling rate from the whole distance matrix, the direct global coordinate reconstruction takes much more computational time. The reason can be explained similarly as computation complexity discussion in subsections 3.1 and 5.2. The step of solving PDEs from local distance has time complexity $\mathcal{O}(\min(l^2 mn, l^3 n))$ while direct global coordinate reconstruction has time complexity $\mathcal{O}(mn^2)$ which is larger if total number of points n is large enough. Although the eigen-decomposition of Laplace-Beltrami matrix also involves global computation but it needs to be performed only once while eigenvalue thresholding is required in every iteration for direct global coordinate reconstruction. Therefore, it is not uprising that the stitching scheme saves a lot of time. We remark that our method of using geodesic distance to reconstructing manifolds is highly related to an important topic called nonlinear dimension reduction in machine learning and statistics community [42]. In our future work, we will investigate more along this direction based on our methods.

6. Conclusion. In this paper, we proposed a framework for discretizing PDEs on manifolds represented as incomplete inter-point distance information. Our approach is to conduct PDE discretization point-wisely from a local coordinate reconstruction. This strategy successfully avoids a time-consuming step of global coordinate reconstruction and leads to a method with complexity linearly scaling to the number of sample size. Our local reconstruction model is inspired from the recent advances of low-rank matrix completion theory, which guarantee accuracy local coordinate reconstruction by only requiring a very small portion of distance information. Our method can be viewed as natural extensions of the moving least square method and the local method method for solving PDEs on point clouds to manifold-structured data represented as incomplete inter-point distance information. As an application of using solutions of PDEs, we propose a new manifold reconstruction model by stitching local patches on the spectrum domain. Intense numerical experiments indicate effectiveness and efficiency of our methods for solving LB eigenvalue problem, Eikonal equations and manifold reconstruction using stitching methods based on global information from the LB eigensystem.
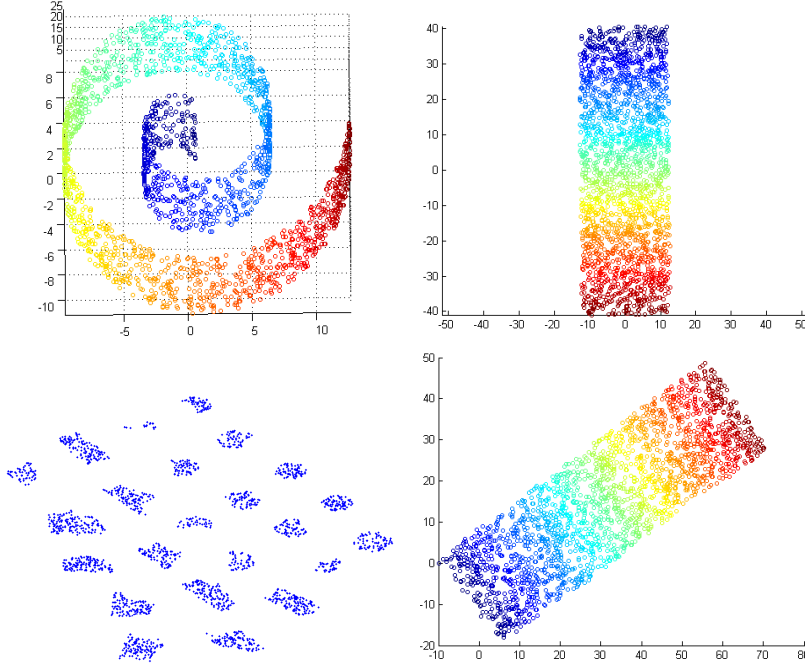
FIG. 11. *Top: the Swiss roll surface (left) and its dimensional reduction result (right) from randomly 3.61% of pair-wise geodesic distance. Bottom: local (left) and global (right) coordinates reconstruction of the Swiss roll from its 80% local geodesic distance.*

REFERENCES

[1] A. Y. ALFAKIH, A. KHANDANI, AND H. WOLKOWICZ, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Computational optimization and applications, 12 (1999), pp. 13–30.

[2] P. AXELSSON, *Processing of laser scanner data—algorithms and applications*, ISPRS Journal of Photogrammetry and Remote Sensing, 54 (1999), pp. 138–147.

[3] M. BELKIN AND P. NIYOGI, *Semi-supervised learning on Riemannian manifolds*, Machine Learning, 56 (2004), pp. 209–239.

[4] M. BELKIN, J. SUN, AND Y. WANG, *Constructing Laplace operator from point clouds in $\mathbb{R}^d$*, in Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, USA, 2009, pp. 1031–1040.

[5] P. BÉRARD, G. BESSON, AND S. GALLOT, *Embedding Riemannian manifolds by their heat kernel*, Geometric & Functional Analysis, 4 (1994), pp. 373–398.

[6] B. BERGER, J. KLEINBERG, AND T. LEIGHTON, *Reconstructing a three-dimensional model with arbitrary errors*, Journal of the ACM (JACM), 46 (1999), pp. 212–235.

[7] M. BERTALMIO, L.-T. CHENG, S. OSHER, AND G. SAPIRO, *Variational problems and partial differential equations on implicit surfaces*, Journal of Computational Physics, 174 (2002), pp. 759–780.

[8] M. BERTALMIO, G. SAPIRO, L.-T. CHENG, AND S. OSHER, *A framework for solving surface partial differential equations for computer graphics applications*, UCLA CAM Report (00-43), (2000).

[9] P. BISWAS, T.-C. LIANG, K.-C. TOH, Y. YE, AND T.-C. WANG, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE transactions on automation science and engineering, 3 (2006), pp. 360–371.

[10] I. BORG AND P. J. GROENEN, *Modern multidimensional scaling: Theory and applications*, Springer Science & Business Media, 2005.

[11] J. BRANDMAN, *A level-set method for computing the eigenvalues of elliptic operators defined on compact hypersurfaces*, Journal of Scientific Computing, 37 (2008), pp. 282–315.

[12] M. M. Bronstein and I. Kokkinos, *Scale-invariant heat kernel signatures for non-rigid shape recognition*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2010), pp. 1704–1711.

[13] E. Candès and B. Recht, *Exact matrix completion via convex optimization.*, Found. of Comput. Math., 9 (2008), pp. 717–772.

[14] K. N. Chaudhury, Y. Khoo, and A. Singer, *Global registration of multiple point clouds using semidefinite programming*, SIAM Journal on Optimization, 25 (2015), pp. 468–501.

[15] I. Chavel, *Eigenvalues in Riemannian geometry*, Academic press. INC, 1984.

[16] R. R. Coifman and S. Lafon, *Diffusion maps*, Applied and computational harmonic analysis, 21 (2006), pp. 5–30.

[17] G. M. Crippen, T. F. Havel, et al., *Distance geometry and molecular conformation*, vol. 74, Research Studies Press Taunton, UK, 1988.

[18] M. Cucuringu, Y. Lipman, and A. Singer, *Sensor network localization by eigenvector synchronization over the euclidean group*, ACM Transactions on Sensor Networks (TOSN), 8 (2012), p. 19.

[19] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische mathematik, 1 (1959), pp. 269–271.

[20] G. Dziuk and C. M. Elliott, *Finite element methods for surface PDEs*, Acta Numerica, 22 (2013), pp. 289–396.

[21] O. Faugeras and M. Hebert, *The representation, recognition, and locating of 3-d objects*, The international journal of robotics research, 5 (1986), pp. 27–52.

[22] X. Ji and H. Zha, *Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling*, in INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4, IEEE, 2004, pp. 2652–2661.

[23] I. Jolliffe, *Principal component analysis*, Wiley Online Library, 2002.

[24] P. W. Jones, M. Maggioni, and R. Schul, *Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels*, Proceedings of the National Academy of Sciences, 105 (2008), pp. 1803–1808.

[25] R. Kimmel and J. A. Sethian, *Computing geodesic paths on manifolds*, Proceedings of the National Academy of Sciences, 95 (1998), pp. 8431–8435.

[26] J. B. Kruskal and M. Wish, *Multidimensional scaling*, vol. 11, Sage, 1978.

[27] R. Lai and T. F. Chan, *A framework for intrinsic image processing on surfaces*, Computer vision and image understanding, 115 (2011), pp. 1647–1661.

[28] R. Lai, J. Li, and A. Tasissa, *Exact reconstruction of distance geometry problem using low-rank matrix completion*, Preprint, (2017).

[29] R. Lai, J. Liang, and H. Zhao, *A local mesh method for solving PDEs on point clouds.*, Inverse Problems & Imaging, 7 (2013).

[30] R. Lai, Y. Shi, K. Scheibel, S. Fears, R. Woods, A. W. Toga, and T. F. Chan, *Metric-induced optimal embedding for intrinsic 3D shape analysis*, Computer Vision and Pattern Recognition (CVPR), (2010), pp. 2871–2878.

[31] R. Lai and H. Zhao, *Multiscale nonrigid point cloud registration using robust sliced-wasserstein distance via laplace-beltrami eigenmap*, SIAM Journal on Imaging Sciences, 10 (2017), pp. 449–483.

[32] B. Levy, *Laplace-Beltrami eigenfunctions: Towards an algorithm that understands geometry*, IEEE International Conference on Shape Modeling and Applications, invited talk, (2006).

[33] J. Liang, R. Lai, T. Wong, and H. Zhao, *Geometric understanding of point clouds using Laplace-Beltrami operator*, CVPR, (2012).

[34] J. Liang and H. Zhao, *Solving partial differential equations on point clouds*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1461–A1486.

[35] L. M. Lui, X. Gu, T. F. Chan, S. Yau, et al., *Variational method on riemann surfaces using conformal parameterization and its applications to image processing*, Methods and Applications of Analysis, 15 (2008), pp. 513–538.

[36] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, *Discrete differential-geometry operators for triangulated 2-manifolds*, Visualization and Mathematics III. (H.C. Hege and K. Polthier, Ed.) Springer Verlag, (2003), pp. 35–57.

[37] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, *Distance geometry: theory, methods, and applications*, Springer Science & Business Media, 2012.

[38] S. Osher and J. Sethian., *Fronts propagation with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations.*, Journal of computational physics, 79 (1988), pp. 12–49.

[39] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Experimental mathematics, 2 (1993), pp. 15–36.

[40] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM review, 52 (2010), pp. 471–501.

[41] M. Reuter, F. Wolter, and N. Peinecke, *Laplace-Beltrami spectra as Shape-DNA of surfaces and solids*, Computer-Aided Design, 38 (2006), pp. 342–366.

[42] S. T. Roweis and L. K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (2000), pp. 2323–2326.

[43] L. K. Saul and S. T. Roweis, *Think globally, fit locally: unsupervised learning of low dimensional manifolds*, Journal of Machine Learning Research, 4 (2003), pp. 119–155.

[44] J. Scott and P. J. Carrington, *The SAGE handbook of social network analysis*, SAGE publications, 2011.

[45] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proceedings of the National Academy of Sciences, 93 (1996), pp. 1591–1595.

[46] Y. Shi, R. Lai, S. Krishna, N. Sicotte, I. Dinov, and A. W. Toga, *Anisotropic Laplace-Beltrami eigenmaps: bridging Reeb graphs and skeletons*, in Computer Vision and Pattern Recognition Workshops, 2008, pp. 1–7.

[47] A. Singer, *A remark on global positioning from local distances*, Proceedings of the National Academy of Sciences, 105 (2008), pp. 9507–9511.

[48] A. Spira and R. Kimmel, *Geometric curve flows on parametric manifolds*, Journal of Computational Physics, 223 (2007), pp. 235–249.

[49] J. Stam, *Flows on surfaces of arbitrary topology*, ACM Transactions On Graphics (TOG), 22 (2003), pp. 724–731.

[50] J.-L. Starck, F. Murtagh, and A. Bijaoui, *Image processing and data analysis: the multiscale approach*, Cambridge University Press, 1998.

[51] J. Sun, M. Ovsjanikov, and L. Guibas, *A concise and provabley informative multi-scale signature baded on heat diffusion*, Eurographics Symposium on Geometry Processing, (2009).

[52] G. Taubin et al., *Geometric signal processing on polygonal meshes*, Eurographics State of the Art Reports, 4 (2000), pp. 81–96.

[53] M. J. Todd, *Semidefinite optimization*, Acta Numerica, 10 (2001), pp. 515–560.

[54] P. Tseng, *Convergence of a block coordinate descent method for nondifferentiable minimization*, Journal of optimization theory and applications, 109 (2001), pp. 475–494.

[55] B. Vallet and B. Levy, *Spectral geometry processing with manifold harmonics*, Computer Graphics Forum (Proceedings Eurographics), (2008).

[56] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Review, 38 (1996), pp. 49–95.

[57] Y. Wang, L. M. Lui, . Gu, K. M. Hayashi, T. F. Chan, A. W. Toga, P. M. Thompson, and S. Yau, *Brain surface conformal parameterization using riemann surface structure*, IEEE transactions on medical imaging, 26 (2007), pp. 853–865.

[58] M. Waterman, *Introduction to computational biology: maps, sequences and genomes*, CRC Press, 1995.

[59] K. Q. Weinberger, F. Sha, and L. K. Saul, *Learning a kernel matrix for nonlinear dimensionality reduction*, in Proceedings of the twenty-first international conference on Machine learning, ACM, 2004, p. 106.

[60] Z. Wen, D. Goldfarb, and W. Yin, *Alternating direction augmented lagrangian methods for semidefinite programming*, Mathematical Programming Computation, 2 (2010), pp. 203–230.

[61] Z. Wen and W. Yin, *A feasible method for optimization with orthogonality constraints*, Mathematical Programming, 142 (2013), pp. 397–434.

[62] G. Xu, *Convergent discrete Laplace-Beltrami operators over triangular surfaces*, in Geometric Modeling and Processing, 2004. Proceedings, IEEE, 2004, pp. 195–204.

[63] H. Zhao, *A fast sweeping method for eikonal equations*, Mathematics of computation, 74 (2005), pp. 603–628.