Geometry-Driven Model Reduction

Thesis by

Maxim A. Budninskiy

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2019
Defended November 27, 2018

Maxim A. Budninskiy
ORCID: 0000-0002-9288-0249

ACKNOWLEDGEMENTS

I recall telling my parents that mathematics was by far my favorite subject after I finished first grade. Of course, I had no idea back then how vast, complex, and fascinating the field of mathematics is. However, I was very fortunate to have many unique opportunities to learn more about it over the years, and I am extremely grateful for that.

I would like to thank my awesome advisor Mathieu Desbrun – this work would not have been possible without his support and guidance. Thank you for introducing me to the world of discrete geometry, for sharing your expertise with me, for encouraging and helping me along the way, and for exploring new ideas together with me. It is always great to work with you!

I would also like to thank my committee members – Houman Owhadi, Venkat Chandrasekaran, and Andrew Stuart – for their comments and questions.

I have been fortunate to collaborate with amazing researchers. I would like to thank Yiying Tong for his insightful suggestions and rigorous explanations in all the research projects we worked on together; Beibei Liu for making difficult projects much more fun to work on, for inspiring discussions, and for her help with implementations; Houman Owhadi for his extensive support and detailed exposition of the game theoretical approach to numerical approximation during our meetings; Fernando de Goes for helping me get started at Caltech and sharing his knowledge of DDG and SEC; Pierre Alliez for his contributions to the anisotropic meshing project; Tudor Ratiu for long discussions about modern geometry, and for introducing me to Mathieu; and finally, Vladimir Alexandrov for his guidance during my years at Moscow State University.

Thank you to my friends – Yong Sheng, Armeen, Corina, Utkan, Surabhi, Ramya, Matteo, Brian, Petr, Vahe, Arthur, Stas, Sasha – for meaningful conversations, exciting adventures, and for all the fun times we had together.

Finally, I would like to thank my family – my parents Andrey and Marina, parents-in-law Tigran and Tagui, and sister-in-law Anush – for always supporting and encouraging me. Мама и папа, спасибо за вашу поддержку, заботу и внимание! I am incredibly grateful to my wife, soulmate, and best friend Diana, who is always there for me in this wonderful journey.

ABSTRACT

In this thesis we bring discrete differential geometry to bear on model reduction, both in the context of data analysis and numerical simulation of physical phenomena.

First, we present a novel controllable as-isometric-as-possible embedding method for low- and high-dimensional geometric datasets through sparse matrix eigenanalysis. This approach is equally suitable for performing nonlinear dimensionality reduction on big data and nonlinear shape editing of 3D meshes and pointsets. At the core of our approach is the construction of a "multi-Laplacian" quadratic form that is assembled from local operators whose kernels only contain locally affine functions. Minimizing this quadratic form produces an embedding that best preserves all relative coordinates of points within their local neighborhoods. We demonstrate the improvements that our approach brings over existing nonlinear local manifold learning methods on a number of datasets, and formulate the first eigen-based as-rigid-as-possible shape deformation technique by applying our affine-kernel embedding approach to 3D data augmented with user-imposed constraints on select vertices.

Second, we introduce a new global manifold learning approach based on metric connection for generating a quasi-isometric, low-dimensional mapping from a sparse and irregular sampling of an arbitrary low-dimensional manifold embedded in a high-dimensional space. Our geometric procedure computes a low-dimensional embedding that best preserves all pairwise geodesic distances over the input pointset similarly to one of the staples of manifold learning, the Isomap algorithm, and exhibits the same strong resilience to noise. While Isomap relies on Dijkstra's shortest path algorithm to approximate geodesic distances over the input pointset, we instead propose to compute them through "parallel transport unfolding", a discrete form of Cartan's development, to offer robustness to poor sampling and arbitrary topology. Our novel approach to evaluating geodesic distances using discrete differential geometry results in a markedly improved robustness to irregularities and sampling voids. In particular, it does not suffer from Isomap's limitation to geodesically convex sampled domains. Moreover, it involves only simple linear algebra, significantly improves the accuracy of all pairwise geodesic distance approximations, and has the same computational complexity as Isomap. We also show that our connection-based distance estimation can be used for faster variants of Isomap such as Landmark-Isomap.

Finally, we introduce an operator-adapted multiresolution analysis for finite-element differential forms. From a given continuous, linear, bijective, and self-adjoint positive-definite operator $\mathcal{L}$, a hierarchy of basis functions and associated wavelets for discrete differential forms is constructed in a fine-to-coarse fashion and in quasi-linear time. The resulting wavelets are $\mathcal{L}$-orthogonal across all scales, and can be used to obtain a Galerkin discretization of the operator with a block diagonal stiffness matrix composed of uniformly well-conditioned and sparse blocks. Because our approach applies to arbitrary differential p-forms, we can derive both scalar-valued and vector-valued wavelets that block diagonalize a prescribed operator. Our construction applies to various types of computational grids, offers arbitrary smoothness orders of basis functions and wavelets, and can accommodate linear differential constraints such as divergence-freeness. We also demonstrate the benefits of the operator-adapted multiresolution decomposition for coarse-graining and model reduction of linear and nonlinear partial differential equations.

We conclude with a short discussion on how future work in geometric model reduction may impact other related topics such as semi-supervised learning.

PUBLISHED CONTENT AND CONTRIBUTIONS

Max is the first author and main contributor of the following publications, which contain contributions presented in this thesis:

[1] Max Budninskiy, Beibei Liu, Yiying Tong, and Mathieu Desbrun. Spectral affine-kernel embeddings. *Computer Graphics Forum*, 36(5):117–129, 2017. doi: 10.1111/cgf.13250.

[2] Max Budninskiy, Houman Owhadi, and Mathieu Desbrun. Operator-adapted wavelets for finite-element differential forms. *Under review*, 2018.

[3] Max Budninskiy, Gloria Yin, Leman Feng, Yiying Tong, and Mathieu Desbrun. Parallel transport unfolding: A connection-based manifold learning approach. *Under review*, 2018.

Other work:

[1] Max Budninskiy, Beibei Liu, Fernando de Goes, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. Optimal voronoi tessellations with hessian-based anisotropy. *ACM Trans. Graph.*, 35(6):242:1–242:12, November 2016. doi: 10.1145/2980179.2980245.

[2] Max Budninskiy, Beibei Liu, Yiying Tong, and Mathieu Desbrun. Power coordinates: A geometric construction of barycentric coordinates on convex polytopes. *ACM Trans. Graph.*, 35(6):241:1–241:11, November 2016. doi: 10.1145/2980179.2982441.

[3] Max Budninskiy, Ameera Abdelaziz, Yiying Tong, and Mathieu Desbrun. Laplacian-optimized semi-supervised learning. *Under review*, 2018.

[4] Fernando de Goes, Beibei Liu, Max Budninskiy, Yiying Tong, and Mathieu Desbrun. Discrete 2-tensor fields on triangulations. *Comput. Graph. Forum*, 33:13–24, August 2014. doi: 10.1111/cgf.12427.

CONTENTS

LIST OF FIGURES

Number *Page*

Chapter 1

INTRODUCTION

With the prevalence of big data and the exponentially growing amount of available information, the demand for numerical tools to analyze high-dimensional data sets and phenomena has become paramount in recent years. Despite significant theoretical and technological advancements, most analytical tools struggle with processing high-dimensional datasets due to the sheer amount of computations involved, an issue often referred to as the "curse of dimensionality". Broadly construed, model reduction tackles the issue of how to reduce the computational complexity (in time and/or in storage space) of common numerical tasks, be they in machine learning or computational physics. It aims to obtain a computationally efficient approximation of the original data or phenomenon by reducing its degrees of freedom without introducing significant numerical errors. For instance, model reduction can uncover a lower-dimensional representation of a given dataset while preserving relevant structural and statistical features (known as *dimensionality reduction* in machine learning), or filter out irrelevant high-frequency effects to simplify the model (called *homogenization* or *coarse graining* in numerical simulation). In this thesis, we show that discrete differential geometry can impact these two very different cases of model reduction.

Manifold learning approaches to dimensionality reduction assume that the input data samples, possibly with added noise, a low-dimensional manifold embedded in a high-dimensional space, and they seek to construct a structure-revealing parametrization of that manifold in low-dimensional Euclidean space. We introduce two novel geometric manifold learning methods: Spectral Affine-Kernel Embedding (SAKE), and Parallel Transport Unfolding (PTU). SAKE relies on local information extracted from overlapping neighborhood patches and uses sparse matrix eigenanalysis to produce an embedding that best preserves all relative point positions within that manifold; in contrast, PTU first estimates all pairwise geodesic distances between input points using parallel transport and then uses this global description of the data to construct a quasi-isometric embedding through eigenanalysis of a dense matrix. Hence, the main qualitative difference between these two proposed methods is the amount of geometric information they use, which manifests itself in the trade-off between computational complexity and stability with respect to noise and irregular

sampling. SAKE, which typically requires only $O(n^{1.5})$ number of operations, is ultimately faster, but less resilient than PTU, which has a complexity of $O(n^3)$ in its original form. We present the improvements that both our methods introduce compared to existing manifold learning algorithms and also formulate the first eigen-based 3D mesh and point set editing technique, built upon SAKE framework.

In the context of partial differential equations and dynamical systems, model reduction aims to simplify the problem with negligible or limited impact on the solution. In general, it is beneficial to express a differential equation in the language of differential geometry and exterior calculus, as it produces a concise and coordinate-independent formulation of the problem, often revealing important properties of the simulated phenomenon. While traditional discretizations of differential equations often fail to preserve crucial geometric structures, Discrete Exterior Calculus (DEC [68, 101]), Subdivision Exterior Calculus (SEC [62, 190]), and Finite Element Exterior Calculus (FEEC [9, 11]) are designed to preserve as many structural properties of differential operators as possible by working with carefully constructed finite element spaces of differential forms. We introduce the first construction of operator-adapted wavelets for spaces of differential forms as a tool for computational model reduction in DEC and SEC framework, extending the notion of "gamblets" ([140], [139]). Given a continuous, linear, bijective, self-adjoint, and positive-definite exterior operator mapping p-forms to p-forms, we describe a computational procedure to obtain adapted wavelets that are operator-orthogonal across all resolution levels, in quasilinear time. In particular, the resulting adaptivity manifests itself in numerical homogenization effect, which allows for a high quality approximation of the solution space with just a subset of wavelets associated with coarser resolution levels. In addition, a Galerkin discretization of the operator in the resulting wavelet space leads to a block diagonal stiffness matrix with sparse and well-conditioned blocks, allowing for efficient and independent linear solves for different scales. We also show that our framework can be modified to incorporate physically relevant constraints by detailing the construction of divergence-free operator-adapted wavelets.

1.1 Organization

- In Chapter 2, we review existing local and global manifold learning approaches, providing the necessary background for our contributions.
- In Chapter 3, we describe a local approach to manifold learning that we called *Spectral Affine-Kernel Embeddings*. We demonstrate the advantages that

this technique brings over existing local nonlinear dimensionality reduction methods on a number of datasets and also present a novel framework for nonlinear shape editing of 3D data as a straightforward modification of SAKE approach.

- In Chapter 4, we introduce Parallel Transport Unfolding, a global approach to manifold learning based on discrete metric connections. We discuss the theoretical foundations and benefits of our approach compared to the traditional Dijkstra-based geodesic estimation employed in the Isomap algorithm, and validate them in practice on a number of datasets.

- In Chapter 5, we present a new axiomatic derivation of the construction of operator-adapted wavelets, recently introduced in [140], that uses only simple linear algebra. We then extend this construction to offer operator-adapted wavelets for differential forms by leveraging the framework of Subdivision Exterior Calculus (SEC). We demonstrate numerical properties of the proposed approach, including homogenization effect, fast wavelet decay, and well-conditioning of wavelet stiffness matrices, for several differential equations acting on 1-forms (or equivalently, vector fields) in 2D.

- In Chapter 6, we provide a summary of our contributions and point out potential directions for future research.

Chapter 2

PREVIOUS WORK

We begin our exposition with a brief review of previous works in the two main applications we target in this thesis: manifold learning and multiresolution analysis.

2.1 Manifold Learning

The manifold assumption, which posits that high-dimensional datasets tend to concentrate on much lower dimensional curved spaces (manifolds), has been proven surprisingly useful in a variety of contexts. Manifold learning algorithms extract reduced dimensional representations of data, producing a low count of "intrinsic variables" to describe the high-dimensional input—with which markedly faster computations can be performed in the context of physical simulation, machine learning (to address the "curse of dimensionality"), facial animation [111], skeletal animation [13], video editing [149], and mesh parameterization [160], to mention a few.

Perhaps the most popular algorithm for dimensionality reduction is Principal Component Analysis (PCA), which uses eigenanalysis to find the most significant low-dimensional coordinates in which to express high-dimensional data in the most informative way; yet it is only useful when data lie on or close to a linear subspace of the high-dimensional space. Manifold learning algorithms can be viewed as nonlinear extensions to PCA: since data points are assumed to be samples from a low-dimensional (and, in general, curved) manifold that is embedded in a high-dimensional space, nonlinear dimensionality reduction algorithms attempt to uncover a structure-revealing parametrization of this manifold.

In this chapter we will present an overview of several popular manifold learning approaches, separating them into two classes: local methods, that rely on information extracted from small overlapping neighborhoods of the input points; and global methods, that take into account the geometry of the whole dataset.

While we focus on graph-based manifold methods that aim to preserve geometrically relevant information (in particular, relative positioning of points with respect to their neighbors, or pairwise distances), we briefly mention that many alternative nonlinear dimensionality reduction approaches exist: some are designed purely

for visualization purposes and actually introduce distortions to emphasize some properties of the data (e.g., tSNE [123]), while others aim to produce efficient representations of the input without preserving geometric features directly (e.g. dictionary learning, tensor factorizations, autoencoders, etc; see [2–4, 8, 12, 16, 98, 137, 169, 188]).

2.1.1 Problem Statement

Throughout our exposition, we consider an input data set $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$ of n points that irregularly sample (possibly with noise) a connected compact orientable d-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$, with $d \ll D$. We only assume that $\mathcal{M}$ possesses an atlas with a single chart, so that an injective d-dimensional parametrization of the manifold exists. The goal of manifold learning is to construct d-dimensional embedding of the input $\mathcal{S}$ as a pointset $\{\mathbf{z}_i\}_{i=1}^n \subset \mathbb{R}^d$. Finally, to simplify further expressions, we assemble a $D \times n$ matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ from the input points and denote the final embedding as a $d \times n$ matrix $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$.

2.1.2 Local Manifold Learning Methods

Local embedding methods, including Laplacian Eigenmaps [17], Locally Linear Embeddings (LLE [155]), Modified Locally Linear Embeddings (MLLE [203]), Local Tangent Space Alignment (LTSA [204]), and their many variants, infer the global structure of a nonlinear manifold by a careful analysis of the interactions between overlapping local neighborhoods. While these methods differ in how they approach the dimensionality reduction problem, they all encourage nearby points in the original space to be mapped to nearby points in the reduced space, relying on a sparse matrix eigenanalysis to deduce a global low-dimensional embedding that best aligns all local neighborhoods. In particular, the aforementioned approaches compute the final embedding as a collection of eigenvectors corresponding to the smallest eigenvalues of a method-dependent sparse matrix. While this procedure does not preserve the notion of scale or orientation (scaled eigenvectors remain eigenvectors), the output of spectral local methods can be interpreted as "nearly-isometric parametrization of input data *up to a linear transformation*". The local nature of this family of methods typically implies a computational complexity of $O(n^{1.5})$ [165] due to the sparsity of the matrix involved in the eigenanalysis. However, this sparsity is also a source of brittleness: while such methods try to preserve the *local* features of data as much as possible, they do not penalize global, large-scale distortions. This shortcoming gives rise to very warped and degenerate embeddings

in practice [63, 184], an issue that can be partially reduced by reinjecting global information (through distances between far away points) [107] at the expense of computational cost. Hessian-based Locally Linear Embedding (Hessian-LLE [71], [199]) may provide much improved results through the construction of approximate Hessian operators in local tangent spaces, thus enforcing local linear precision and avoiding spurious (harmonic) warping that Laplacian-based methods, including LLE and Laplacian Eigenmaps, suffer from. However, in the presence of noise and irregular sampling, the approximation of Hessians becomes unreliable, and this technique can end up with even worse warping than LLE (see Fig. 3.8): automatically "stitching" local flattened neighborhoods into a non-degenerate global unfolding that is as isometric as possible to the original high-dimensional manifold is inherently difficult. Local manifold learning methods are thus general and efficient enough to handle arbitrary inputs, but are rarely robust enough to offer a reliable global embedding. The motivation behind our Spectral Affine-Kernel Embeddings (Ch. 3) was to develop a fast local method with improved resilience to noise and irregular sampling.

2.1.3 Global Manifold Learning Methods

The geometry of an input dataset $\mathcal{S}$ can be fully characterized in multiple ways. One particularly convenient description is given by a set of all pairwise distances between its points, as it uniquely defines $\mathcal{S}$ up to an isometric transformation. As a result, it is often incorporated as a key component of global dimensionality reduction methods. In particular, Multidimensional Scaling (MDS) [54, 182] relies on all pairwise distances evaluated in the original input space $\mathbb{R}^D$ and poses a variational problem that aims to preserve those distances in the embedding. The Isomap technique [180] is a variant of MDS that attempts to reduce distortion in the mapping by preserving pairwise *geodesic* distances between all data points as well as possible. After forming a k-nearest neighbor graph of the data points as a representation of the input manifold, Isomap first solves the all-pairs shortest path problem (using Dijkstra's or Floyd-Warshall algorithm) to approximate geodesic distances between every pair of points in the graph. Finally, a low-dimensional embedding that best preserves these pairwise distances is then constructed with MDS through eigenanalysis of the corresponding Gram matrix [24]. The high computational complexity due to the eigenanalysis of a dense Gram matrix (in $O(n^3)$) can be further improved through probabilistic linear algebra [91], variants such as Landmark-Isomap [63], or by exploiting various numerical improvements

of MDS [32, 93, 110] that approximate its solution in near linear time. Since *all* pairwise geodesic distances are used, Isomap is particularly robust to noise, and for this reason it remains one of the most popular algorithms used for manifold learning.

In the remainder of this chapter, we will review MDS and Isomap in more detail, as they will be leveraged in Ch. 3 and Ch. 4.

Multidimensional Scaling Algorithm

The goal of classical MDS is to recover an $\mathbb{R}^d$ embedding of n points given a symmetric $n \times n$ matrix $\mathbf{D}$ of pairwise *squared* distances between them (with $n \geq d$). First, an $n \times n$ Gram matrix $\mathbf{G}$ is constructed from $\mathbf{D}$ via "double-centering" [24], using the relationship between a norm and associated inner product; that is, denoting identity matrix by $\mathbf{I}$ and the n-vector of ones by $\mathbf{e}$,

$$\mathbf{G} = -\frac{1}{2}(\mathbf{I} - \frac{1}{n}\mathbf{e}\,\mathbf{e}^T)\,\mathbf{D}\,(\mathbf{I} - \frac{1}{n}\mathbf{e}\,\mathbf{e}^T).$$

This Gram matrix may be further altered to represent a proper kernel, see [45]. The final embedding $\mathbf{Z}$ is found as the product of square roots of the d largest eigenvalues and corresponding eigenvectors of $\mathbf{G}$:

$$\mathbf{Z} = \sqrt{\mathbf{\Lambda}_d}\,\mathbf{Q}_d^T,$$

where $\mathbf{G} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ is the eigendecomposition of $\mathbf{G}$, and $(\mathbf{\Lambda}_d, \mathbf{Q}_d)$ denote the truncated matrices containing only the d largest eigenvalues of $\mathbf{G}$ and corresponding eigenvectors respectively. Consequently, the embedding is found by computing partial eigendecomposition $(\mathbf{\Lambda}_d, \mathbf{Q}_d)$, a numerical task of expected complexity $O(n^3)$ since the Gram matrix is dense.

MDS is, in fact, a variational approach, since $\mathbf{Z}$ can be interpreted as a square root of a matrix given by the minimizer of the following optimization problem:

$$\begin{aligned} \min_{\mathbf{G}_d \in \mathbb{R}^{n \times n}} \quad & ||\mathbf{G} - \mathbf{G}_d||_2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{G}_d) \leq d. \end{aligned}$$

In other words, MDS recovers a d-dimensional embedding of n points whose Gram matrix is as close to $\mathbf{G}$ as possible (in 2-norm). Notice that a natural alternative approach would be to construct an embedding whose pairwise squared distances approximate $\mathbf{D}$ directly. However, since both formulations achieve zero error (perfect approximation) simultaneously [64], the first one is preferable due to its closed form solution. Despite the technical difference between these two formulations, MDS is

often interpreted as an embedding procedure that preserves pairwise distances as well as possible.

Proposition 2.1. *MDS is "linearly precise": assuming the pointset* $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$ *samples a linear* d*-dimensional subspace* $\mathcal{R}$ *of* $\mathbb{R}^D$ *(with* $n > d$*), its MDS embedding* $\mathbf{Z} \in \mathbb{R}^{d \times n}$ *preserves all pairwise distances of the input data and can be expressed as composition of a projection and a rigid body transformation of* $\mathbf{X} \in \mathbb{R}^{D \times n}$ *.*

Proof. Without loss of generality, assume that $\mathbf{X}$ is centered around origin; otherwise it can be translated first (and translation is an isometric transformation). Because $\mathcal{S}$ samples a d-dimensional linear subspace $\mathcal{R}$ of $\mathbb{R}^D$, the rank of $\mathbf{X}$ is at most d. Since $\mathbf{G} = \mathbf{X}^T\mathbf{X}$ by construction, we also have $\operatorname{rank} \mathbf{G} \leq d$. Therefore, the MDS embedding $\mathbf{Z} = \sqrt{\mathbf{\Lambda}_d}\,\mathbf{Q}_d^T$ (using the notations from above) satisfies $\mathbf{G} = \mathbf{Z}^T\mathbf{Z}$. As pointsets $\mathbf{X}$ and $\mathbf{Z}$ have the same Gram matrix $\mathbf{G}$, they also have the same pairwise distance matrix; in particular, $\mathbf{Z}$ can be seen as a composition of a projection and an orthogonal transformation of $\mathbf{X}$. □

However, when the data $\mathcal{S}$ lie on a curved d-dimensional manifold in $\mathbb{R}^D$, its MDS embedding is rarely informative (in particular, it may be not injective) due to the fact that Euclidean distances between data points completely ignore the manifold structure.

Isomap Algorithm

The Isomap algorithm [180] for finding a low-dimensional, quasi-isometric embedding of a point set S consists of three steps:

1. Construct a proximity graph G over the point set S;
2. Evaluate pairwise geodesic distances between elements of $\mathcal{S}$ via Dijkstra's algorithm on G;
3. Perform MDS on the resulting distances to find a quasi-isometric d-dimensional embedding.

Proximity Graph. A graph is first constructed by creating undirected edges between neighboring input points. Two simple ways have been proposed to define whether two points should be connected by an edge of the neighborhood graph: the first (k-nearest neighbor, or k-NN) approach declares two points neighbors iff one is

among the k nearest neighbors of the other based on Euclidean distances in $\mathbb{R}^D$; the second (ϵ-ball) approach declares two points neighbors iff the Euclidean distance between them is smaller than a user-defined threshold ϵ. Both constructions can be done efficiently (typically, in $O(n \log n)$) using a locality sensitive hashing data structure for instance [59].

Geodesic Distances. After setting edge weights to the Euclidean distances between corresponding pairs of points, the next step is to run Dijkstra's algorithm on the resulting weighted graph G to compute approximations of all pairwise geodesic distances between points of $\mathcal{S}$ in $O(n^2 \log n)$. The squares of these pairwise geodesic distances are then assembled into an $n \times n$ symmetric matrix $\mathbf{D}$.

Multidimensional Scaling Embedding. Finally, the (classical) MDS procedure is performed on $\mathbf{D}$ to obtain the low-dimensional embedding that preserves these squared distances as well as possible in $O(n^3)$.

In spite of the higher computational complexity compared to local methods reviewed earlier, Isomap remains one of the most popular nonlinear dimensionality reduction approaches. Its reliance on a proximity graph to measure disparity between points allows it to handle data sampling curved manifolds, and the use of *all* pairwise approximate geodesic distances makes it particularly stable in the presence of noise.

However, Isomap can produce severely distorted embeddings in the case of irregular sampling and large sampling voids. These issues are associated with the errors arising from the use of Dijkstra-based shortest polylines to approximate true geodesics: this approximation retains good quality only if the point set $\mathcal{S}$ forms a fairly dense sampling of a geodesically convex manifold patch [180].

This problem will be addressed in Ch. 4 with our construction of Parallel Transport Unfolding, a global quasi-isometric manifold learning method inspired by Isomap, where geodesic distances are approximated through parallel transport.

2.1.4 Geometry processing perspective

Computing embeddings of discrete manifolds is one of the most general geometry processing tasks. Surface parameterization, for instance, seeks to embed a three-dimensional triangulated surface into the plane while minimizing some form of distortion (see [66, 103, 130, 160]); mesh deformation is another example where a different embedding of a surface or volume is sought after through minimizing

distortion while satisfying user-specified constraints (see [29, 97, 104, 170, 173, 205]). While geometry processing has been mostly focusing on 3D datasets, its techniques can be leveraged in the context of manifold learning, as the latter seeks to find a low-distortion parametrization of the low-dimensional manifold embedded in high-dimensional space. This link between geometry processing and dimensionality reduction has even been exploited for parameterization [43, 160, 206] and other graphics applications [201].

In Ch. 3 and 4, we bring discrete differential geometry concepts to bear on the the inherently geometrical problem of manifold learning and demonstrate that our approaches offer significant improvements over existing local and global methods.

2.2 Multiresolution Analysis

The relevance of wavelets to the representation of integral and differential operators emerged very early on in the development of multiresolution analysis (MRA), a framework for analyzing a given space through a construction of a hierarchy of nested subspaces and their complements [15, 19, 21, 44, 49, 55–57, 75, 127, 185]. Wavelets have not only been used for fast inversion of a given operator [81, 177] and analysis of corresponding solutions [80, 163], but also for denoising [70, 72] and operator compression [20, 58, 85].

2.2.1 First and Second Generation Wavelets

While first generation adaptive wavelets (such as bi-orthogonal wavelets [48]) can be constructed with arbitrarily high regularity, their shift and scale invariance prevents their adaptation to irregular domains or non-homogeneous coefficients. This problem has stimulated the emergence of second generation wavelets [39, 122, 178, 179, 187] offering stronger adaptability. While much more versatile, these wavelets have found greater adoption in signal processing than in other computational fields.

2.2.2 Wavelets and Finite Element Method

Multiresolution analysis consists in considering a family of nested spaces $\{\mathcal{V}^k\}_{k=1}^{q}$ (such that $\mathcal{V}^{k-1} \subset \mathcal{V}^k$ for $k = 2, \ldots, q$) that approximate the solution space of a given operator, together with their L^2-orthogonal complements $\{\mathcal{W}^k\}_{k=1}^{q-1}$ (defined through $\mathcal{V}^{k+1} = \mathcal{V}^k \oplus \mathcal{W}^k$ for $k = 1, \ldots, q-1$), in order to construct the following decomposition:

$$\mathcal{V}^q = \mathcal{V}^1 \oplus \mathcal{W}^1 \oplus \mathcal{W}^2 \oplus \ldots \mathcal{W}^{q-2} \oplus \mathcal{W}^{q-1}. \tag{2.1}$$

Compactly supported basis elements of the spaces $\{\mathcal{W}^k\}_{k=1}^{q-1}$ and $\{\mathcal{V}^k\}_{k=1}^{q}$ are referred to as *wavelets* and *basis* (or scaling) *functions* respectively. While the L^2-orthogonal decomposition in Eq. 2.1 has been proven particularly useful for signal processing and data compression, using such wavelet basis in FEM framework to solve differential equations is not substantially better than the classical Galerkin method: indeed, the wavelet-Galerkin stiffness matrix (defined as the Gram matrix of all wavelets and coarsest scaling functions with respect to the energy scalar product) is polluted with cross-terms from different resolution levels and becomes ill-conditioned for fine meshes. Consequently, the FEM linear system represented in the wavelet basis is difficult to solve efficiently and reliably in practice. We will review the wavelet-Galerkin approach more thoroughly in Ch. 5 to illustrate the issues with L^2-orthogonal decomposition in detail, and show how these problems can be remediated with an operator-adapted multiresoltion decomposition.

2.2.3 The Need for Operator-adapted Wavelets

In the context of numerical simulation, a consensus has now emerged that the ideal notion of "operator-adapted wavelets" should have three properties: they should be scale orthogonal with respect to the energy scalar product of the operator, the operator should be well-conditioned within each sub-band defined by those wavelets, and they should be localized in space. These properties are particularly desirable when wavelets are used as a basis of finite elements for the Galerkin discretization of the operator as the resulting stiffness matrix becomes block diagonal, with uniformly well-conditioned and sparse blocks. Furthermore, these properties also imply that the wavelets can be used as Wannier functions as well [112, 125, 193] given their dual localization in space and eigenspace. Recently, scalar-valued operator-adapted wavelets satisfying these three properties for linear symmetric positive definite scalar differential operators have been formulated in [138, 140, 141, 158]. These so-called "gamblets" were derived from a game theoretic approach to numerical approximation [139], and were shown to be efficiently computable in a fine-to-coarse fashion in quasilinear complexity. Gamblets have been successfully applied to develop fast solvers for parabolic and hyperbolic PDEs [141], and fast solvers for dense kernel matrices through modified Cholesky decomposition [158], as well as to perform efficient PDE denoising [200] and eigenpairs computation [197].

In Ch. 5, the framework of operator-adapted multiresolution analysis will be extended to incorporate spaces of discrete differential forms, leading to efficient and practical construction of vector-valued operator-adapted wavelets.

Chapter 3

SPECTRAL AFFINE-KERNEL EMBEDDING

3.1 Contributions at a Glance

In this chapter, we propose a controllable embedding method for high- and low-dimensional geometry processing. Our spectral approach leverages simple and efficient geometry processing techniques and can not only perform nonlinear dimensionality reduction on big data, but also offers a nonlinear shape editing tool for 3D meshes and pointsets when applied to 3D data.

Our method is formulated through minimization of a positive semidefinite "multi-Laplacian" quadratic form that is assembled from local linear operators whose kernels only contain affine functions. This optimization problem has a closed form solution based on partial eigendecomposition of the sparse matrix associated with the quadratic form. The optimal solution provides an embedding that best preserves all relative positions of points with respect to each other within their local neighborhoods.

We propose a three-step approach to the embedding problem: a) we first parameterize each local neighborhood from $\mathbb{R}^D$ into $\mathbb{R}^d$ as isometrically as possible via Isomap, using an additional geometric correction of geodesics to improve robustness to irregular sampling; b) we then compute an exhaustive set of relative coordinates that captures the position of every point with respect to its local neighborhood; c) lastly, a global embedding is found as the new point positions in $\mathbb{R}^d$ that best preserve all the relative coordinates from all the neighborhoods. Each of these three steps is efficiently formulated as a spectral problem: (a) requires independent (thus, trivially parallelizable) partial eigendecompositions of dense, but small matrices; (b) is done via independent partial SVDs; and (c) is achieved via a partial eigendecomposition of a sparse, positive semidefinite symmetric matrix. Our approach can be understood in geometric terms as constructing the global embedding coordinates that are as linear as possible in the local most-isometric parameterization of each small neighborhood of the original input. Note that this is in marked contrast with most previous methods which fail to properly unfurl even flat or developable manifolds (see Figs. 3.2 and 3.3). Finally, we explore applications of manifold learning to 3D geometry processing by formulating a spectral, as-rigid-as-possible shape

deformation technique that computes the most-isometric embedding of an original shape based on user-defined position constraints.

Besides our novel embedding approach, we also provide along the way a series of contributions covering various aspects of manifold learning and shape deformation:

- We introduce a local geodesic curvature correction to the distances used in Isomap [180] that adds much improved stability to this staple of nonlinear dimensionality reduction in the case of irregular and sparse sampling, without affecting its computational complexity or adding new parameters;
- We add robustness to Locally-Linear Embeddings [155] and Laplacian Eigenmaps [17] by using an exhaustive set of relative coordinates to guarantee the absence of spurious harmonic deformation in the final embedding. Our extension is much simpler and more robust to irregular sampling and noise than Hessian-LLE [71], which unnecessarily requires quadratic accuracy of their local Hessian operators;
- We introduce a simple approach for user-guided deformation of meshes and pointsets which, unlike previous nonlinear editing approaches [29, 173], can be efficiently and reliably computed through eigenanalysis of a sparse matrix.

3.2 SAKE Algorithm

Using the notation and assumptions of Sec. 2.1.1, we introduce a procedure to map input pointset $\mathcal{S}$ sampling a d-manifold $\mathcal{M}$ from $\mathbb{R}^D$ into $\mathbb{R}^d$, where $D > d$. As before, we assume $\mathcal{S}$ to be arbitrarily indexed, and use a $D \times n$ matrix $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ and a $d \times n$ matrix $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_n)$ to denote the coordinates of input points and the final embedding respectively.

Our dimensionality reduction proceeds in four distinct steps:

- We first form a proximity graph G by linking nearby input points; we then assemble a local geodesic neighborhood $\mathcal{N}(i)$ of each input point $\mathbf{x}_i$ based on the proximity graph.
- We compute for each $\mathbf{x}_i$ an as-isometric-as-possible embedding of its neighborhood in $\mathbb{R}^d$.
- For each of these resulting embeddings, we assemble a sparse matrix $\mathbf{L}_i$ representing a linear operator ("multi-Laplacian") whose kernel is restricted to constant and linear functions.

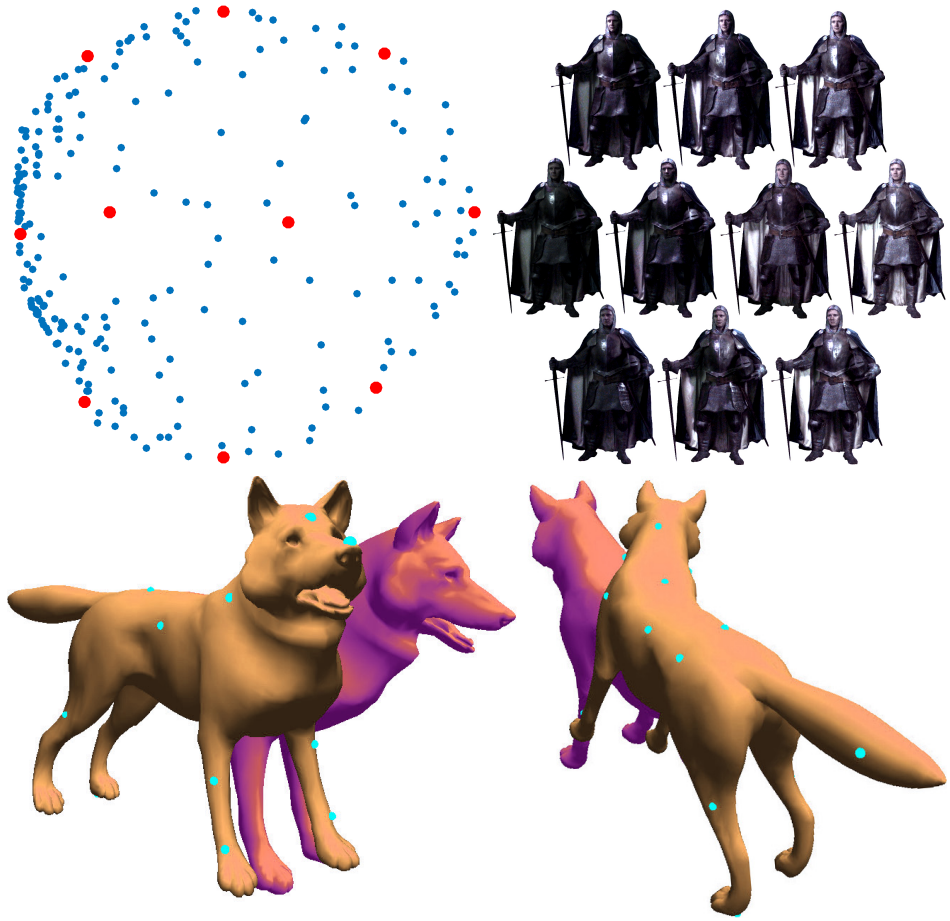


Figure 3.1: **SAKE: High- and Low-dimensional Geometry Processing.** We introduce a new technique to perform controllable embedding through eigenanalysis. Our approach allows us to find structure in high dimensional datasets: (Top) from 221 RGB images (with 608×456 pixels; 10 are shown on the right) of an actor in a knight costume captured from different lighting directions covering a large sphere of illumination [117], a 2D embedding is computed solely based on local pixel differences (left). Our Spectral Affine-Kernel Embedding method finds a 2D parameterization of the images corresponding to the direction and the intensity of the lighting (the knight images correspond to red dots). (Bottom) The same spectral embedding approach can also be used for user-guided shape editing of 3D meshes and pointsets, where a few handles are moved to precisely control the deformation of an initial object through a simple sparse matrix eigenanalysis.

- We then assemble a quadratic form $\mathbf{Q}$ derived from the affine-kernel matrices $\mathbf{L}_i$, and find the final embedding $\{\mathbf{z}_i\}_{i=1}^n$ in $\mathbb{R}^d$ by computing the lowest $(d+1)$ eigenvectors of $\mathbf{Q}$.

We now review each step in order to provide both algorithmic details and mathematical justification for our embedding approach.

3.2.1 Proximity Graph and Geodesic Neighborhoods

We begin by forming a *proximity graph* G by linking every point of $\mathcal{S}$ with its k nearest neighbors based of the Euclidean distance in $\mathbb{R}^D$, found efficiently through a kd-tree, cover tree, or locality sensitive hashing data structure. The value of k must be small to make sure the edges of the graph are short enough to offer reliable approximations of geodesics. Knowing that we will approximate a d-dimensional manifold, we typically choose $k = 4d$, i.e., a valence proportional to d like for a regular grid. We then assign to each point $\mathbf{x}_i$ the set of indices $\mathcal{N}(i)$ corresponding to the K nearest neighbors of the ith point in the proximity graph—hence, defining a *geodesic neighborhood*—for K larger than k. By default, we pick K in the interval $[k; 2^d k]$ to account for the dimensionality of the data and the amount of noise. These K-neighborhoods define n overlapping patches that will be unfolded into $\mathbb{R}^d$, and then glued together into a global d-dimensional embedding.

Note that our choice of a small value k for the proximity graph and a distinct, larger value K to define neighborhoods prevents the traditional issue of "shortcutting" the manifold: existing methods do not make the distinction between these two values, and pick a neighborhood based on *either* an ϵ-ball around the point *or* its k-nearest-neighbors in $\mathbb{R}^D$. However, robustness to noise requires a large number of neighbors, which creates graph edges that are far from being geodesics. Instead, our simple alternative allows the use of large neighborhoods (to be robust to noise), while still keeping a sparse edge graph to better approximate geodesic distances. Of course, any additional knowledge on the sampling (such as noise level) can be used to adjust the two parameters k and K. These parameters can even vary from point to point if needed to better deal with varying sampling density for instance, although our

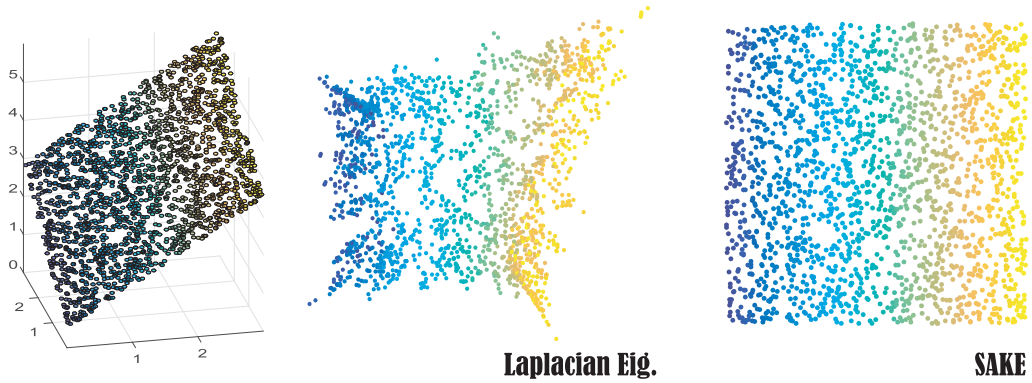


Figure 3.2: **Affine Precision.** Laplacian Eigenmap (middle) is not linearly precise, as it fails to properly capture a uniformly sampled flat patch in 3D (left). SAKE (right) does not suffer from this common limitation.

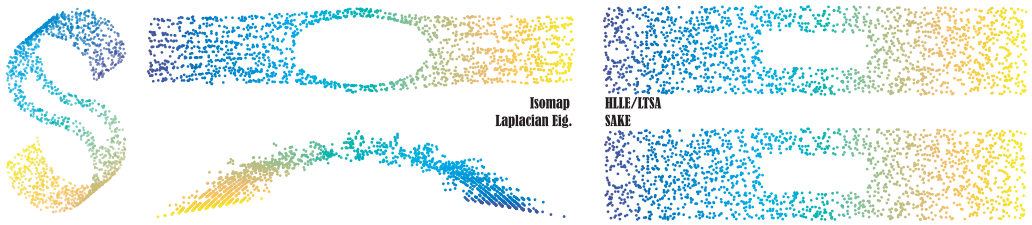


Figure 3.3: **Non-simply Connected Manifold.** For points uniformly distributed over a 2-dimensional S-shaped manifold containing a hole (left), Isomap (top middle) overestimates the size of the hole, while Laplacian Eigenmaps result (bottom middle) is severely distorted. Note that HLLE with $k = 12$ (top right) and Sake with $k = K = 12$ (bottom right) recover close to perfect embeddings in this noiseless case.

strategy can already handle large density variations (see Fig. 3.10).

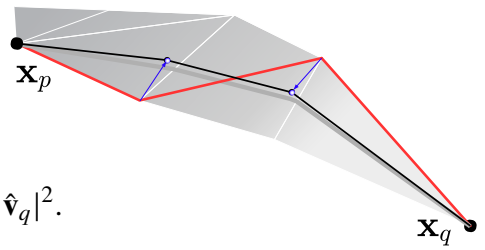
3.2.2 Mapping Neighborhoods into Reduced Space

Each geodesic neighborhood is then mapped as isometrically as possible into dimension d. Given that these neighborhoods are small and contractible, we use Isomap [180] to achieve the mapping reliably: using PCA-based projection onto local tangent space instead would be significantly less isometric if the input manifold is curved.

Geodesic Distances via Dijkstra's Algorithm. Each edge of the proximity graph G is considered an intrinsic geodesic curve, and its Euclidean length in $\mathbb{R}^D$ is assumed to be an accurate estimate of the intrinsic distance between its end points. For every pair of points in the geodesic neighborhood of point i that are not directly connected by an edge, we compute its approximate geodesic distance by solving the all-pairs-shortest-path problem, which Dijkstra's algorithm [69] achieves optimally in $O(K^2 \log K)$.

Geodesic Curvature Correction. Shortest paths computed on the neighborhood graph suffer from non-zero geodesic curvatures: Dijkstra's algorithm will return shortest paths that are polylines made of graph edges, which are very unlikely to be the actual geodesics. To improve geodesic distance estimates, we correct for these graph-induced errors by post-processing each shortest path to remove spurious geodesic curvatures; more precisely, we construct an improved shortest path by shifting its vertices parallel to the neighborhood's local tangent space so as to eliminate its geodesic curvature. To this end, we first compute an estimate of the local tangent d-dimensional space of the neighborhood through PCA, which returns a d-dimensional basis of orthonormal vectors $\mathbf{t}_1, \ldots \mathbf{t}_d$ in $\mathbb{R}^D$. Then for each "geodesic" polyline be-

tween two points $\mathbf{x}_p$ and $\mathbf{x}_q$ (representing their shortest connecting path in graph G), we displace the intermediate vertices of the polyline parallel to the d-dimensional tangent space and orthogonal to the line $(\mathbf{x}_p\mathbf{x}_q)$ to project out any "zigzagging", thus straightening the geodesic. This is easily achieved numerically: if $\mathbf{x}_r$ is a vertex of the polyline, first project the $\mathbb{R}^D$ vectors $\mathbf{v}_r = \mathbf{x}_r - \mathbf{x}_\mathbf{p}$ onto the tangent d-dimensional space to obtain $\hat{\mathbf{v}}_r = \sum_j (\mathbf{t}_j \cdot \mathbf{v}_r)\ \mathbf{t}_j$; from the resulting vectors $\hat{\mathbf{v}}_r$, further extract their projection along the direction $\hat{\mathbf{v}}_q$ through



$$\tilde{\mathbf{v}}_r = \hat{\mathbf{v}}_r - \left[\hat{\mathbf{v}}_r \cdot \hat{\mathbf{v}}_q\right] \hat{\mathbf{v}}_q / |\hat{\mathbf{v}}_q|^2.$$

These final (tangent) vectors $\tilde{\mathbf{v}}_r$ of the points along the path between $\mathbf{x}_p$ and $\mathbf{x}_q$ are then subtracted from their corresponding original vertices $\mathbf{x}_r$ to determine their improved locations. The length of the resulting polyline (see inset), which has now close to zero geodesic curvature, is a more accurate geodesic distance l_{pq} between $\mathbf{x}_p$ and $\mathbf{x}_q$. Note that this correction guarantees *exact* distance evaluations when the neighborhood is flat, and provides a robust and consistent estimate of distances in the general case of a curved manifold—even applied on a well-shaped triangle surface (where the graph is now formed by the edges of the mesh), our simple projection reduces the relative geodesic length errors fivefold as shown in Fig. 3.4. The only assumption our approach makes is that the PCA-based estimate of the local tangent

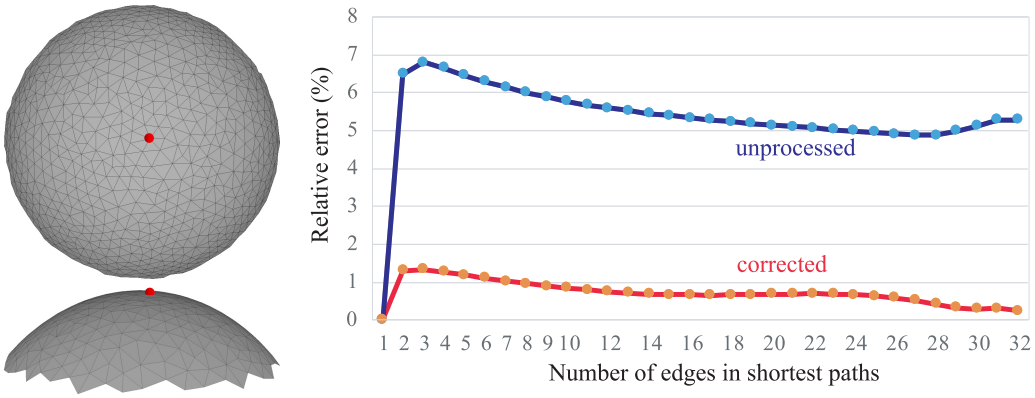


Figure 3.4: **Geodesic Distance Correction.** From a triangulated patch of a spherical cap with good aspect ratio elements (left, top, and side views), the average relative error of pairwise distances computed based on Dijkstra's shortest paths is 5.6%; our simple post-processing reduces the error to 0.8%. If we bin all the distances based on the number of edges in the edge-based connecting paths, we see that aside from the case of a single edge (where the geodesic distance cannot be improved), our geodesic distance correction reduces errors by a factor of five or more.

Figure 3.5: **Geodesic Curvature Correction.** Our geometric correction improves geodesic approximations for coarse and irregularly sampled data; without it, extreme distortion can happen on imperfectly sampled datasets. We set $k = K = 12$ for this noiseless Swiss Roll example (left).

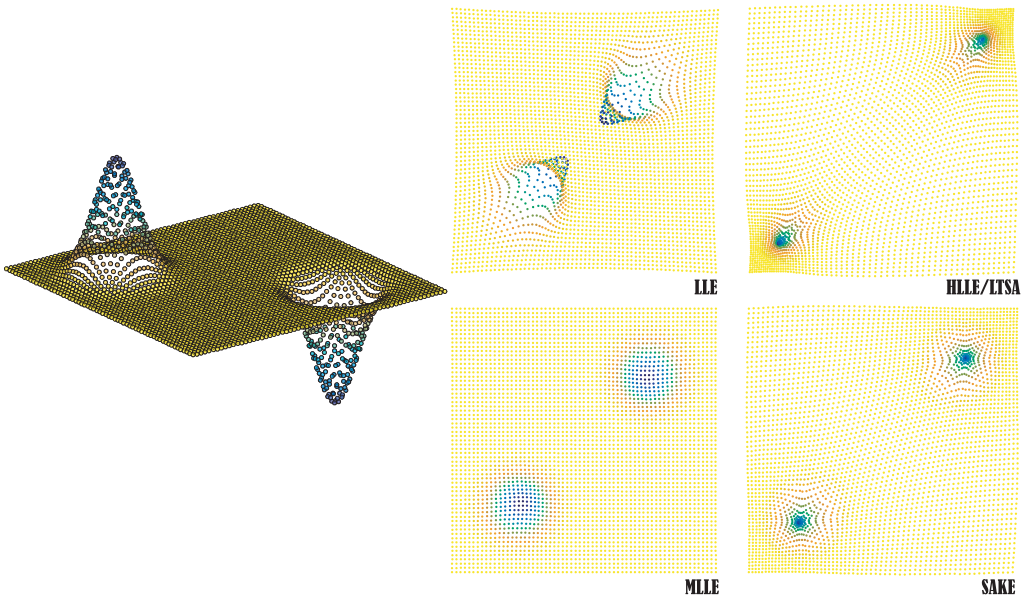


Figure 3.6: **Gaussian Landscape.** For a regularly-sampled "two-bump" surface (left), LLE creates foldovers; HLLE/LTSA both significantly distort areas around the bumps in a non-isometric way; MLLE recovers an almost regular grid, completely ignoring the local curvature of the original data; $k = 24$ neighbors were used for all the methods. Instead, SAKE ($k = 8, K = 24$) finds a most-isometric embedding.

space on a neighborhood of size K is of reasonable quality. As a result, if the neighborhoods are chosen to be too small relative to the noise present in the data or too large compared to data features, the quality of the correction may deteriorate.

From Distances to a d-dimensional Embedding. For all the pairwise geodesic distances l_{pq} (for p and q in $i \cup \mathcal{N}(i)$) of the local neighborhood of point $\mathbf{x}_i$, we apply MDS (see Sec.2.1.3) to obtain a lower dimensional flat embedding of a neighborhood

that best preserves the geodesic distances. Note that if the points are lying on a flat manifold, no distortion occurs in this process thanks to our correction step—a property that regular Isomap fails to enforce due to the zigzagging of Dijkstra's shortest paths.

Discussion. Pairwise distances on triangle or tetrahedral meshes can of course be computed faster and better through fast marching [206] or heat-based distances [53]. Diffusion distances have also been proposed to offer approximate geodesic distances through truncation of the graph Laplacian spectrum and tuning of a diffusion time t [50]. However, our approach applies to pointsets in *arbitrary* dimensions, and its simplicity makes it particularly convenient: improvements over original Dijkstra distances are already significant with only a small computational overhead.

3.2.3 Relative Coordinates

Now that we have unfurled all geodesic neighborhoods in d-dimensions, we assemble for each of these mapped neighborhoods a set of linear equations that represent all possible relative coordinates of a point with respect to its neighbors. Equivalently, we will show that these equations enforce harmonicity in all constant metrics. Since the neighborhood of point $\mathbf{x}_i \in \mathbb{R}^D$ has been nearly-isometrically mapped to points $\mathbf{y}_j \in \mathbb{R}^d$ for each j in the index set $\{i\} \cup \mathcal{N}(i)$, we will denote by $\mathbf{Y}_i$ the $(K+1)\times d$ matrix containing all the coordinates of the neighborhood points, i.e.,

$$\mathbf{Y}_i = \begin{pmatrix} \mathbf{y}_i & \mathbf{y}_{j_1} & \mathbf{y}_{j_2} & \dots & \mathbf{y}_{j_K} \end{pmatrix}^T.$$

Affine-precise Linear Combinations. Define the $d\times K$ matrix

$$\mathbf{E}_i = \begin{pmatrix} \mathbf{y}_{j_1}-\mathbf{y}_i; & \mathbf{y}_{j_2}-\mathbf{y}_i; & \dots & ; & \mathbf{y}_{j_K}-\mathbf{y}_i \end{pmatrix}$$

containing in the m-th column the d coordinates of the edge vector $\mathbf{y}_{j_m} - \mathbf{y}_i$. This matrix can be thought of as a redundant basis of the d-dimensional tangent space, and its kernel is formed by the space of all linear combinations of edges summing to zero. Because we picked $K > d$ (i.e., more neighbors than the dimensionality), the rank-nullity theorem directly implies that the size of this kernel is $K-d$, as the edges span the entire d dimensions in practice. The basis of this kernel can be constructed efficiently through a Singular Value Decomposition (SVD) of $\mathbf{E}_i = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{U}$ is a $d\times d$ orthogonal matrix, $\mathbf{\Lambda}$ is a $d\times K$ rectangular diagonal matrix with the d singular values in decreasing magnitude on the diagonal, and $\mathbf{V}$ is a $K\times K$ orthogonal matrix. We simply select the last $K-d$ right singular vectors $\{\mathbf{w}^p \in \mathbb{R}^K\}_{p=1..K-d}$

of unit length as the basis vectors of the kernel. Now $\mathbf{y}_i$ can be written as a linear combination of its neighbors $\mathbf{y}_j$ for each of these vectors $\mathbf{w}^p$ since, by construction,

$$\forall p \in [1..K-d], \quad \left(\sum_{j \in \mathcal{N}(i)} [\mathbf{w}^p]_j \right) \mathbf{y}_i = \sum_{j \in \mathcal{N}(i)} [\mathbf{w}^p]_j \; \mathbf{y}_j, \tag{3.1}$$

where $[\mathbf{w}^p]_j$ denotes the j-th coordinate of $\mathbf{w}^p$. Consequently, the matrix $\mathbf{L}_i$ of size $(K-d)\times(K+1)$ defined as

$$\mathbf{L}_i = \begin{pmatrix} \sum\limits_{j \in \mathcal{N}(i)} [\mathbf{w}^1]_j & \sum\limits_{j \in \mathcal{N}(i)} [\mathbf{w}^2]_j & \dots & \sum\limits_{j \in \mathcal{N}(i)} [\mathbf{w}^{K-d}]_j \\ -\mathbf{w}^1 & -\mathbf{w}^2 & \dots & -\mathbf{w}^{K-d} \end{pmatrix}^T$$

satisfies $\mathbf{L}_i \; \mathbf{Y}_i = \mathbf{0}$. Note that the set of weights $\mathbf{w}^p$ can be seen as a linearly-independent basis of all relative coordinates, describing the position of $\mathbf{y}_i$ in terms of its neighbors $\mathbf{y}_j$ for $j \in \mathcal{N}(i)$. Using the *whole space of relative coordinates* instead of picking just one results in a more complete encoding of the data, allowing us to capture the geometry of a patch in a reliable way.

Affine Kernel. Note that matrix $\mathbf{L}_i$ has an important property: its kernel consists of all (discrete) affine functions. Indeed, consider a scalar function $f : \mathbb{R}^d \to \mathbb{R}$ and call $\mathbf{f}$ the $(K+1)$-dimensional column vector representing the sampling of f in the neighborhood, i.e. $\mathbf{f}_j = f(\mathbf{y}_j)$ for $j \in \{i\} \cup \mathcal{N}(i)$. Due to our construction of $\mathbf{L}_i$, any constant function will satisfy $\mathbf{L}_i \mathbf{f} = \mathbf{0}$. The same property holds for linear functions as well because the weights form a basis of the kernel of $\mathbf{E}_i$ (Eq. (3.1)). Thus, Ker $\mathbf{L}_i$ contains *all* sampled affine functions. Furthermore, the rows of $\mathbf{L}_i$ are $(K-d)$ independent vectors (by construction via the SVD), so by the rank-nullity theorem the kernel of $\mathbf{L}_i$ is of dimension $(K+1)-(K-d) = d+1$. However, the space of affine functions $f(\mathbf{y}) = \mathbf{a}^T\mathbf{y} + b$ ($a \in \mathbb{R}^d, b \in \mathbb{R}$) also has dimension $d+1$. This implies that sampled affine functions are the *only* elements of $\operatorname{Ker} \mathbf{L}_i$.

Multi-Laplacian Interpretation. Note that our construction can be understood as an extension of Laplacian eigenmaps [17]: while their approach uses a single linear equation per neighborhood corresponding to a local condition of harmonicity, we have instead a whole set of linear equations. Each row of our local operator $\mathbf{L}_i$ can be interpreted as a discrete harmonicity condition in a different metric—hence our use of the term "multi-Laplacian." Indeed, the action of a row of $\mathbf{L}_i$ on a discrete function $\mathbf{f}$ is of the form $d \star d\, \mathbf{f}$ in the DEC notation [68], and equating it to zero corresponds to a harmonic condition in a possibly non-Euclidean metric. The matrix $\mathbf{L}_i$, in fact, encodes harmonicity conditions for *all constant metrics* σ over the

unfolded neighborhood. The anisotropic Laplacian operator $\Delta_\sigma f = \nabla\cdot(\sigma\nabla f)$ returns zero for affine functions as discussed in [60, 61], since constant metrics σ satisfy $\nabla \cdot (\sigma\nabla f) = \mathrm{Tr}(\sigma\ \mathrm{Hess}\ f)$. (Linear accuracy is actually valid for all divergence-free metrics, but the restricted case of constant metrics is sufficient for our purpose.) Therefore, our construction can be thought of as identifying the intersection of the spaces of σ-harmonic functions for all constant σ: these "multi-harmonicity conditions" restrict the kernel to be sampled affine functions only—while the normal Laplacian is blind to any non-zero off-diagonal term of the Hessian. Note that the *multi-Laplacian* $\mathbf{L}_i$ we end up with is less difficult to assemble than a Hessian operator on which Hessian-LLE is based, since it requires no specific behavior when applied to sampled quadratic functions; yet the two operators have the exact same kernel, bringing robustness to the process as the authors of [71] argued.

3.2.4 Quadratic Form

In order to find the final embedding $\{\mathbf{z}_i\}_{i=1}^n$ in $\mathbb{R}^d$ of the pointset $\mathcal{S}$, we assemble a sparse, symmetric, and positive semidefinite $n\times n$ matrix $\mathbf{Q}$. Let $\mathbf{S}_i$ be the $(K+1)\times n$ selection matrix of neighborhood i, i.e., the sparse matrix such that each component (p, q) is 1 if $q \in \mathcal{N}(p)$, and 0 otherwise—so that $\mathbf{S}_i\mathbf{Z}^T$ is the mapped neighborhood of i in the final d-dimensional embedding. We then define our global quadratic form $\mathbf{Q}$ as

$$\mathbf{Q} = \sum_{i=1}^{n} \mathbf{S}_i^T\ \mathbf{L}_i^T\ \mathbf{L}_i\ \mathbf{S}_i.$$

Given our interpretation of the matrices $\mathbf{L}_i$ as storing harmonicity conditions for all constant metrics, one can understand the global quadratic form as the sum of local Dirichlet energies computed in all locally constant metrics. The resulting quadratic form thus penalizes any non σ-harmonic functions—that is, any non-affine functions— in any given neighborhood: it is therefore much more "discerning" than a simple Laplacian.

3.2.5 Final Embedding

From the sparse SPD matrix $\mathbf{Q}$, we extract the final, global embedding of the input pointset as a set of positions $\{\mathbf{z}_i\}_{i=1}^n$ in $\mathbb{R}^d$ by computing the first $(d+1)$ smallest eigenvectors $\mathbf{q}_m$ of $\mathbf{Q}$ (satisfying $\mathbf{Q}\,\mathbf{q}_m = \lambda_m\mathbf{q}_m$). Note that the first eigenvector is constant and corresponds to zero eigenvalue by construction. The d coordinates of points $\mathbf{z}_i$ simply correspond to the second, third, ..., and $(d+1)$st smallest d eigenvectors:

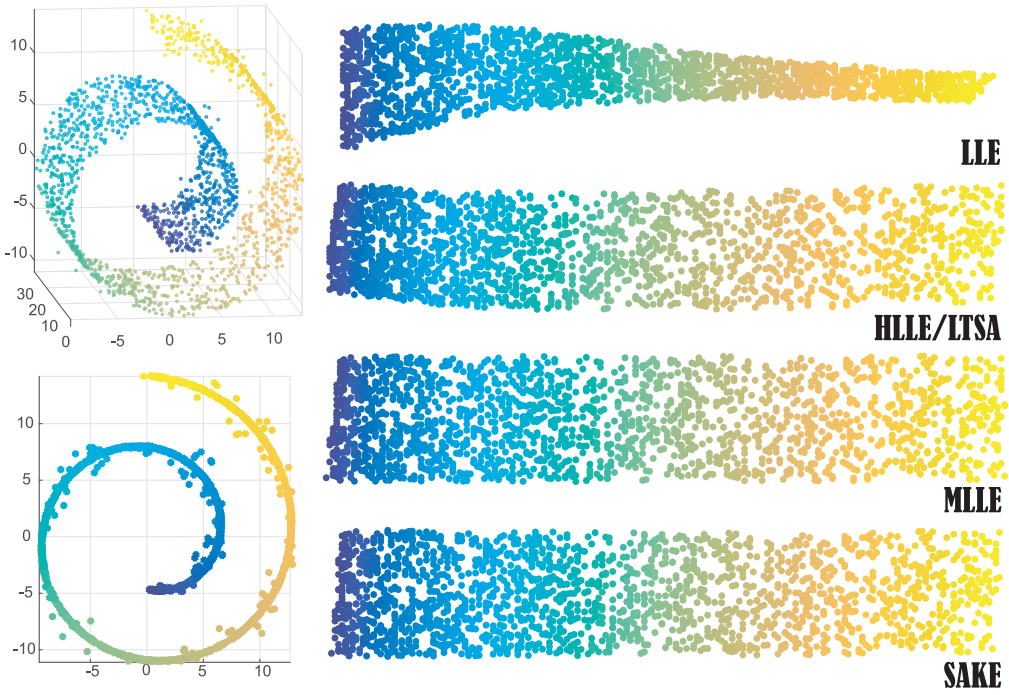


Figure 3.7: **Swiss Roll with Sparse Noise.** Flexibility in the size of geodesic neighborhoods renders SAKE stable to noise. We use points sampled from an intrinsically 2D Swiss roll embedded in 3D with sparse, uniformly-distributed noise in the normal direction (left, 2 views). Results of nonlinear 2D embeddings (right column, from top to bottom): LLE ($k = 15$), HLLE/LTSA ($k = 15$), MLLE ($k = 15$), and SAKE ($k = 15, K = 30$). Results of LTSA and HLLE are visually indistinguishable on this example.

$$\mathbf{z}_i = \begin{pmatrix} [\mathbf{q}_2]_i & [\mathbf{q}_3]_i & \dots & [\mathbf{q}_{d+1}]_i \end{pmatrix}^T.$$

Indeed, these are the positions that make every local neighborhood satisfy the multi-harmonic conditions as closely as possible, in a least squares sense, since the matrix $\mathbf{Z} = (\mathbf{z}_1\ \mathbf{z}_2\ \dots\ \mathbf{z}_n)$ is the solution of the following minimization:

$$\underset{\mathbf{Z}\in\mathbb{R}^{d\times n}}{\arg\min}\ \mathrm{Tr}\left[\mathbf{Z}\mathbf{Q}\mathbf{Z}^T\right] \quad \text{s.t.}\ \ \mathbf{Z}\mathbf{Z}^T = \mathbf{Id},\ \ \mathbf{Z}\mathbf{1} = \mathbf{0}$$

where the constraints are used to prevent degenerate solutions (similar to the use of Fiedler vectors in [130]). The final embedding thus preserves the local relative coordinates as well as possible since, by construction of $\mathbf{w}^p$, we have $\mathbf{L}_i\mathbf{Y}_i = 0$ over all the isometrically parametrized patches of the original manifold.

Proposition 3.1. *SAKE is "linearly precise" up to a linear transformation: assuming the pointset $\mathcal{S}$ samples a linear d-dimensional subspace $\mathcal{R}$ of $\mathbb{R}^D$, and the proximity graph G is such that each sample point has enough neighbors to span a*

d-dimensional subspace, the SAKE embedding $\mathbf{Z} \in \mathbb{R}^{d \times n}$ *is a linear transformation of input data* $\mathbf{X} \in \mathbb{R}^{D \times n}$.

Proof. Because input data sample a linear subspace, our corrected estimates of pairwise geodesic distances are *exact*. Consequently, applying Isomap to local geodesic neighborhoods is equivalent to using classical MDS. By Prop. 2.1, the produced mapping $\mathbf{Y}_i$ is a linear distance preserving transformation of i-th geodesic neighborhood $\mathbf{S}_i\mathbf{X}^T$ for any $1 \leq i \leq n$; as a result, it also preserves all relative positions. This implies that the input data $\mathbf{X} \in \operatorname{Ker} \mathbf{Q}$ by construction ($\mathbf{L}_i\mathbf{Y}_i = \mathbf{L}_i\mathbf{S}_i\mathbf{X}^T = 0$ for any $1 \leq i \leq n$). Since SAKE embedding procedure extracts $(d+1)$ eigenvectors of $\mathbf{Q}$ corresponding to smallest eigenvalues, it parametrizes its null space and effectively introduces an orthogonal coordinate frame for d-dimensional linear subspace $\mathcal{R}$ of $\mathbb{R}^D$ that contains $\mathbf{X}$ (the first eigenvector is ignored). Therefore, $\mathbf{Z}$ represents $\mathbf{X}$ projected onto a different orthogonal coordinate frame and is formally a linear transformation of the latter. □

Notice that if the input pointset is finely sampling a developable manifold, we recover a high quality intrinsic discretization, up to an affine transformation, of the manifold isometrically unfolded in $\mathbb{R}^d$ (the inaccuracies are incurred during local neighborhood unfolding through Isomap); the first $(d+1)$ eigenvalues of $\mathbf{Q}$ are close zero in that case. If the original manifold is not developable, this extraction picks the unfolding which is as affine as possible in each locally isometric coordinates—and the eigenvalues of $\mathbf{Q}$ inform us on how non-developable the initial manifold was, and how much *metric distortion* we can expect from our optimal low-dimensional embedding.

3.3 Analysis and Results

We provide an extensive analysis to clearly identify the properties of SAKE, before presenting numerical tests to confirm our claims. We used the SciKit Learn [162] manifold learning implementation of Isomap, LLE, HLLE, LTSA, and Laplacian Eigenmaps.

3.3.1 Computational Complexity

Once the initial proximity graph is computed (a task that is common to all manifold learning methods), our approach involves four distinct stages. First, we compute an all-pair-shortest-path algorithm per neighborhood so that each approximate geodesic distance between point pairs can be evaluated; this step has a complexity

of $O(K^2 \log K)$ for each of the n neighborhoods. Second, we compute an as-isometric-as-possible parameterization of each neighborhood, which requires the $O(K^3)$ eigenanalysis of a dense $(K+1)\times(K+1)$ matrix of distances. Third, we compute a SVD in $O(K^3)$ to find the affine-kernel matrix $\mathbf{L}_i$ in each neighborhood. Note that these three first steps can be done in a massively parallel way, as they proceed independently on each neighborhood. Finally, finding the $(d+1)$ bottom eigenvectors of the sparse matrix $\mathbf{Q}$ requires an expected $O(n^{1.5})$ number of operations (as already experimentally found in [165] and confirmed in our tests). Assuming that K is small compared to the number of input points, the SAKE embedding algorithm scales much better than Isomap, and does not suffer from any restriction on the geometric nature of the manifold.

3.3.2 Comparison to Prior Methods

Before delving into numerical comparisons, we first detail how our approach markedly differs from previous work. In particular, we explicitly describe the key differences between SAKE and the two most closely related approaches, HLLE and MLLE. In our discussion of the computational complexity of the various existing methods, we will denote by k the average number of neighbors for each input point $\mathbf{x}_i$ to be consistent with our notation.

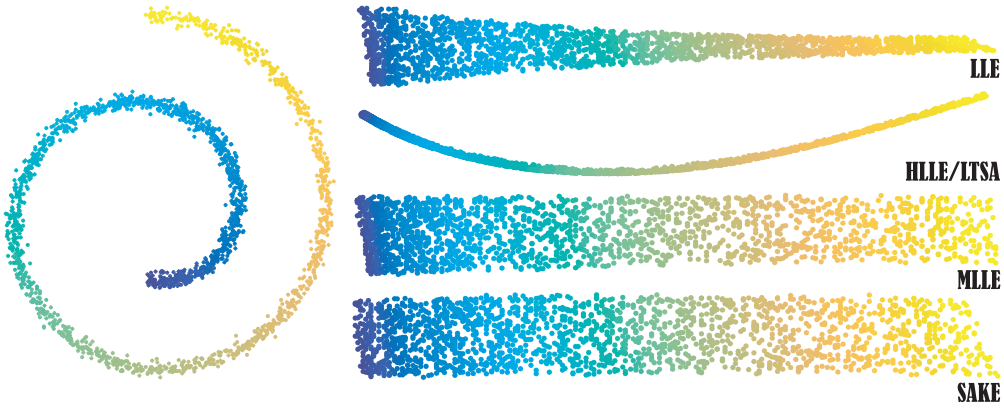


Figure 3.8: **Swiss Roll with Gaussian Noise.** SAKE can handle strong noise in the input. Here, points are sampled from a 2D Swiss roll embedded in 3D with added Gaussian noise along the normal (left, profile view). Results of 2D embeddings (right, top to bottom): LLE (k = 15), HLLE/LTSA (k = 15), MLLE (k = 15), and SAKE (k = 15, K = 75). Again, LTSA and HLLE return visually identical results, significantly deformed due to the instability of PCA-based tangent plane approximation in the presence of strong noise.

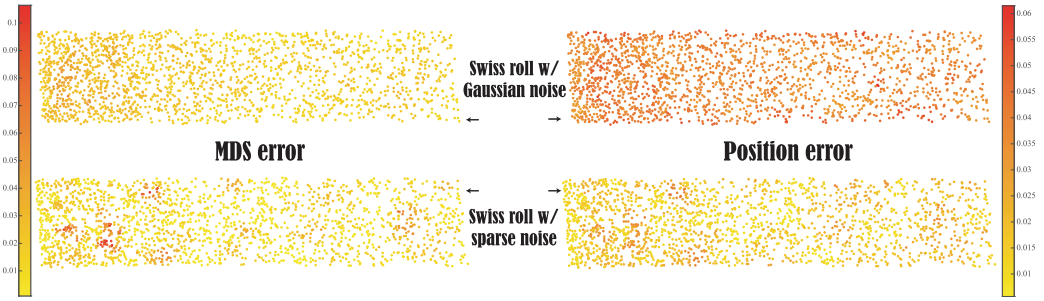


Figure 3.9: **Embedding Errors.** For the SAKE results shown in Fig. 3.8 (top) and Fig. 3.7 (bottom), we show the local reconstruction errors in pairwise distances (MDS errors, left) and in relative l^2 position errors (right).

SAKE vs. Isomap. Besides its issues of computational complexity and stringent limitation to simply connected patches, Isomap is not robust to low sampling: shortest-path geodesic approximations can become quite poor. Despite the existence of proofs of convergence to the proper geodesic distances [63], the local *differences* between local geodesic distances are often very inaccurate. Our improvement through geodesic curvature correction allows for much more robust results; see Figs. 3.4 & 3.5. While we use this distance correction only for local neighborhoods in the context of SAKE, the same simple geometric insight can also enhance the approximation of geodesic distances used in Isomap algorithm by performing this correction over long geodesics through local corrections. Proper testing of this local/global correction of Dijkstra's shortest paths in Isomap is left as future work.

SAKE vs. Laplacian Eigenmaps. The approach of Belkin and Niyogi [17] relies on an eigenanalysis of the Laplacian operator. However, the null space of this operator may contain much more than linear functions: it includes all harmonic functions such as $f(x, y) = xy$ for instance. The final embedding is thus often polluted by harmonic deformations, as observed in practice in Fig. 3.3. SAKE, instead, reduces the kernel of its quadratic form by enforcing more than a single linear equation per vertex, in order to guarantee a better isometry of the final embedding. Moreover, the approximation of the Laplacian in high dimension by a matrix $\mathbf{L}$ with entries L_{ij} proportional to $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ as proposed in Laplacian Eigenmaps does not satisfy linear precision. Consequently, a flat manifold would not even be properly mapped without distortion (Fig. 3.2), while our approach returns perfect parametrization free of nonlinear pollution in this case.

SAKE vs. Locally Linear Embedding. While LLE [155] does not rely on the Laplacian operator, its foundations are quite similar to Laplacian Eigenmaps. The

only difference in practice is that the linear equation assembled per point $\mathbf{x}_i$ is not derived from a local Laplace estimate, but determined by solving a constrained least squares problem to best capture the local linear structure: it computes the linear combination of neighborhood points $\{\mathbf{x}_j\}_{j\in\mathcal{N}(i)}$ that best reconstructs $\mathbf{x}_i$. The first advantage of this modification is that in the case of a flat manifold, this linear equation is exactly satisfied by the input, so the result will be perfect. However, as soon as the input manifold is curved, the lack of isometric unfolding of each neighborhood and the reliance on a single linear equation per point to find the final embedding renders the approach extremely brittle: different samplings of a same manifold may result in dramatically different embeddings; see Figs. 3.7 and 3.8. A follow-up work [87] proposes to first project the neighborhood in d-dimensions through PCA. While this tends to reduce the sensitivity to noise since regularization is no longer needed, this approach remains quite brittle due to the reliance on a single linear equation per point.

SAKE vs. LTSA. Local Tangent Space Alignment [204] proceeds by first constructing an approximation for the tangent space at each data point, before aligning these tangent spaces to form global coordinates. In essence, this approach is similar to ours; but they instead directly enforce that transitions between neighborhood charts be as affine as possible, and the initial construction of the tangent space relies on PCA, which does not handle curved manifolds well. As a consequence, LTSA is quite successful at unfolding nearly flat manifolds (Fig. 3.6) because it also relies on more than a single linear equation per neighborhood (in fact, in many cases, LTSA and HLLE have identical results); but LTSA systematically fails on more challenging examples, as demonstrated in Fig. 3.8.

SAKE vs. Multiple-weights LLE. A variant of the original LLE method, named Multiple-weights LLE [203], is worth discussing further: the authors were the first to notice that the brittleness of LLE-type methods is mostly due to their reliance on the enforcement of a single linear equation per neighborhood. Consequently, they added to the regularized set of weights that LLE uses all the sets of weights corresponding to singular values close enough to zero (requiring a user-specified threshold), which offered much improved stability. However, just like LLE, the authors do not use a local, as-isometric-as-possible embedding, so the kernel of their operator is "polluted" by the way the manifold is embedded in higher dimension. Consequently, the result of their dimensionality reduction is far from being isometric,

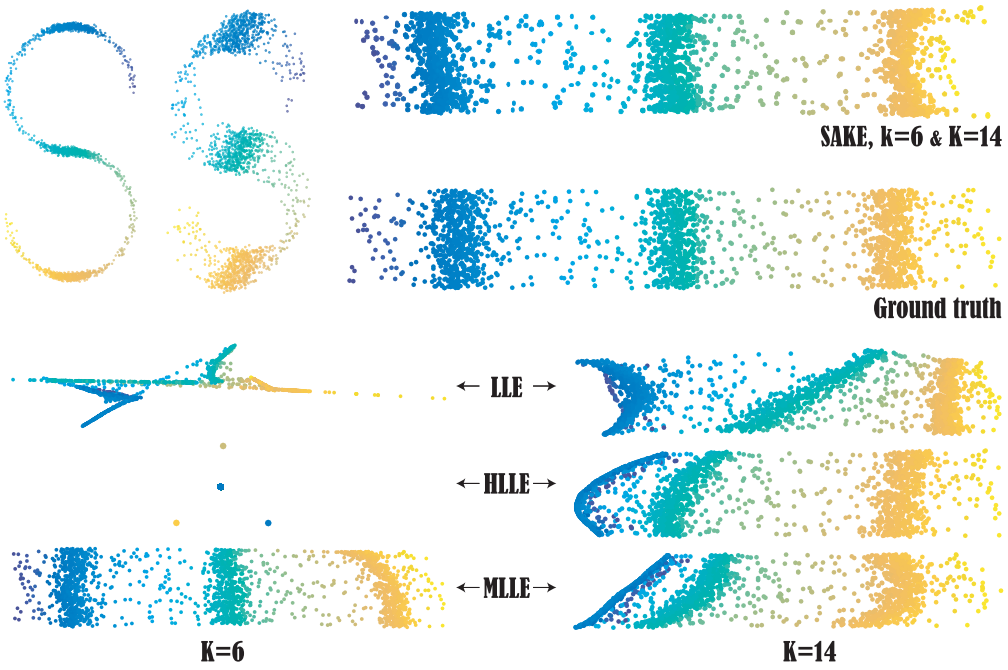


Figure 3.10: **S-shaped Manifold with Variable Density and Gaussian Noise.** SAKE is also robust to variable density in the input. A non-uniformly sampled 'S' shape is embedded in 3D with Gaussian noise added (left, profile views). We compare our results with other existing approaches, for various parameters.

even if the input is nearly developable. Additionally, consider the example in Fig. 3.6: while the result of MLLE may at first glance look best, close inspection reveals that the mapped points are simply ignoring the curvature of the initial manifold. This non-isometric behavior thus obscures the nature of the input data, and is hardly useful in concrete applications.

SAKE vs. Hessian-LLE. Hessian-LLE [71], shares also a close relationship to our approach: the authors recognized the value of reproducing affine functions on each chart of the manifold to find a reliable embedding. Consequently, they propose to construct local Hessian operators since their theoretical null space is restricted to affine functions. They proceed by first unfolding each neighborhood via PCA, which fails to be robust as soon as the input data is a bit noisy; see Fig. 3.8. From the neighborhood projected onto a low-dimensional tangent embedding, they construct a discrete basis of constant, linear, and quadratic functions (this last subspace is constructed through pointwise product of linear functions). They finally orthonormalize these bases through Gram-Schmidt. Thus, their method requires $O(ds^2)$ more operations [88], where $s = d(d+1)/2$ is the dimensionality of the space

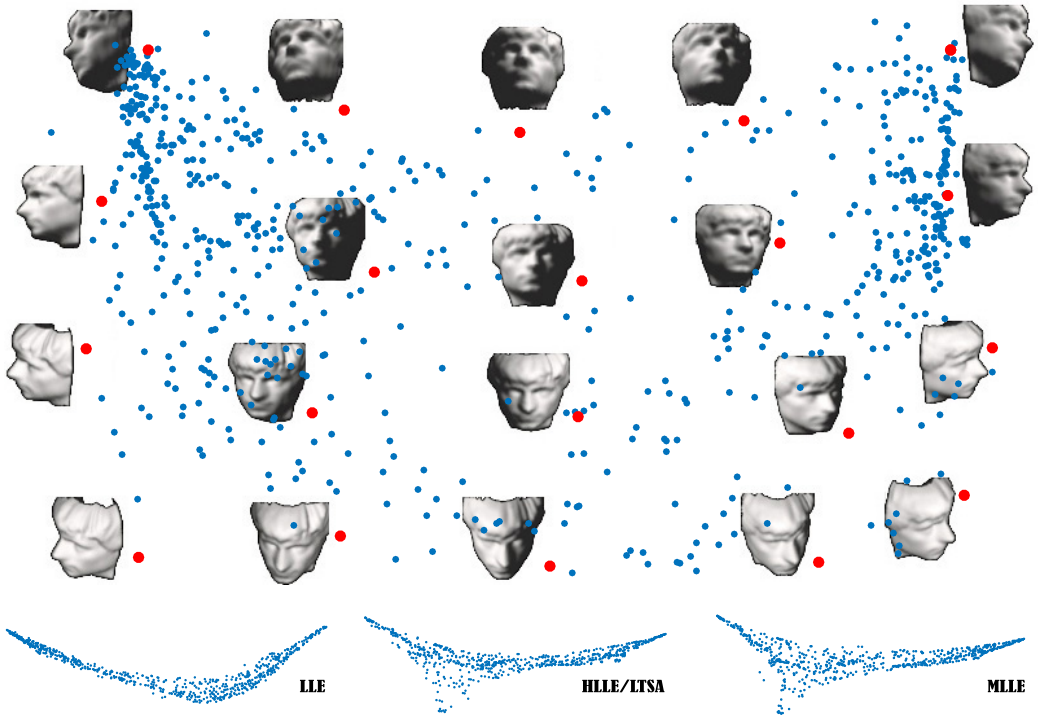


Figure 3.11: **Faces Dataset.** From 698 images representing the same 3D face from different viewpoints (each image is a point in $\mathbb{R}^{4096}$, see [180]), we compute the 2D SAKE ($k = 6, K = 36$) embedding purely based on local pixel-per-pixel distances of the images. The result parameterizes the camera angle quite accurately, with no a-priori knowledge. Examples of the original face images are given for the 20 red dots (image backgrounds removed for clarity). Other methods give markedly more distorted results (bottom).

of quadratic functions. Additionally, we note that HLLE uses tangent planes at the center of their local PCA as noticed in [198]; however, this means that, for very irregular sampling, two vertices may lead to nearly identical Hessian estimates, thus not providing added information. In sharp contrast, our approach does *not* try to construct an approximate Hessian, but an operator whose kernel contains only affine functions—which can be done faster and is significantly less sensitive to noise in the data; see Figs. 3.7 and 3.8 for example.

3.3.3 Experimental Results

Synthetic 3D datasets. For the past decades, manifold learning approaches have used a number of synthetic examples to demonstrate their results. We provide our results on a number of classical examples for comparison purposes, with and without noise, for regular and irregular sampling, and for low and high-dimensional datasets.

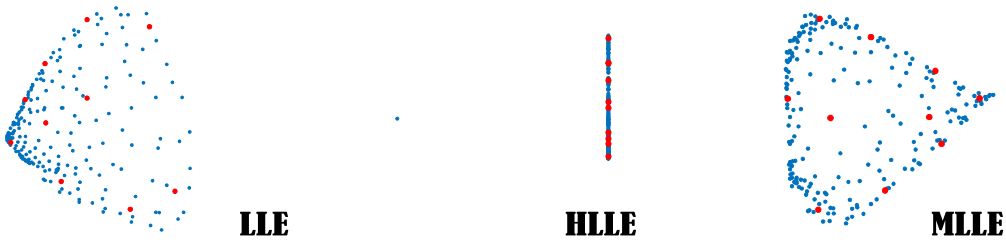


Figure 3.12: **Knights.** Examples of results for the knights images (see Fig. 3.1 for comparison). Other methods give markedly more distorted results; HLLE, in particular, fails to return a valid parameterization.

3D-to-2D Datasets. Fig. 3.5 shows that SAKE can handle the usual "Swiss roll" example that all other methods use in their tests, even with highly irregular sampling. Note that our geodesic curvature correction makes a big difference on such an example: if the regular, non-corrected Isomap is used to unfold every neighborhood, trying to make the final coordinates affine in these distorted local coordinates induces a large global distortion, confirming that our correction is crucial in our approach. Fig. 3.3 shows the noise-free 'S-shape', with a uniform sampling and a hole in the middle. As expected, Isomap cannot unfold this non-simply connected domain without distortion. The Laplacian eigenmap is surprisingly deformed, most likely due to the presence of harmonic functions. In sharp contrast, HLLE and LTSA return very similar (and correct) results, and SAKE matches these results. Fig. 3.6 shows as-isometric-as-possible flattenings of a noise-free two-bump height function sampled on a regular grid. On this example, LLE creates foldovers on the bumps. HLLE and LTSA are once again visually indistinguishable, but exhibit shrinkage on the two corners near the bumps. MLLE keeps the symmetry of the domain perfectly, but the curvature of the bumps is totally ignored, amounting to a orthogonal projection onto the support plane—hence creating significant metric distortion between the original manifold and its 2D parameterization. SAKE captures the domain and its symmetries, with the expected metric-preserving parameterization of the bumps. Fig. 3.2 exhibits how even an irregularly sampled simple plane fails to be correctly captured by a Laplacian eigenmap, while SAKE guarantees perfect projection. Note that Isomap would also fail to keep the original sampling since shortest distances are computed on a graph, which creates zigzagging (and thus inaccurate) geodesics.

Noisy Datasets. Testing robustness to noise is also informative. We tried the well-known 'Swiss roll' with noise "peppered" around: only 10% of the points are being displaced with a uniform noise distribution along the surface normal of the

roll to simulate inaccurate samples (see Fig. 3.7). LLE already suffers from this noise (other values of k result in worse deformation), and so are HLLE and LTSA to a lesser extent. MLLE and SAKE return very similar and visually plausible results. If we now try the Swiss roll with Gaussian noise (standard deviation of 0.28) as shown in Fig. 3.8, HLLE and LTSA fail entirely, while LLE is still quite deformed. MLLE appears relatively good, although the right end of the strip is squeezed in a way similar to LLE, but with a less pronounced effect. SAKE remains best among all the methods. We analyze two types of error, arising from the MDS and the global embedding construction steps of SAKE, respectively: one corresponds to local distance preservation computed pointwise as the relative 2-norm of the rank d approximation of the Gram matrix $\mathbf{G}_i$, while the other measures the quality of the local relative position preservation as the 2-norm of the changes in relative positions $||\mathbf{L}_i\mathbf{S}_i\mathbf{Z}||_2$. Looking at Fig. 3.9, we see that Gaussian noise results in small, uniform reconstruction and distance errors, while sparse outliers do not disrupt the non-noisy parts of the domain, proving robustness to various types of noise.

Handling Large Density Variations. We also tried significant variations in sampling density (in addition to noise) to test the robustness of various manifold learning approaches in Fig. 3.10. Here again, SAKE matches or surpasses other results, which prove to be extremely dependent on the number of neighbors used. Note that we did not try to adapt the value of k and K based on density (which would improve accuracy) to offer a fair comparison.

Handling Large Noise and Outliers. While previous methods rarely discuss the issue of outliers and large amounts of noise, SAKE offers a number of opportunities to derive more robust strategies to noise and outliers. As we discussed in Sec. 3.2.1, one can tweak our use of k closest neighbors to remove obvious outliers and noise: for instance, the authors of [145] propose a weighted local linear smoothing and one-ring minimum spanning trees (to detect significantly large edge size in the neighborhood, a tell-tale for outliers) for noise reduction and outlier handling. The choice of K geodesic neighbors may also be made adaptive by using iterative robust PCA [198] instead, to select the number of neighbors based on a local noise level estimate. Finally, our curvature corrected approach to intrinsic distances can also be extended to "smooth out" (through a curve straightening flow) the geodesic curves obtained by our process if noise is significant. These add-ons are trivial to incorporate, but a full assessment of how they fare on real defect-ladden data is out

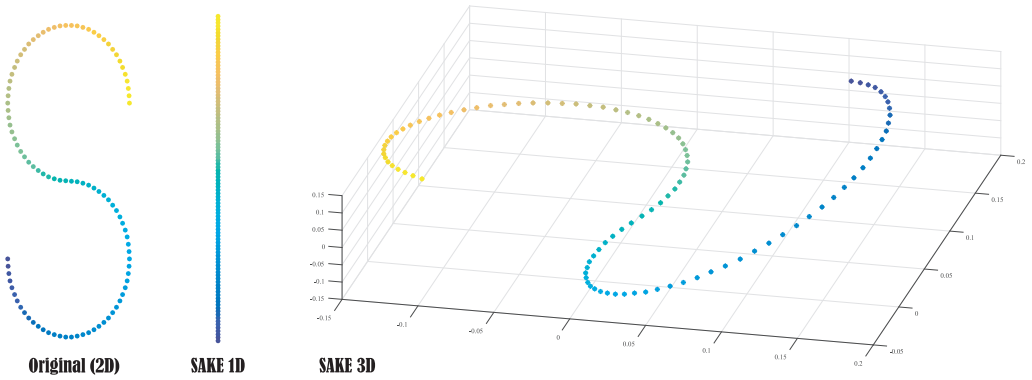


Figure 3.13: **Choice of Dimensionality.** For a 2D S-shaped pointset (left), its reduction to 1D provides the expected parameterization (middle); a 2D embedding would return the exact same input shape as the original; a 3D embedding, instead, exhibits unexpected distortions in the extra dimension (right); and of course, a 0D embedding would lose injectivity.

of the scope of this chapter. We favored instead a parameter-free SAKE, which is already more noise-resilient than the existing methods we tested.

High-dimensional Datasets. We also ran our SAKE algorithm on the classical 'Faces' datasets, where grayscale pictures with 64×64 pixels capture the same face from different viewpoints [180]. From this set of 698 images, each represented as a point in $\mathbb{R}^{4096}$ (one dimension per pixel), we can ask SAKE to reduce the dataset to a 2D projection purely based on pixel-per-pixel *differences* between pairs of images. Fig. 3.11 shows the result, where the points are clearly embedded in a position dependent on the left-right, up-down angle of view. For comparison purposes, we also show the results of a few other methods for this dataset on the same figure.

We also tried our approach on a dataset of reflectance fields captured using the Light Stage apparatus [117]. A static character (in a knight costume) was captured under 221 individual lighting directions covering a large sphere of illumination. Once again, we use all 608×456 RGB images (given in random order) stored as points in $\mathbb{R}^{831744}$ and try to learn a flat 2D manifold that best fits this high dimensional dataset. The result, shown in Fig. 3.1(top), recovers positions related to light angles without any knowledge of the setup, while other approaches lead to (sometimes severely) distorted embeddings; see Fig. 3.12. Note that the black background of each image was removed for clarity of the figure.

3.4 Mesh and Pointset Deformation using SAKE

3.4.1 Nonlinear Deformation as an Embedding Problem

Mesh deformation has a long history in geometry processing as it is one of the most important tools in practical graphics applications [171]. One can understand mesh editing methods as also being embedding methods, but for $D = d = 3$ (the term reduction becomes hardly appropriate in this case) and with added embedding constraints in the form of user-specified handles to control how to deform the shape—see, e.g., [97, 105, 166]. In fact, a simple nonlinear 3D modeling approach known as ARAP [173] (for as-rigid-as-possible) is precisely a special case of Local Tangent Space Alignment (LTSA) restricted to special orthogonal transformations: both the local alignment from the original shape to the final embedding through SVD and the alternating optimization between fixed rotations and positions were already spelled out in the Appendix of [204]. The only differences in the mesh case of ARAP reside in the choice of local neighborhoods (mesh one-rings for ARAP vs. K nearest neighbors in LTSA) and the additional constraints controlling the coordinates of the few handles. Based on these similarities, it is clear that we can leverage our new SAKE embedding approach to provide an alternative mesh (or pointset) deformation technique, where a standard off-the-shelf eigensolver can be used to solve the resulting nonlinear optimization.

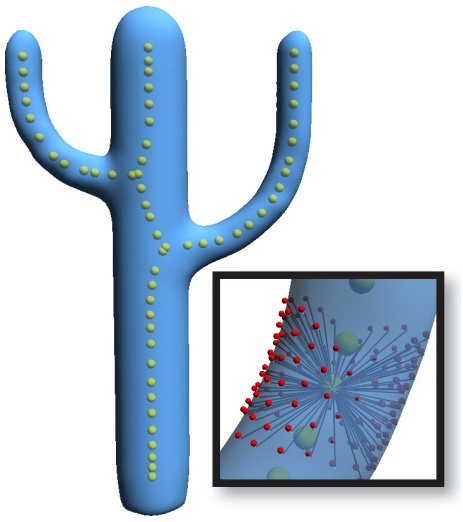
Conceptually, mesh deformation techniques are embedding methods as well, although they do not map onto a reduced space: given an input simplicial 2- or 3-manifold in $\mathbb{R}^3$ and a few select displacements of handle vertices in space, they compute a deformed embedding that interpolates constrained vertices while preserving the local neighborhoods of the original mesh as closely as possible. In fact, differential coordinates based on the Laplace-Beltrami operator (see a review of existing methods in [171, 172]) can be seen as a close relative of Laplacian Eigenmaps, where a particular embedding is found based on the initial Laplacian operator through a linear solve instead of an eigenvalue problem. Nonlinear approaches to mesh deformation have been introduced as well to prevent visual artifacts of previous linear methods [29, 97, 104, 205]. One of the simplest nonlinear methods is the as-rigid-as-possible (ARAP) modeling approach [173], which extends the Laplacian editing paradigm to better enforce surface rigidity—at the price of a slow-converging alternate minimization instead of a simple linear solve. We will show that our spectral approach, when used on low-dimensional data and with a few added terms to accommodate position constraints, provides a nonlinear alternative to surface deformation which applies to meshes as well as pointsets and for which

off-the-shelf libraries can be used for fast and robust results.

3.4.2 Geometry Deformation through Eigenanalysis

We propose to use the SAKE framework to achieve mesh deformation with handle position constraints. Notably, we aim at formulating a nonlinear deformation approach that only requires a sparse matrix eigenanalysis—instead of relying on customized nonlinear solvers with often slow convergence properties. This is trivially achieved (and without the need for as-isometric-as-possible unfolding of neighborhoods in this pure 3D case) by leveraging and extending our multi-Laplacian operator and its affine kernel: we accumulate constraints for the 3D points to stay close to their original positions with respect to their neighbors, and treat user-provided absolute handle positions by systematically replacing them by positions *relative* to other handle positions. We bypass the use of absolute position constraints or local rotations, and can thus invoke a regular eigensolver instead. This allows us to efficiently accommodate deformation properties of the 3D volume defined by the input geometry that are typically desirable in surface editing tools such as low local stretching and low volume dilation.

Surface and Volume Neighborhoods. In order to make our explanations valid for meshes and pointsets alike, we will use the term "point" to refer either to a point within a pointset, or to a vertex of an input mesh. For each point $\mathbf{x}_i$ of the input, we fix a set of K neighbors $\mathbf{x}_j$ for indices j in $\mathcal{N}(i)$ just like in Sec. 3.2.1—if we are dealing with an input mesh, the neighbors can also be defined via a fixed number of rings around the vertex instead. We also add a few internal points that are coarsely sampling the skeleton (also called medial axis) of the input geometry; this is easily achieved by, e.g., computing a Voronoi tessellation of the input points and extracting a subset of the *inside poles* [7] that form a discrete approximation of the skeleton. Each of these sparsely inserted skeletal points are then given a neighborhood which consists of their immediate skeletal neighbors and the few closest input points (i.e., on the medial ball [7]) along with their associated neighborhoods: these neighborhood can be understood as forming an internal set of radial



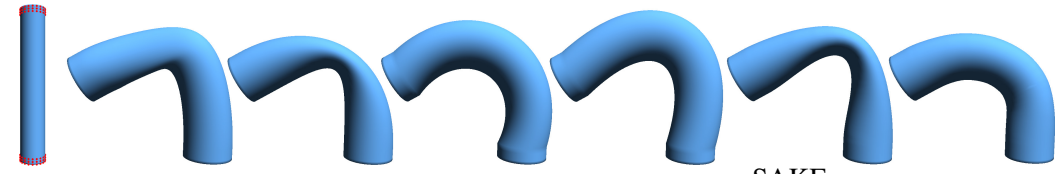


Figure 3.14: **Bending Bars.** We compare our approach (with and without added medial points for clarity) with a number of existing deformation methods on the 120° bend example with handles (marked in red on the original mesh) taken from [28]. Note the absence of artifacts near the pinned extremities for SAKE.

trusses (see inset) for which we will *also* impose isometry preservation during the deformation—thus avoiding local collapses of the initial volume as much as possible. This is very much in the spirit of the volumetric graph Laplacian approach in [205] where a sparse sampling of the interior of a mesh is also used to prevent local volume loss. Finally, the user picks a subset of input points $\{\mathbf{x}_h\}$ as "handles" for which she assigns target positions $\bar{\mathbf{x}}_h$ to guide the deformation. For clarity thereafter, we denote by $\mathcal{X}$ the set of indices of the input points that are not handles, $\mathcal{V}$ the set of indices of the added inner skeletal points, and $\mathcal{H}$ the set of indices of the handles.

Multi-Laplacian Assembly. For every point $\mathbf{x}_i$ (be it a handle, regular, or skeletal point) and its associated set $\{\mathbf{x}_j\}_{j\in\mathcal{N}(i)}$ of neighbors, we assemble the multi-Laplacian matrix $\mathbf{L}_i$ as described in Sec. 3.2.3. Moreover, we also assemble an additional multi-Laplacian, denoted $\mathbf{H}_h$, per handle point as follows: first we create virtual neighborhoods by assigning for each handle $\mathbf{x}_h (h \in \mathcal{H})$ the set of indices $\mathcal{N}(h) = \mathcal{H} \setminus \{h\}$ containing all the other handles (or only the K nearest handles when the number of handles is large); the matrix $\mathbf{H}_h$ is then defined as a regular multi-Laplacian, but for this virtual neighborhood made of handles only, and for the user-prescribed target positions $\bar{\mathbf{x}}_h$ instead of the original positions $\mathbf{x}_h$. The linear equations encoded by this matrix will force the handle positions to be placed, *relative to each other*, the way the user asked for.

Quadratic Form to Enforce Constraints. We can now formulate our geometry deformation approach as a special instance of SAKE where we look for the vertices $\mathbf{Z}$ satisfying

$$\mathbf{Z} = \underset{\mathbf{Z}\in\mathbb{R}^{3\times n}}{\arg\min}\ \mathrm{Tr}\big[\mathbf{Z}\mathbf{Q}\mathbf{Z}^T\big] \quad \text{s.t. } \mathbf{Z}\mathbf{Z}^T = \mathbf{Id},\ \ \mathbf{Z}\mathbf{1} = \mathbf{0}$$

,

where the quadratic form $\mathbf{Q}$ is defined through

$$\begin{aligned}\mathbf{Q} = & \sum_{i\in\mathcal{X}} \mathbf{S}_i^T\ \mathbf{L}_i^T\ \mathbf{L}_i\ \mathbf{S}_i \\ & + \sum_{h\in\mathcal{H}} \mathbf{S}_h^T\ \left(w_{\mathcal{H}}\mathbf{L}_h^T\ \mathbf{L}_h + w_C\mathbf{H}_h^T\ \mathbf{H}_h\right)\ \mathbf{S}_h \\ & + w_{\mathcal{V}} \sum_{i\in\mathcal{V}} \mathbf{S}_i^T\ \mathbf{L}_i^T\ \mathbf{L}_i\ \mathbf{S}_i.\end{aligned} \tag{3.2}$$

The form $\mathbf{Q}$ thus enforces the linear constraints of not only the surface neighborhoods (first term and part of the second term, where handle neighborhoods are scaled by $w_{\mathcal{H}}$ to strengthen the local rigidity around handles), but also of the volume neighborhoods (third term, with a specific volume control coefficient $w_{\mathcal{V}}$), and of the handles with respect to each other (final part of the second term, using a strength w_C for these constraints).

Final Shape Extraction. Finding the final deformed shape is achieved by extracting the first four eigenvectors $\mathbf{q}_m$ corresponding to the lowest eigenvalues of the sparse SPD matrix $\mathbf{Q}$ (the first one being constant and corresponding to the zero eigenvalue; see Sec. 3.2.5). Because the eigensolver returns the positions up to a rigid transform and up to scale, we must compute a global affine transform to map these resulting positions where they need to be. Denoting the resulting handle positions $\mathbf{p}_h = (1, (\mathbf{q}_2)_h, (\mathbf{q}_3)_h, (\mathbf{q}_4)_h)^T$ for $h \in \mathcal{H}$, we find through a least squares solve the affine transformation encoded by a 3×4 matrix $\mathbf{A}$ such that $\mathbf{A}\mathbf{p}_h$ best fits the user-prescribed $\bar{\mathbf{x}}_h$. Then the final, deformed geometric positions are simply set via

$$\mathbf{z}_i = \mathbf{A}\begin{pmatrix}1 & (\mathbf{q}_2)_i & (\mathbf{q}_3)_i & (\mathbf{q}_4)_i\end{pmatrix}^T.$$

3.4.3 Implementation Details

Our approach is rather simple as it only requires the assembly a large, sparse, and symmetric matrix $\mathbf{Q}$ and the use of sparse matrix eigenanalysis tool—we use the *Spectra* library [150] in our implementation. Aside from the modeling parameters to influence the way a shape deforms (we use $w_{\mathcal{H}} = 5$, and $w_C = w_{\mathcal{V}} = 1$ in all 3D examples in this chapter, but other choices including geometry-dependent coefficients can be used to achieve desired effects), there is no need for a customized numerical technique to solve for our nonlinear deformation. Note that if the total number of handles is 4 or less, we cannot express their target 3D locations as linear combinations of the others as we did in Eq. (3.1) since their edge matrices $\mathbf{E}_h$ will have trivial

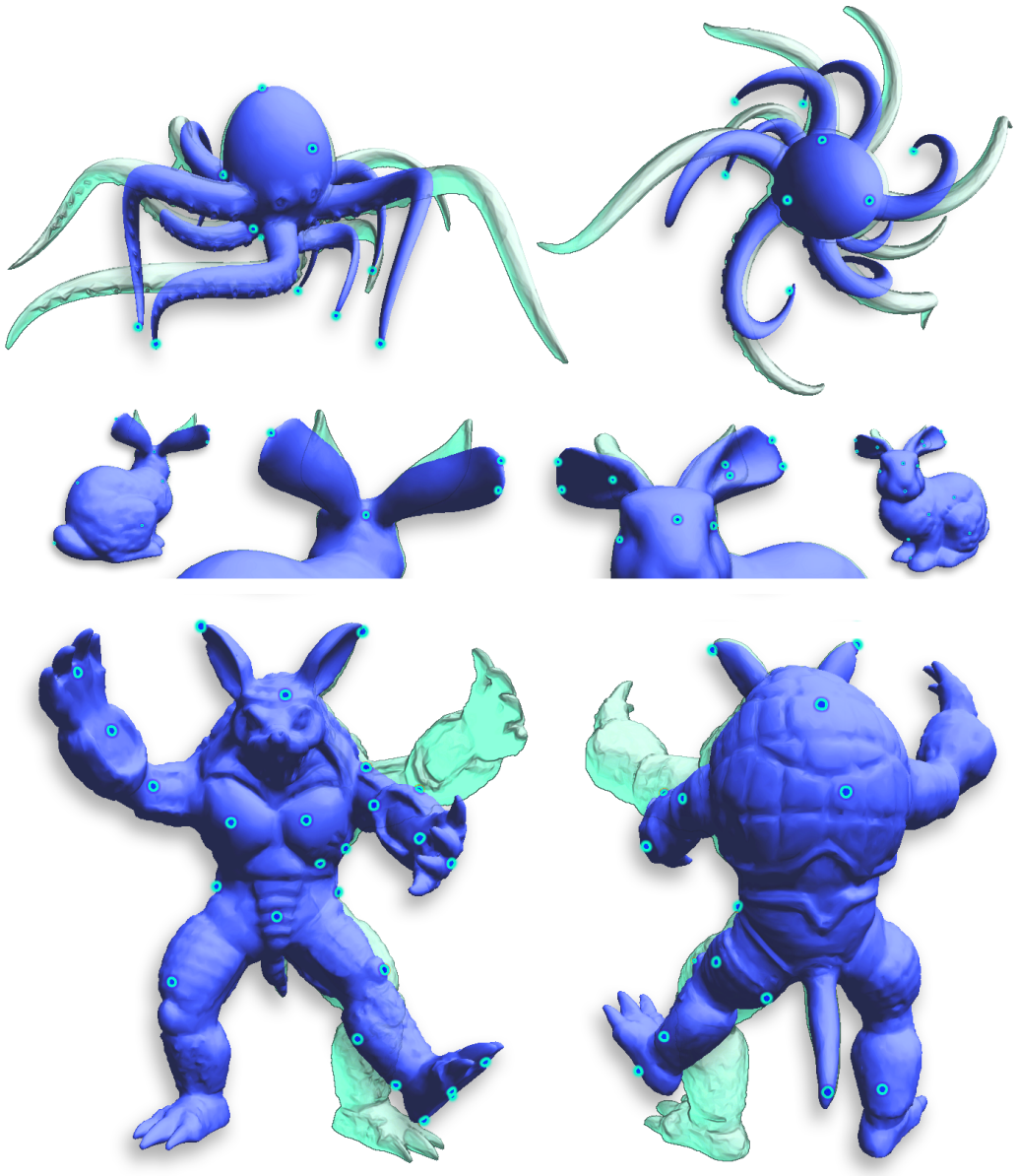


Figure 3.15: **Zoo.** We use our SAKE editing approach on a few basic 3D models. Light blue markers indicate handle vertices (held fixed or displaced). All models use the same editing parameters; a subsampled medial axis was only added to the octopus model (20 added points) to increase the volume rigidity of the long and thin tentacles.

kernels. We thus create weaker constraints by enforcing that only one or two of the handle coordinates are linear combinations of the other handles' coordinates.

Possible Variants. More specific deformation behavior of the input geometry can be also easily added; for instance, a shell-like behavior can be incorporated by adding offset points along the positive and/or negative normal direction to the surface to

mimic the way PriMo [29] was adding bending rigidity. While we described how to apply SAKE to surfaces, discrete volumes (given as meshes or pointsets) are handled in exactly the same way. Additionally, notice that our multi-Laplacian based quadratic form can also be used to achieve realtime deformation using handle constraints as in [192]: substituting the modified natural boundary Laplacian used in their approach with our multi-Laplacian construction will add rigidity as our multiple relative coordinates capture the original geometry more tightly. However, requiring every handle position to be exactly satisfied can inherently lead to a loss of local smoothness (see, for instance, artifacts near pinned extremities [192] compared to our results in Fig. 3.14), an issue that our spectral approach avoids—without significant computational overhead. Finally, we note that our treatment of handles through *relative coordinates* can also turn their work into a spectral approach.

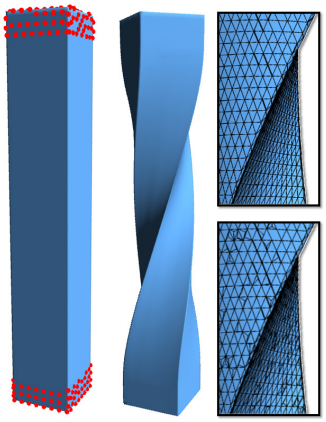
Computational Efficiency. Once an input geometry is given, one can compute the local multi-Laplacian operator for each neighborhood as a preprocessing step. When the handles have been selected and positioned as desired, the small matrices $\mathbf{H}_h$ can be computed on the fly and the corresponding terms added to the quadratic form according to Eq. (3.2). Assembling $\mathbf{Q}$ is done in $O(n)$ as in any other mesh deformation method. An eigensolver can then be used on $\mathbf{Q}$ to find its lowest eigenvectors, and the undeformed positions can be used as a good initialization. On a Intel dual-core Core i7 powered laptop, creating and Choleski-factorizing the quadratic form takes 5 seconds on a 10kV mesh like the bunny with a straightforward non-optimized code, while the eigensolver returns the proper embedding in 0.2 seconds. Note that the result of the deformation could be updated quite efficiently as the handles are moved if realtime editing is needed: low-rank update of the partial eigendecomposition of our symmetric matrix $\mathbf{Q}$ can be leveraged [31, 37] since only the matrices $\mathbf{H}_h$ are changing—although we did not need to implement this variant given the current timings. Finally, note that control over rigidity could have been achieved by changing the locality of the neighborhoods: the larger they are, the more rigid the local volumes become. Our addition of a few points along the skeleton was preferred to offer rigidity control by only marginally increasing the size of the matrix, but without impacting its sparsity. The user can easily add more constraints by inserting extra rows and columns in $\mathbf{Q}$.

Numerics. Our approach uses an eigensolve to handle fully nonlinear mesh deformation. Compared to other nonlinear methods, solving an eigenvalue problem

is simple and efficient: many off-the-shelf libraries exist as well, offering efficient eigensolvers on multi-core architectures with convergence guarantees (often lacking both in theory and in practice for other nonlinear methods) even for significant deformations.

3.4.4 Experimental Results

We present examples of deformation in Fig. 3.15 on the bunny, armadillo, and octopus meshes. In each example, the same default values for the weights in Eq. 3.2 are used. We only used inside poles on the octopus (20 virtual points) to add volume rigidity to the tentacles. We also provide comparisons with previous deformation techniques on a simple cylinder mesh (Fig. 3.14) based on handle constraints provided in [28]; without inside poles, the surface deformation is isometric but the resulting shape exhibits local volume change (it behaves like an inextensible shell). With just 10 inside poles added as virtual points, the bar keeps its round section through the whole cylinder length. Contrary to the results of [29, 119, 192] in Fig. 3.14, our approach does not exhibit smoothness artifacts near the pinned extremities of the tube. We also tested twisting the bar from [28] without poles — see inset; note that the resulting shape does not depend on the quality of the triangulation: both the mesh and a refined version with nearly degenerate elements (closeups) lead to the same twisted shape. With our eigensolver, user-defined handle positions were always matched within 10^{-3} by the reconstruction for normalized bounding boxes, and further reduced to 10^{-5} if the handle weight $w_{\mathcal{H}}$ is raised to 50. Finally, we note that one can think of our SAKE deforming tool for 3D shapes as a ***multi-cage approach***: every point is considered as part of a local cage, as it is expressed as a linear combination of neighbors like in a conventional cage-based deformation method. The only difference is that the final embedding is found via a nonlinear solve involving a global, sparse eigenvalue problem to satisfy *all* cage constraints as well as possible.



3.5 Conclusion

In this chapter, we introduced an approach for finding a low dimensional embedding of a d-manifold originally embedded in a high dimension. A global embedding in $\mathbb{R}^d$ is found such that its coordinates are as affine as possible within the local

isometric parameterization of each small neighborhood, enforcing a most-isometric map (up to a global linear transformation). In the process, we revisited a variety of well-established manifold learning approaches, providing geometric improvements such as local linear precision to these nonlinear reduction methods. We formulated a spectral affine-kernel embedding framework, based on local eigenanalysis of each neighborhood followed by a global, sparse spectral solve to find the best low-dimensional embedding, which is significantly more robust to irregular sampling and to reasonable amounts of noise than previous methods. We also proposed a spectral as-rigid-as-possible deformation tool for 3D data which leverages our SAKE approach to offer nonlinear shape editing without having to resort to dedicated nonlinear solvers.

Chapter 4

PARALLEL TRANSPORT UNFOLDING

4.1 Contributions at a Glance

Manifold learning through Isomap is one of the most used nonlinear dimensionality reduction methods, even if its use of a dense Gram matrix implies a higher computational complexity than local methods: Isomap offers a remarkable robustness to noise. Its entire reliance on graph-based shortest paths has, however, far-reaching calamitous consequences. First, spurious geodesic curvature (i.e., zigzags) in the Dijkstra shortest paths (seen as a piecewise linear curves in $\mathbb{R}^D$) between two nodes of the graph introduces inaccuracies in the estimation of geodesic distances. This issue impacts irregular sampling in general: unlike PCA, Isomap fails to produce a non-distorted embedding for a dataset lying precisely on a linear subspace of the ambient high-dimensional space; see Fig. 4.2. In practice, this shortcoming is exacerbated by the sparse sampling that real applications often have to deal with, even if short paths can be locally rectified through a straightening correction described in Sec. 3.2.2. Worse, some computational accelerations of Isomap rely on a subsampling of the initial data [64], making this inaccuracy issue all the more limiting. More importantly, Isomap can only provide a quasi-isometric low-dimensional mapping for *geodesically convex sets*: the presence of holes or non-convex boundaries in the input sampling brings significant overestimations of geodesic distances, as Dijkstra shortest paths are forced to go around sampling voids, thus distorting the results (see Fig. 4.3 and Fig. 4.5). There have been attempts to alleviate this limitation, such as the Topologically Constrained Isometric Embedding (TCIE) approach [154] which first tries to detect the boundaries of holes and the geodesic paths that go through these boundaries, before constructing a map that ignores such paths. However, detecting boundaries in a reliable way is a complex problem as voids can have a variety of sizes and shapes in the case of noisy and irregular sampling. Moreover, the subsequent minimization is significantly more difficult to perform as it no longer relies on a simple partial eigendecomposition. Consequently Isomap is often used in practice, even when the topology and geometry of the sampled manifold are not known, because of the lack of another available robust learning approach.

In this chapter we introduce Parallel Transport Unfolding (PTU), a manifold learn-

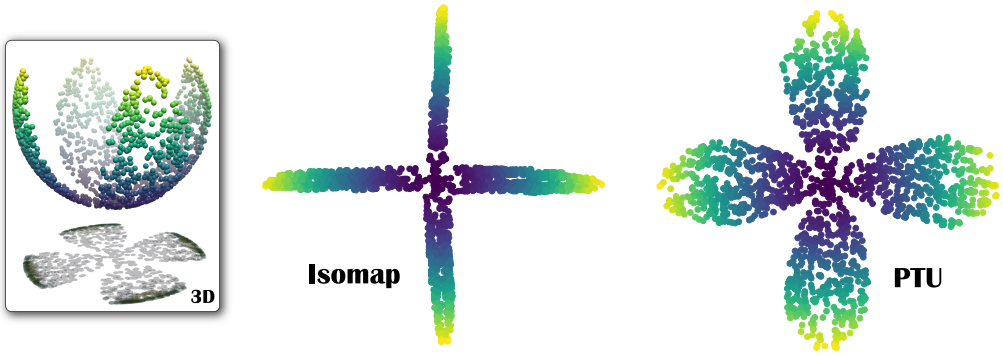


Figure 4.1: **Correctly Unfolding Petals.** For a 3D sampling of a 4-petal shaped portion of a sphere (left), Isomap (a staple of manifold learning) fails to find a near isometric 2D parameterization (middle) due to geodesic non-convexity of the intrinsic geometry. Our parallel transport approach instead handles this case as expected (right). Lifting the pointset to much higher dimensions and applying random rotations and reflections would not change these results.

ing technique that performs nonlinear dimensionality reduction to produce a quasi-isometric, low-dimensional embedding of an arbitrary set of high-dimensional data points. By simply replacing the Dijkstra-based geodesic distance estimates with parallel transport based approximations instead, our approach removes the geometric limitations of Isomap: it can reliably handle arbitrary data with strongly irregular sampling while retaining its resilience to noise. We show that this new geometric procedure no longer requires geodesic convexity of the domain sampled by the input data, only involves simple linear algebra, significantly improves the accuracy of all pairwise geodesic distance approximations, and does not change the overall computational complexity of the original Isomap procedure. Moreover, our approach exploits the low dimensionality of the manifold by using a connection on the d-dimensional tangent bundle for efficiency purposes: this markedly differs from previous parallel transport based geodesic computations on 2-manifolds in 3D, where the codimension is 1 and thus, where either the normal field or 2D polar coordinates can be leveraged to derive fast geodesic approximations [126, 159, 161]. Finally, we also demonstrate that our connection-based distance estimation applies equally well to the Landmark-Isomap approach [63], a variant of Isomap offering significant improvements in computational time but suffering from the same convexity limitation as Isomap.

4.2 Parallel Transport Unfolding Algorithm

Throughout our exposition, we use the notations and assumptions of Sec. 2.1.1. Our global parallel transport based manifold learning algorithm seeks to find an

as-isometric-as-possible d-dimensional embedding of the input data $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$ sampling a d-dimensional manifold $\mathcal{M} \in \mathbb{R}^D$ as a pointset $\{\mathbf{z}_i\}_{i=1}^n \subset \mathbb{R}^d$.

We now present our Parallel Transport Unfolding (PTU) algorithm, which has a very similar structure to Isomap since it comprises the following steps:

- Construct a proximity graph G of the pointset $\mathcal{S}$ and compute local tangent spaces at each point;
- Approximate all pairwise geodesic distances using parallel transport along shortest paths on G;
- Perform MDS to find a d-dimensional embedding that best preserves all the geodesic distances.

We detail each step next, stressing the key differences with Isomap in the first two steps of the algorithm, i.e., the use of approximate tangent spaces and of the Levi-Civita connection to better approximate intrinsic geodesic distances.

4.2.1 Proximity Graph

The construction of a proximity graph G on $\mathcal{S}$ proceeds similarly to the original Isomap algorithm: one can link each point to its neighbors contained in an ϵ-ball, or to its k nearest neighbors—both based on Euclidean distances in $\mathbb{R}^D$. Variants such as a mix of the two [189] or the mutual k nearest neighbors approach [34] can also be used to naturally discard outliers. For clarity of presentation, we will use a vanilla k-NN graph in our exposition and all of our tests. The value k should be chosen such that the edges of the resulting graph are good approximations of geodesics between corresponding points; we will typically set k around $4d$ to induce a valence greater than the usual connectivity of a regular grid of the d-dimensional embedding ($k = 2d$), yet less than the number of 1-ring neighbors on that same grid ($k = 3^d - 1$); but knowledge about noise levels in the input pointset can be employed to improve the graph quality by varying the number of neighbors from point to point. Once the graph connectivity is defined, each edge is assigned a weight equal to its Euclidean length in $\mathbb{R}^D$ as an approximation of its geodesic length on the manifold $\mathcal{M}$.

4.2.2 Tangent Spaces and Orthonormal Bases

For each input point $\mathbf{x}_i$, its K nearest neighbors on the proximity weighted graph (for $K \geq k$) are used to define a *geodesic neighborhood*. The matrix, whose rows are the

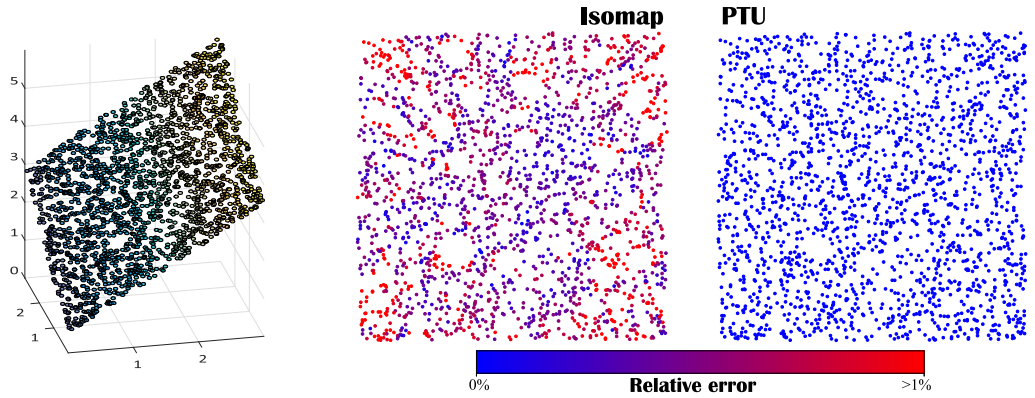


Figure 4.2: **Linear Precision.** Due to its reliance on graph-based paths, Isomap distorts a flat 2D pointset embedded in 3D; instead, our connection-based approach (right) flattens it exactly. A linear color ramp from blue (0% ℓ^2-error in position relative to bbox size) to red (> 1% error) is used in the visualization of the two embeddings.

data points in this neighborhood (centered by subtracting $\mathbf{x}_i$), induces, through its d left singular vectors corresponding to the d largest singular values, an orthonormal basis $\mathbb{T}_i$ of d vectors in $\mathbb{R}^D$ spanning the approximate tangent space:

$$\mathbb{T}_i = \begin{pmatrix} \mathbf{t}_1^i & \cdots & \mathbf{t}_d^i \end{pmatrix} \in \mathbb{R}^{D \times d}. \tag{4.1}$$

While choosing $K = k$ often suffices to provide a good estimate of local tangent spaces, using a value of K distinct from k allows the definition of arbitrarily large geodesic neighborhoods (useful in the presence of strong noise) around each point of $\mathcal{S}$, while alleviating the traditional issue of "manifold shortcutting" associated with increasing k (as leveraged in Ch. 3). As a substitute to this frame construction via partial SVD, note that an improved approximation of the local tangent spaces in the presence of strong noise and outliers can also be computed via ℓ_1-based robust PCA (see, e.g., [202]).

With these orthonormal frames of tangent spaces in place, we can now discuss how to approach discrete parallel transport and how to use it for geodesic length estimation.

4.2.3 Dicrete Parallel Transport

We now cover the core of our approach, i.e., exploiting parallel transport in high dimension to better evaluate geodesic distances.

Parallel Transport in Differential Geometry. The notion of parallel transport plays a central role in differential geometry. It induces a way to connect the geome-

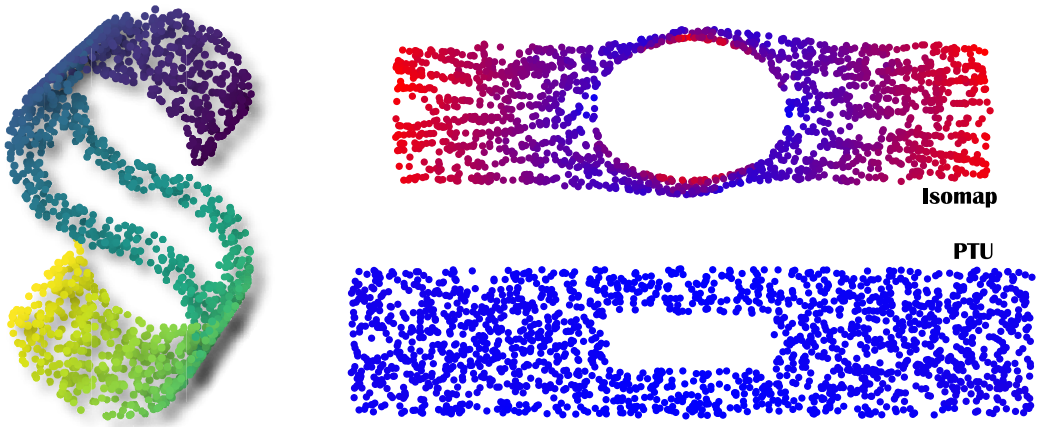


Figure 4.3: **Holey S.** Isomap (top right) fails to find a quasi-isometric embedding of a uniformly sampled developable S-shaped 2-manifold with a sampling void in the middle (left) in 3D. Our approach (bottom right) unfolds it almost perfectly. A linear color ramp (blue: 0% error, to red: 15% error) is used in the visualization of the two embeddings.

tries of nearby points, thus prescribing how a basis of the tangent space at one point of a manifold should be adjusted to produce a "parallel" basis of another tangent space at a nearby point. While this procedure is straightforward for flat spaces (it corresponds to a simple translation), it becomes more involved for manifolds with non-trivial curvature. Its differential geometric treatment involves the definition of a connection on the tangent bundle [79, 174], which represents an infinitesimal analogue of parallel transport. Most relevant to our work are *metric* connections, i.e., connections such that the parallel transport they define preserves the intrinsic metric of the manifold. The well-known Christoffel symbols are, in fact, the components of a particularly canonical metric connection called the Levi-Civita connection, which we will leverage in our application. Another geometric property we will exploit is the fact that geodesics, usually described through variational analysis as locally shortest curves, can also be defined through parallel transport: a geodesic is a curve that parallel transports its own tangent vector, as directly implied by the geodesic equation [79, 174]. This simple property will guide our evaluation of geodesic distances.

Discrete Parallel Transport. Given points $\mathbf{x}_i$ and $\mathbf{x}_j$ sharing an edge in the proximity graph G, we define the *discrete metric connection* between $\mathbf{x}_i$ and $\mathbf{x}_j$ as the orthogonal $d \times d$ matrix $\mathbf{R}_{j,i}$ in $O(d)$ representing the change of basis that best aligns,

in the Frobenius norm, the frames $\mathbb{T}_i$ and $\mathbb{T}_j$, i.e.,

$$\mathbf{R}_{j,i} = \underset{\mathbf{R}\in O(d)}{\arg\min} ||\mathbb{T}_i - \mathbb{T}_j\mathbf{R}||_F^2. \tag{4.2}$$

By definition, $\mathbf{R}_{i,j}$ is the inverse of $\mathbf{R}_{j,i}$: $\mathbf{R}_{i,j} = \mathbf{R}_{j,i}^T$. Note that we use the group $O(d)$ of orthogonal matrices because the SVD used in tangent space extraction produces arbitrarily-oriented tangent frames. A pre-processing of these tangent bases could be performed (e.g., via a minimum spanning tree) if one wants to ensure that their orientations are consistent, enforcing that the discrete connections are pure rotations (in $SO(d)$) as in the continuous case. The transformation $\mathbf{R}_{j,i}$ can thus be understood as a discrete equivalent to the Levi-Civita connection induced by the metric on $\mathcal{S}$ inherited from the Euclidean space $\mathbb{R}^D$. It is also a high-dimensional extension of previous discretization of metric connections on triangle meshes [52, 121].

Computing this discrete connection is easily achieved via singular value decomposition:

Proposition 4.1. *Let $\mathbb{T}_i{}^T\mathbb{T}_j = \mathbf{U}\Sigma\mathbf{V}^T$ be the SVD. The discrete connection is expressed as*

$$\mathbf{R}_{j,i} = \mathbf{V}\mathbf{U}^T. \tag{4.3}$$

Proof. Expanding the objective function in Eq. (4.2), we conclude that $\mathbf{R}_{i,j}$ is the maximizer of $\mathrm{Tr}\left[\mathbb{T}_i{}^T\mathbb{T}_j\mathbf{R}\right]$ over $\mathbf{R}\in O(d)$ as the other terms do not depend on $\mathbf{R}$. Since $\mathbb{T}_i{}^T\mathbb{T}_j\mathbf{R} = \mathbf{U}\Sigma\mathbf{V}^T\mathbf{R}$ and given the invariance of the trace under cyclic permutations, the objective function equals to $\mathrm{Tr}\left[\left(\mathbf{V}^T\mathbf{R}\mathbf{U}\right)\Sigma\right]$ where $\mathbf{V}^T\mathbf{R}\mathbf{U}\in O(d)$. The optimal solution is achieved for $\mathbf{V}^T\mathbf{R}_{j,i}\mathbf{U} = \mathbf{Id}$; so $\mathbf{R}_{j,i}$ must be $\mathbf{V}\mathbf{U}^T$. One recognizes the Procrustes superimposition of two nearby frames [108], just extended to handle arbitrary choices of frame orientation for tangent spaces. □

4.2.4 Path Unfolding

With a discrete metric connection, we can now "unfold" a polyline path on $\mathcal{S}$ (made out of a series of adjacent proximity graph edges) into $\mathbb{R}^d$, i.e., we seek to *map a polyline of graph edges into a flat d-dimensional space while best preserving its metric properties such as length, intrinsic curvatures, etc*. This process of "unrolling" a curve onto a tangent space through parallel transport is known as Cartan's development [136, 164] in differential geometry. Since geodesic curve parallel transports its tangent vector, its associated Cartan's developement is simply a (segment of) straight line.

Two-edge Unfolding. Let us define how to unfold in $\mathbb{R}^d$ a three-point $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ polyline path on $\mathcal{M}$, i.e., a path made of two adjacent edges of the proximity graph on $\mathcal{S}$. Its extensions to arbitrary paths will be straightforward. Without loss of generality, we can consider the unfolding to be happening in the tangent space at $\mathbf{x}_i$, equipped with its orthonormal frame $\mathbb{T}_i$. The first point $\mathbf{y}_i$ of the unfolded polyline can be chosen to be at the origin of this subspace, or coinciding with $\mathbf{x}_i$. The first edge, represented by the vector $\mathbf{e}_i = \mathbf{x}_j - \mathbf{x}_i$, is projected (in the ℓ^2 sense) onto the tangent space at $\mathbf{x}_i$ to form a d-dimensional tangent vector $\mathbf{v}_i$ through:

$$\mathbf{v}_i = \mathbb{T}_i^T \mathbf{e}_i.$$

This vector $\mathbf{v}_i$ is then added to $\mathbf{y}_i$ to form the point $\mathbf{y}_j$, thus defining the first unfolded edge. The second edge, $\mathbf{e}_j = \mathbf{x}_k - \mathbf{x}_j$ is similarly projected onto the local tangent space defined by frame $\mathbb{T}_j$ at $\mathbf{x}_j$. The resulting vector can then be parallel transported onto the tangent space at $\mathbf{x}_i$ to become

$$\mathbf{v}_j = \mathbf{R}_{i,j} \left[\mathbb{T}_j^T \mathbf{e}_j \right],$$

where the term in brackets is the ℓ^2 projection of $\mathbf{e}_j$ onto $\mathbb{T}_j$. From this vector, now represented in the original tangent frame at $\mathbf{x}_i$, we construct the final point of the unfolded polyline as: $\mathbf{y}_k = \mathbf{y}_j + \mathbf{v}_j$.

Preservation of Geodesic Curvature. Due to our use of a discrete metric connection, the unfolding procedure we described has an important property: it nearly preserves the geodesic curvature of the initial curve. Indeed, we used a discretization of the metric-preserving Levi-Civita connection, so the *intrinsic* angle between vectors $\mathbf{e}_i$ and $\mathbf{e}_j$ (meaning, the angle measured on the manifold $\mathcal{S}$) is preserved by parallel transport, and corresponds to the angle between $(\mathbf{y}_i, \mathbf{y}_j)$ and $(\mathbf{y}_j, \mathbf{y}_k)$ in $\mathbb{R}^d$—up to discretization errors. This means that if the two-edge polyline were a good approximation of a geodesic on $\mathcal{S}$, the unfolded polyline would be (nearly) straight in $\mathbb{R}^d$ since a geodesic curve parallel transports its tangent vector. As a corollary, the Euclidean distance $\|\mathbf{v}_i + \mathbf{v}_j\|_2$ between $\mathbf{y}_i$ and $\mathbf{y}_k$ is a good approximation of the geodesic distance between $\mathbf{x}_i$ and $\mathbf{x}_k$ along $\mathcal{S}$—whether the polyline is an approximate geodesic or not.

Length Rescaling. In the case of highly-irregular samples with small or no noise, it can be beneficial to preserve the length of each edge $\mathbf{e}_i$ once it is projected onto the corresponding frame, as the projection $\mathbf{v}_i$ on the tangent space can be shortened

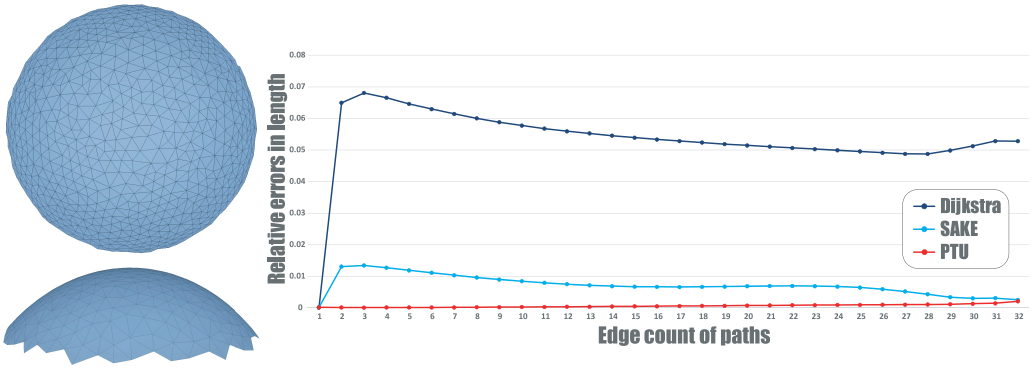


Figure 4.4: **Geodesic Distances.** For a graph based on a triangulated spherical cap (left, 2 views), the average relative error of pairwise Dijkstra-based geodesic distances is 5.6%; SAKE already reduces the error to 0.8%, but PTU brings it down to 0.046%. Plots (right) show relative errors of pairwise distances as a function of their number of edges.

if the manifold is curved. This adjustment is easily achieved by, for instance, setting

$$\mathbf{v}_i = \mathbb{T}_i^T \mathbf{e}_i \frac{\|\mathbf{e}_i\|_2}{\|\mathbb{T}_i^T \mathbf{e}_i\|_2}.$$

If strong noise and/or outliers are present, this modification should be avoided as it tends to overestimate local distances around noisy points. We only used this length rescaling procedure in Figs. 4.5 and 4.13 to slightly improve the results in these extreme cases.

4.2.5 Distance Approximation

The unfolding procedure we just introduced can be used to evaluate pairwise geodesic distances without the traditional shortcomings of Dijkstra's algorithm for graphs over an irregular sampling.

Correcting Shortest Paths. Consider a Dijkstra shortest polyline $(\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_m})$. Using the unfolding procedure into $\mathbb{T}_{i_1}$, we iteratively project the edges $\mathbf{e}_{i_s} = \mathbf{x}_{i_{s+1}} - \mathbf{x}_{i_s}$ onto the tangent space spanned by $\mathbb{T}_{i_s}$, before parallel transporting the resulting vector back to the original tangent space, i.e.,

$$\mathbf{v}_{i_s} = \left(\prod_{j=1..r-1} \mathbf{R}_{i_j, i_{j+1}} \right) \left[\mathbb{T}_{i_r}^t \mathbf{e}_{i_s} \right]. \tag{4.4}$$

After accumulating the results into a single vector $\mathbf{v} = \sum_{s=1..m} \mathbf{v}_{i_s}$, we set the approximate geodesic distance between $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_m}$ to be its length $||\mathbf{v}||_2$: as discussed above, the span of the unfolded path is a better approximation of the geodesic distance between the end points of the initial path, as it ignores the intrinsic twists and

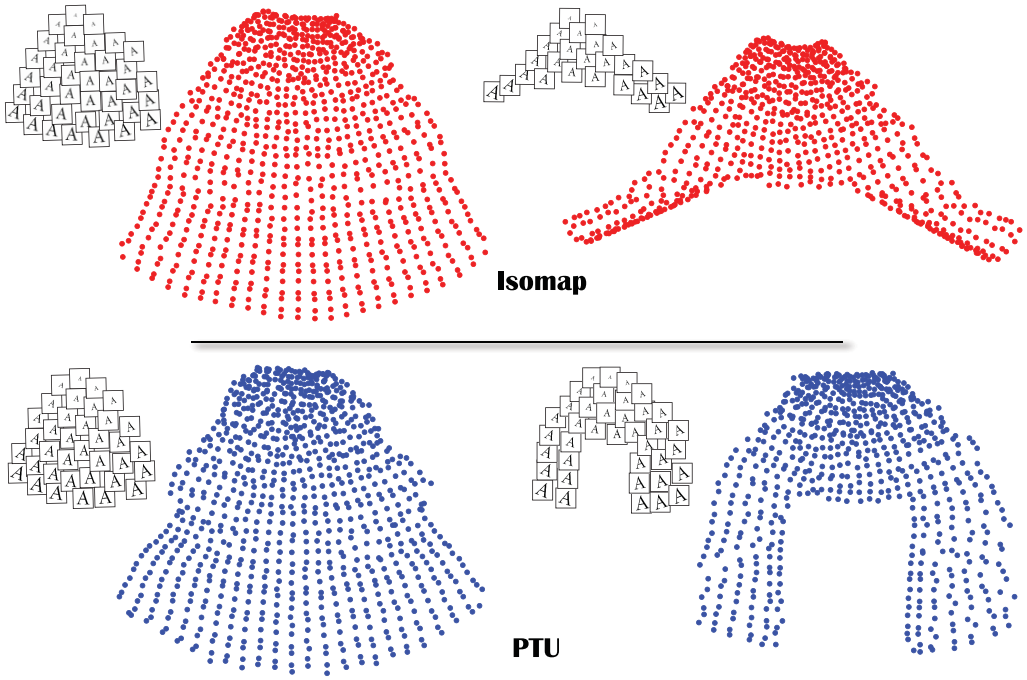


Figure 4.5: **Letter A.** From 888 images (120×120 pixels) of rotated and resized letters 'A', both Isomap (red) and PTU (blue) produce similar 2D embeddings (left, each dot indicating an image; insets show a subset of the images and their embedding). However, if a part of these input images is removed (rendering the set non-geodesically-convex), Isomap dramatically changes the embedding, while PTU properly reflects the missing images (right).

turns that the path went through. Since our estimate of the geodesic distance from $\mathbf{x}_p$ to $\mathbf{x}_q$ is in general not the same as the estimate from $\mathbf{x}_q$ to $\mathbf{x}_p$ due to the asymmetry of the unfolding process, we average the two spans in a final post-processing step to determine all the pairwise geodesic lengths. Fig. 4.4 demonstrates an improvement of *over two orders of magnitude* compared to Dijkstra's approach to estimating geodesic distances.

Leveraging Unfolded Paths. While we only need the Euclidean distance between the two end points of the unfolded path in $\mathbb{R}^d$ to perform nonlinear manifold learning, the unfolded path can also be useful for gathering additional information. For instance, the largest distance between the straight line between the two end points and the actual unfolded path indicates how far off the path is from being a geodesic. A large value is a typical telltale of the presence of a large void in the data within the manifold, which can be exploited to suggest where to insert new samples in order to improve the results. Similarly, a large difference between the distance estimates computed from one end point to the other and in the opposite direction implies

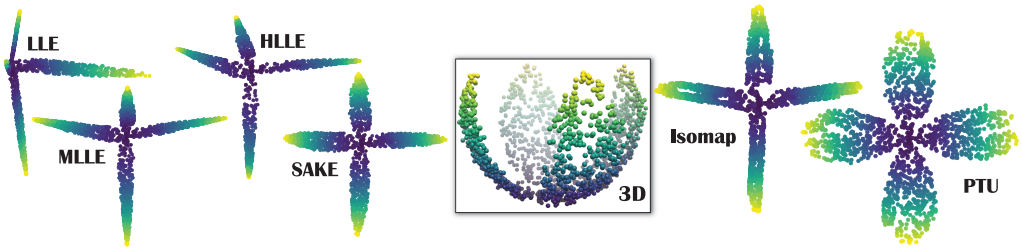


Figure 4.6: **Noisy Petals.** Given a 3D sampling of a 4-petal shaped portion of a sphere (see Fig.4.1) with added Gaussian noise in the normal direction (σ: 3% of sphere radius, middle), PTU recovers an almost perfect quasi-isometric 2D parametrization, while Isomap still fails (right). Local methods, not exploiting large geodesic distances, fail even worse (left) with the notable exception of SAKE that performs better than Isomap.

potential issues with sampling density of the data compared to the local curvature of the manifold.

Connection-based Dijkstra's Algorithm. Finally, we point out that our parallel transport approach to estimating geodesic distances could be done along *approximately-shortest* polyline between a given pair of points: fast approximations of shortest paths could thus be used to lower this $O(n^2 \log n)$ step without significant effects on the results. However, the longer the polyline, the more likely numerical inaccuracies induced by repeated alignment of frame fields will accumulate. Since Dijkstra's algorithm is not the computational bottleneck in Isomap, we decided to utilize graph-based shortest paths: in fact, our construction can be neatly incorporated within the dynamic programming approach that Dijkstra's algorithm uses—i.e., storing the predecessor to each point in the shortest path found thus far. Parallel transport unfolding only requires the insertion of six lines in the original Dijkstra's algorithm in order to reduce the unnecessarily repeated calculations that result from the fact that sub-paths of shortest paths are also shortest paths. Specifically, every time a point is removed from the priority queue (i.e., every time the algorithm finds a shorter path to a point), we can compute the corresponding vector $\mathbf{v}_i$, before storing the cumulative connection (i.e., the product of connection matrices along the path) to be the cumulative connection of the predecessor multiplied by the local discrete connection. Proceeding in this way guarantees that the calculations are performed without redundancies, adding a negligible amount of computational time to the traditional Dijkstra's algorithm. Pseudocode is given in Alg. 4.2.

4.2.6 Final Embedding

Finally, once we evaluated all pairwise geodesic distances, the results can be processed through MDS: double-centering the matrix of *squared* distances produces the Gram matrix, whose partial eigendecomposition returns the final embedding as described in Sec. 2.1.3. Numerical improvements (see [32, 91, 93, 110]) can be applied to reduce the cubic complexity order of this stage. Note that robust alternatives to classical MDS could also be used here (e.g., [41, 114]) to offer resilience to outliers, but at higher computational cost.

4.3 Theoretical Analysis

We conclude this section with a few properties worth mentioning.

Proposition 4.2. *Unlike Isomap, PTU is "linearly precise": assuming the pointset $\mathcal{S}$ samples a linear d-dimensional subspace $\mathcal{R}$ of $\mathbb{R}^D$, and the proximity graph G is such that each sample point has enough neighbors to span a d-dimensional subspace, the PTU embedding $\mathbf{Z}$ is a composition of a projection and a rigid body transformation of $\mathbf{X}$.*

Proof. When the sampled data lie on a linear subspace of dimension d, each orthonormal frame $\mathbb{T}_i$, computed using SVD as described in Sec.4.2.2, forms a basis for $\mathcal{R}$. Moreover, every pair $\mathbb{T}_i$ and $\mathbb{T}_j$ are perfectly aligned by the discrete connection $\mathbf{R}_{j,i}$ (Eq.(4.2)). As a result, unfolding a polyline $(\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_m})$ reduces to rewriting it in the basis $\mathbb{T}_{i_1}$ and our geodesic length estimation recovers the exact (Euclidean) distance between $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_m}$, independent of the sampling irregularities or of the geodesic convexity of the domain. PTU thus becomes equivalent to classical MDS, and produces a d-dimensional embedding $\mathbf{Z}$ that is isometric to $\mathbf{X}$ by Prop. 2.1. □

Proposition 4.3. *Under mild assumptions on the regularity of a manifold $\mathcal{M}$ and assuming that the input pointset $\mathcal{S}$ samples $\mathcal{M}$ finely enough with potential sampling voids over regions of small (sectional) curvature of the manifold, the PTU estimate $d_{PTU}(\mathbf{x}_i, \mathbf{x}_j)$ of the geodesic distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$ based on a Dijkstra (shortest) polyline $(\mathbf{x}_i, ..., \mathbf{x}_j)$ computed on the proximity graph G of $\mathcal{S}$ approximates the real geodesic distance $d_g(\mathbf{x}_i, \mathbf{x}_j)$.*

Discussion: We quantify this proposition more rigorously in App. A by providing a concrete error bound (and its proof) between d_{PTU} and d_g. Our bound relies on three key components: 1) if the pointset $\mathcal{S}$ is dense enough, then for appropriate choice of parameters k and K the discrete tangent spaces approximate their continuous equivalents, and the resulting discrete connection converges to the Levi-Civita

connection in the sampling limit as proved in [168], where the authors used an equivalent notion of parallel transport to define and study a discrete connection Laplacian; 2) sampling voids in $\mathcal{S}$ are allowed as long as the integral of the intrinsic sectional curvature of $\mathcal{M}$ over the regions of the underlying manifold corresponding to these voids is small—in other words, we will assume that a Dijkstra polyline lies within a tubular neighborhood of its corresponding geodesic, for a tubular diameter less than $O(1/\sqrt{\kappa_s})$ where κ_s is the local maximum absolute value of the sectional curvature of $\mathcal{M}$; and 3) for a dense enough sampling of $\mathcal{M}$, a straight $\mathbb{R}^D$ vector between two nearby points $\mathbf{x}_i$ and $\mathbf{x}_j$ on $\mathcal{M}$ has approximately the same length as the Cartan development of the geodesic between them on the tangent space $T_i\mathcal{M}$ at $\mathbf{x}_i$. Note that our statement involves *no geodesic convexity requirement*.

The benefit of using parallel transport over regular Dijkstra geodesic length approximation is thus clear: our construction eliminates spurious geodesic curvature that graph approximations inevitably suffer from, bringing significant improvement even for well sampled domains (see Fig. 4.4). It also allows for almost perfect recovery of pairwise geodesic distances for developable manifolds ($\kappa_s = 0$) with *arbitrary* topology in the sampling limit, even if the sampled manifold is not geodesically convex. When voids are present in pointsets that sample non-developable manifolds, PTU computes approximate geodesic distances without having to explicitly fill in the voids. Instead, it extracts geometric information from paths surrounding each hole to recover high accuracy geodesic estimates on the manifold, provided that the voids were not over regions with large curvature.

Proposition 4.4. *Just like Isomap, the complexity of PTU is* $O(n^3)$.

Proof. Our approach does not change the computational complexity of the various steps compared to Isomap: the proximity graph construction is still $O(n \log n)$, the construction of tangent planes takes $O(n\,D\,K^2)$, our Parallel Transport Unfolding using Dijkstra shortest paths is in $O(n^2 \log n)$ (the unfolding process part itself requires $O(n^2(Dd^2 + d^3)$ to perform matrix multiplications and SVDs), while the partial eigendecomposition of a dense matrix is expected to take $O(n^3)$ operations. □

Additional Control. Compared to Isomap, our parallel transport based approach has a few extra parameters that can be exploited to offer more control over the manifold learning process:

- Because we compute local tangent spaces at each input point, the neighborhood size K used in Isomap can be adapted (either globally or locally) to the input. While using $K = k$ is sufficient in most cases (as we prove in our examples), raising this value can help deal with very noisy inputs as mentioned in Sec. 4.2.1.
- Similarly, local PCA of these neighborhoods may not lead to the best estimations of tangent spaces in extreme cases: robust PCA, or even local averaging of the PCA estimates[1] can help manage large amounts of noise and outliers. Again, we did not have recourse to these variants to prove the robustness of PTU in its default form; but they can be easily incorporated in a practical implementation of our approach to add flexibility.
- Finally, our parallel transport estimation of geodesic lengths assumes the knowledge of the dimension d of the data, just like Isomap requires as well—and many approaches have been proposed to estimate this dimension directly from the data, see, e.g., [115, 148]. Yet, this intrinsic dimension can, in fact, differ from the dimensionality of the visualization one wishes to produce. Fig. 4.7 illustrates this point: dimensionality reduction approaches applied on the MNIST dataset of digits often use a 2D illustration of their results for easy visual display; however, local analysis of the dimensionality of the zero digit image set indicates an intrinsic dimension of $d = 4$, reflecting the variety of ways to pen a zero (slant, thickness, smoothness, ...). While this information cannot be exploited in Isomap if a 2D depiction is desired, PTU can exploit this estimate of d for its parallel transport procedure, but use only the first two eigenvectors of the Gram matrix—essentially showing a 2D projection of a 4D parameterization of the dataset. Fig. 4.11 shows another example of a 2D visualization of an intrinsic parameterization for $d = 3$.

Related Geometric Methods. Note finally that two related works have proposed using parallel transport for data analysis, albeit for different purposes. Vector diffusion maps (VDM [168]) also exploit parallel transport, but focus instead on computing a low-dimensional embedding that preserves *vector diffusion distances* derived from a connection Laplacian, not true geodesics (see their Figs. 6.2 &

[1] As a side note, having a discrete connection makes the local averaging of vectors or frames particularly simple, as neighboring values can be parallel transported to a common point, where a pointwise average is computed.

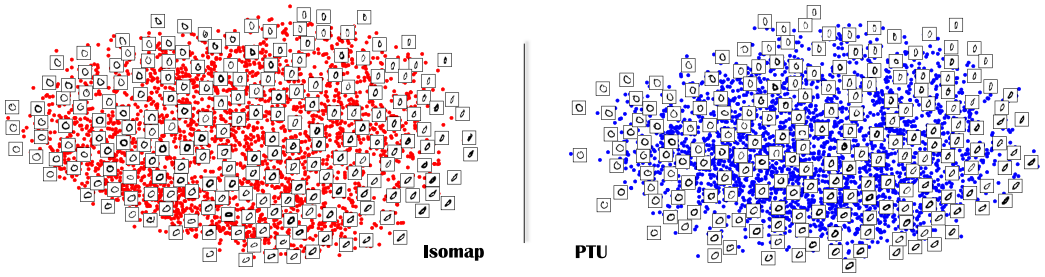


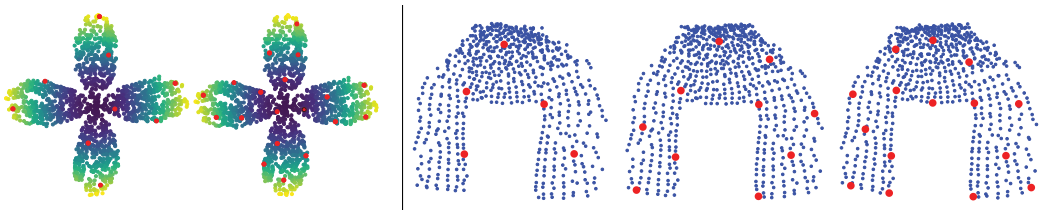
Figure 4.7: **Digit Zero.** The 3000 images of handwritten zeros (with a resolution of 28x28 pixels) from the MNIST dataset are mapped quite similarly by both Isomap and PTU for the first two most significant coordinates (here, $d = 4$ was used). A few images are displayed next to their corresponding 2D point for visualization purposes.

6.3). Parallel vector field embedding (PVF [118]) proposes a different discretization of the connection-Laplacian relying on an extrinsic definition of the covariant derivative. Coordinates of an embedding are constructed via Poisson solves, so that the coordinate lines in $\mathbb{R}^d$ are mapped to parallel vector fields on the original data. Just like our approach, PVF perfectly recovers an isometric parametrization if the manifold is developable, since it is then isometric to a subset of $\mathbb{R}^d$. However, for non-developable manifolds, coordinate lines of the resulting parameterization do *not* correspond to geodesic curves on the original manifold. While they provide a valid definition of *an* embedding, PVF look for a low-dimensional "quasi-parallel" embedding. PTU, instead, targets the same goal as Isomap, i.e., a quasi-isometric mapping for arbitrary sampled manifolds.

4.4 Acceleration via Landmarks

A particularly simple way to accelerate Isomap is through the use of "landmarks" [63], a small fraction of samples of the original pointset: the MDS procedure is applied just to the landmarks to find their quasi-isometric embedding $\mathbb{Z}$ in $\mathbb{R}^d$, before positioning all other points *relative to those landmarks* in linear time, significantly reducing computational complexity. This approach, however, often fails in practice: the sensitivity of the original Isomap method to poor sampling quality makes the low-dimensional embedding of a few landmarks very brittle.

Given its much greater robustness to irregular sampling, PTU is particularly amenable to this landmark-based acceleration without any other alteration than replacing distance estimation by our parallel transport approach. If ℓ landmarks are used, the amount of computations can decrease quite dramatically: the MDS complexity (which was the bottleneck for Isomap and PTU) changes from $O(n^3)$ to $O(\ell^3)$. Let



(a) 9 and 19 landmarks on Petals

(b) 5, 10, and 15 landmarks on letter A

Figure 4.8: **Landmark-PTU.** Combining parallel transport unfolding with the landmark-based approach of Landmark-Isomap [63] drastically reduces computational times, with only small differences in the embedding: (top) for 9 (left) and 19 (right) landmarks, results on the Noisy Petals dataset are visually indistinguishable from the full treatment (see Fig. 4.6); (bottom) for the Letter A dataset, 5 landmarks (left) already capture the proper embedding, but 10 (middle) and 15 (right) landmarks result in a better approximation of the full treatment found in Fig. 4.5 (landmarks are in red).

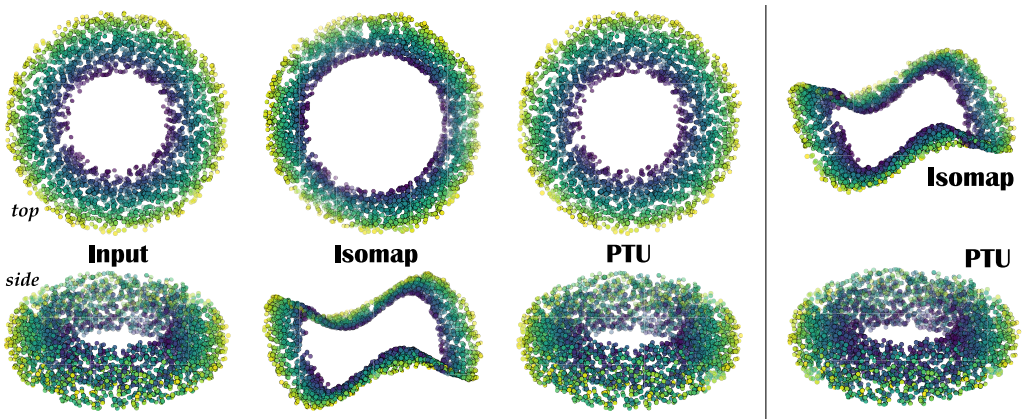
us briefly discuss the implementation of such Landmark-PTU variant.

Setup. From the n original points, we extract ℓ landmarks with $\ell \ll n$, used as a coarse approximation of the input geometry. Landmark selection is not a sensitive part of our approach as long as the landmarks provide a good spatial coverage of the initial pointset. Note also that we compute our discrete connection for the entire pointset, since 1) it is not the computational bottleneck of the original PTU treatment, and 2) we need the distance from every data point to every landmark in order to compute the final embedding anyway. The choice of computing a "full resolution" connection guarantees accurate estimation of geodesic distances even if very few landmarks are used.

From Landmark Embedding to Pointset Embedding. After computing a low-dimensional embedding of the ℓ landmarks using PTU as described in Sec. 4.2 through

$$\mathbb{Z} = \sqrt{\mathbb{\Lambda}_d}\, \mathbb{Q}_d^T,$$

(where now $\mathbb{\Lambda}_d$ and $\mathbb{Q}_d$ are the d largest eigenvalues stored in a diagonal matrix and corresponding eigenvectors of the Gram matrix derived from the $\ell \times \ell$ matrix $\mathbb{D}$ of the *squared* geodesic distances between landmarks only), the embedding of the remaining points can be performed using its pseudoinverse $\mathbb{Z}^\dagger$ and the knowledge of geodesic distances between input points and landmarks. Note that the pseudoinverse of $\mathbb{Z}$ has a simple explicit formulation due to its basic form, thus requiring no



(a) 3D torus in 4D (b) Curved torus in 4D

Figure 4.9: **Tori Embedded in 4D.** (a) From a 3D pointset filling up a toroidal domain that we trivially embedded in 4D by adding an extra constant coordinate, the 3D embedding computed via Isomap leads to an unexpected global distortion due to the manifold being not convex. Instead, PTU is nearly perfect: Isomap has a normalized average relative error of 40.6%, while PTU is 0.06%. (b) A mildly curved torus in 4D is obtained by mapping a set of points $(x, y, z) \in T^2$ to $(x, y, z, (x^2 + y^2)/2)$. Now it has non-zero curvature as a manifold in 4D. 3D Isomap embedding still suffers from similar global distortion (left); PTU recovers the torus very well (right) despite the non-trivial curvature.

additional computations since

$$\mathbb{Z}^\dagger = \mathbb{Q}_d \sqrt{\mathbb{\Lambda}_d}^{-1}$$

. Denote by $\overline{\mathbb{d}}$ the columnwise mean of $\mathbb{D}$ (i.e., $\overline{\mathbb{d}}_p = \frac{1}{\ell} \sum_q \mathbb{D}_{pq}$). Then as described in Landmark-Isomap [63], the position of a non-landmark point $\mathbf{x}_i \in \mathcal{S}$ in the low-dimensional embedding can be directly computed using a vector $\mathbf{d}_i$ of squared geodesic distances from $\mathbf{x}_i$ to the ℓ landmarks through:

$$\mathbf{z}_i = \left(\mathbb{Z}^\dagger\right)^T \left(\overline{\mathbb{d}} - \mathbf{d}_i\right).$$

That is, knowing how distant a point is from the landmarks originally, we deduce its final position based on the low-dimensional embedding of the landmarks in $O(\ell^2)$. Consequently, the proposed Landmark-PTU procedure (consisting of computing all pairwise geodesic distances between ℓ landmarks and all input points, constructing embedding of the landmarks via MDS, and enriching it with non-landmarks through matrix-vector multiplications) brings the complexity down to $O(\ell n \log(n) + \ell n(Dd^2 + d^3) + \ell^3 + \ell^2 n)$. As Fig. 4.8 demonstrates, this simple variant returns nearly the same embeddings as full PTU with as little as 0.1% to 0.5% of the points used as landmarks. More results using Landmark-PTU are available in App. B.

Algorithm 4.1 Dimensionality Reduction via Parallel Unfolding

Input: Pointset $\mathcal{S} = \{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^n$
1: Construct proximity graph G of $\mathcal{S}$ (Sec. 4.2.1)
2: Compute tangent frames $\{\mathbb{T}_i\}_{i=1}^n$ (Sec. 4.2.2)
3: Compute geodesic distances $\mathbf{D}$ via parallel transport (Sec. 4.2.5)
4: Perform MDS on $\mathbf{D}$ to obtain $\mathbf{Z}$ (Sec. 2.1.3)
Output: Low-dimensional embedding $\mathbf{Z} = \{\mathbf{z}_i \in \mathbb{R}^d\}_{i=1}^n$

4.5 Results

We now discuss implementation details and provide a series of tests, on synthetic and real datasets, to compare our parallel transport approach to the original Isomap and other nonlinear dinensionality reduction methods.

4.5.1 Implementation details

Our implementation follows the steps described in the previous sections as summarized in Alg. 4.1. We use a modified Dijkstra's algorithm to compute shortest paths and find geodesic distances concurrently, as detailed in Alg. 4.2 (the only new lines of code to handle connections are in blue). The final partial eigensolve was implemented using the Spectra C++ library [150]. If the dimension D is high (i.e., larger than 100), vectorizing matrix-matrix and matrix-vector multiplications is crucial for efficiency.

4.5.2 Simulated Datasets

We first test the performance of PTU on artificial datasets (embedded in 3D or higher) to quantifiably evaluate its behavior.

Linear Precision. A nonlinear extension to PCA should, at the very least, be linear-precise, i.e., input data lying on a flat d-manifold in $\mathbb{R}^D$ should be isometrically mapped to $\mathbb{R}^d$. From an input pointset densely sampling a flat square shape embedded in 3D space, we compare the performance of Isomap and PTU (with $k = K = 10$) in Fig. 4.2: we visualize the error for each point based on the normalized distance between ground truth and its embedding location. As expected PTU recovers the data exactly (to numerical precision), while Isomap introduces distortions due to its use of graph-based distances. We also use a 4D pointset that randomly samples a 3D torus in Fig. 4.9a, with the embedding space being 3-dimensional. Unlike Isomap, which introduces severe distortion because of the non-convexity of the sampled domain, PTU ($k = K = 10$) perfectly reproduces the torus. This last

Algorithm 4.2 Parallel Transport Dijkstra

Create minimum priority queue P
for $i \in [1, n]$ **do**
  $\mathbf{R}[i] \leftarrow$ identity matrix, $\mathbf{v}[i] \leftarrow 0$
  for all $\mathbf{x}_j$ adjacent to $\mathbf{x}_i$ in G **do**
    $Pred[j] \leftarrow i$
    $dist[j] \leftarrow |\mathbf{x}_j - \mathbf{x}_i|$
    $P.push(dist[j], j)$
  end for
  while not P.isEmpty() **do**
    $\mathbf{x}_r \leftarrow P.$ pop_min()
    $q \leftarrow Pred[r]$
    $\mathbf{U}S\mathbf{V}^T \leftarrow \text{SVD}(\mathbb{T}_q{}^T \mathbb{T}_r)$
    $\mathbf{R}[r] = \mathbf{R}[q] \cdot \mathbf{U}\mathbf{V}^T$
    $\mathbf{v}[r] = \mathbf{v}[q] + \mathbf{R}[q] \cdot \mathbb{T}_q^T(\mathbf{x}_r - \mathbf{x}_q)$ (Eq. (4.4))
    $geo\_dist[r] = |\mathbf{v}[r]|$
    for all $\mathbf{x}_j$ adjacent to $\mathbf{x}_r$ **do**
      $temp\_dist \leftarrow dist[r] + |\mathbf{x}_j - \mathbf{x}_r|$
      if $temp\_dist < dist[j]$ **then**
        $dist[j] = temp\_dist$
        $Pred[j] = r$
        $P.update(dist[j], j)$ ▷ $update()$ inserts $j \notin P$
      end if
    end for
  end while
  $\mathbf{D}[i, 1 : n] \leftarrow geo\_dist[1 : n]$
end for
Symmetrize distance matrix $\mathbf{D} \leftarrow (\mathbf{D} + \mathbf{D}^T)/2$

Output: Pairwise geodesic distance matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$

example also proves that Isomap can significantly distort data (in unexpected ways) in higher dimensions.

Non-geodesically-convex Domains. An S-shaped manifold with a rectangular void (see Fig. 4.3) is well parameterized by PTU, while Isomap introduces spurious distortion associated with biased geodesic distance estimations around the void. Since this S-shaped manifold is isometrically developable, we can compare errors in parameterization on a per-point basis: Isomap reaches 15% of relative error, while PTU (still using $k = K = 10$) stays below 0.2%. The effect of non-geodesically-convex domains is even more pronounced in higher dimensions: Fig. 4.1 shows a 3D pointset sampling four petals from a surface of a sphere in 3D; this pointset is then lifted

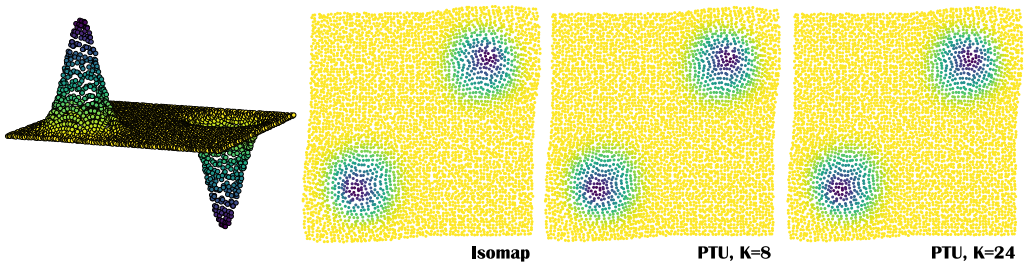


Figure 4.10: **Gaussian Landscape.** For a noise-free dense sampling of a non-developable height field (left), Isomap and PTU return comparable quasi-isometric 2D parameterizations. Using $K = 8$ or $K = 24$ neighbors (right) does not visually affect the result of the PTU embedding.

to $D = 100$ dimensions and rotated by a random orthogonal transformation. The resulting non-developable and non-convex 100D dataset is then embedded in 2D by Isomap, clearly demonstrating that graph-based distances bias results dramatically. Our algorithm, run with $k = K = 10$, correctly unfurls the petal-like set.

Non-developable Manifolds. We also demonstrate our results on highly non-developable manifolds. Fig. 4.10 shows a dense, noise-free pointset that is sampling a 2D height field with two Gaussian bumps. The differences between Isomap and PTU ($k = 8$) are small with such a dense dataset. This example also demonstrates that the size of the geodesic neighborhood used for tangent space estimation does not dramatically affect the quality of results: PTU outputs for $K = 8$ and $K = 24$ are visually indistinguishable. We also test an input pointset sampling a slightly curved 3D torus in 4D: from the 3D torus in Fig. 4.9a, each point (x, y, z) is mapped in 4D to $(x, y, z, (x^2 + y^2)/2)$. While Isomap gets even more distorted, PTU ($k = K = 10$) still produces a toroidal 3D embedding, see Fig. 4.9b.

Geodesic Distance Estimation. We also compare the accuracy of Dijkstra, PTU, and SAKE-corrected (Ch. 3) estimations of geodesic distances, by applying these 3 methods to a low-density, regular sampling of a spherical cap in Fig. 4.4 so that geodesic distances are known analytically. PTU geodesics are over 120 times more accurate than Dijkstra's, and more than 20 times better than the local geodesic correction method used in SAKE (where the whole cap is treated as a single neighborhood).

Sensitivity to Irregular Sampling. For a highly irregular sampling of a simple S shape, the robustness of PTU is shown in Fig. 4.13. Observe that Isomap completely

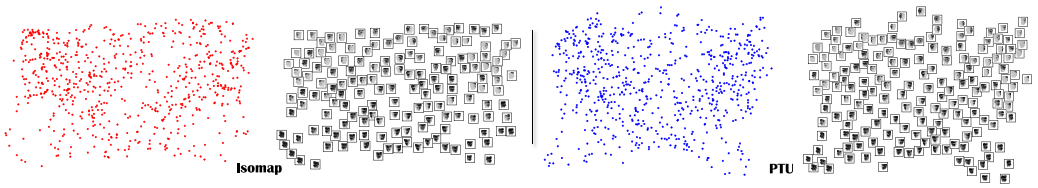
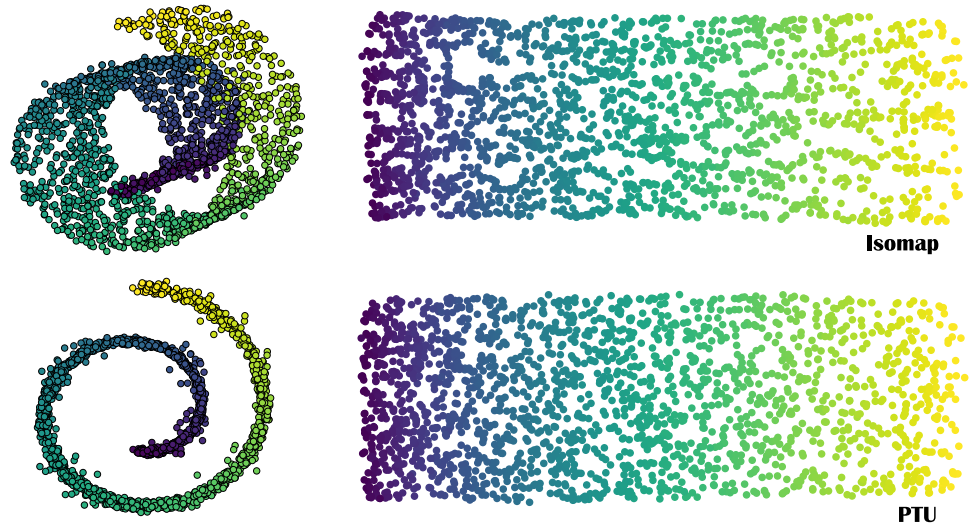


Figure 4.11: **Faces Dataset.** For the classical Faces dataset (known to be, by construction, of intrinsic dimension 3), the first two coordinates of both Isomap and PTU are quite similar. Left: 2D parameterization; right: a few of the face images are shown at their actual positions to better understand the parameterization.

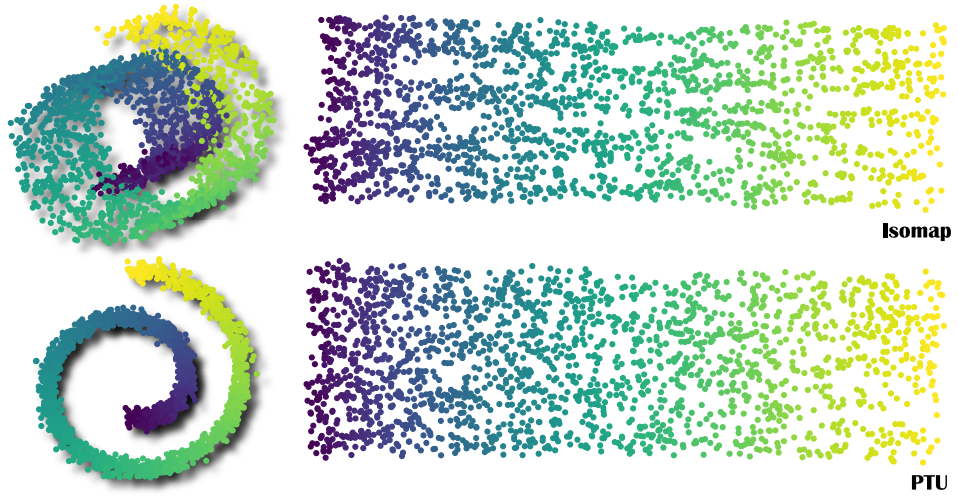
collapses a section of the data in the right bottom corner, while PTU correctly captures the features of the input data even for a non-adapted choice of neighborhood parameters ($k = K = 10$), introducing only small distortion throughout the domain. These two behaviors are visualized by the error plots using a linear color ramp from blue (0% error) to red (10% error).

Sensitivity to Noise. Figs. 4.12a and 4.12b verify that PTU (with $k = 10$) is as resilient to strong noise and outliers as Isomap even without locally adapting neighborhood sizes ($K = 25$ was used in the Gaussian noise case, and $K = 10$ in the case of sparse noise). Local methods such as LLE [155], Hessian LLE [71], LTSA [204], MLLE [203] or SAKE (with $k = 10$ for fairness of comparison) fail to unwrap noisy datasets (see Fig. 4.14) as they do not exploit the geodesic distance estimates between pairs of points that are far apart. Fig. 4.6 shows that adding noise to the petals dataset from Fig. 4.1 does not alter the result of our approach significantly (we used $k = 10, K = 30$); yet Isomap remain unable to unfurl the petals properly, and all local methods fail on this seemingly simple data, at times spectacularly. Please refer to the Supplemental Material in App. B for more systematic testing of the effect of noise levels on both local and global methods; as expected, PTU is systematically as good or better than all other methods.

Timing. A performance analysis of our algorithm shows perfect agreement with the expected time complexity orders: the eigensolve dominates the computational time and scales as $O(n^3)$, parallel transport Dijkstra scales as $O(n^2 \log n)$, and graph construction as $O(n \log n)$, while tangent estimates are linear in n. Examples of timings on an Intel i7 2GHz, 8 GB RAM laptop are 6.4s for the petals parametrization in Fig.4.1,10.9s for the irregular sampling of S-shaped manifold in Fig. 4.13, and 9.8s for the noisy Swiss roll in Fig. 4.12b (with $n = 2000$). Note that these timings



(a) Sparse noise



(b) Gaussian noise

Figure 4.12: **Noisy Swiss Rolls.** (a) If one adds noise to the Swiss Roll dataset in the normal direction by displacing 10% of points with a uniform distribution of amplitude equal to 8% of the max bounding box size and adding a Gaussian noise to the other points with standard deviation equal to 0.4% of the bounding box (see two views of the 3D dataset on the left), Isomap accentuates a few low sampled regions compared to PTU. (b) For a strong Gaussian noise (standard deviation equal to 2% of the max bounding box size), Isomap suffers from clear visual artifacts while PTU returns a good parameterization without the need for robust estimations.

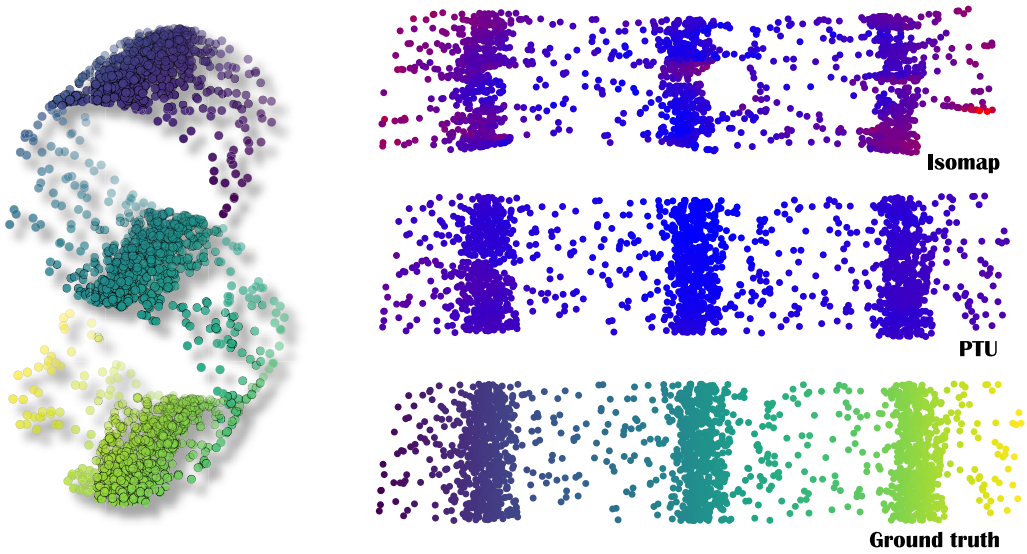


Figure 4.13: **Varying Density S.** For a widely varying density of points (left), Isomap (top right) introduces large spurious distortions, unlike PTU (middle right).

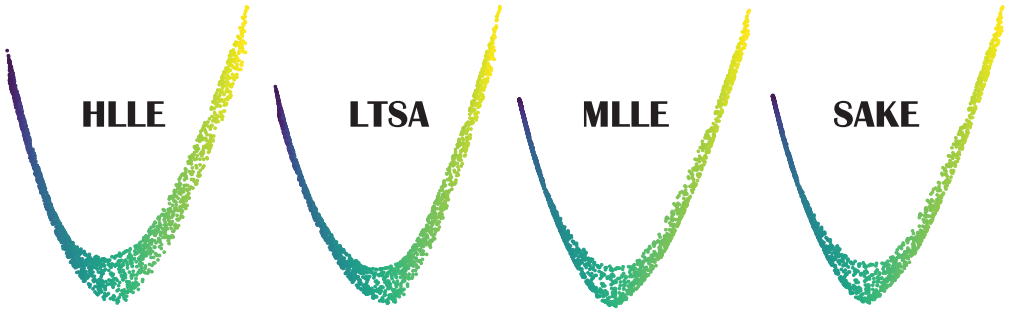


Figure 4.14: **Failure of Local Methods.** For the very noisy Swiss Roll example of Fig. 4.12b, none of the local manifold learning methods returns a decent parameterization, as no large intrinsic distances are exploited.

are for the full-blown MDS procedure, with roughly half of time spent on the other steps. Using the Landmark-PTU variant with 0.1%-0.5% of the points as landmarks improved the efficiency of the MDS step by 500 to 5000 times on tested datasets, with virtually no visual difference compared to the full treatment; see Fig. 4.8 and App. B.

4.5.3 High-dimensional Datasets

We also present results on a number of real and/or high-dimensional datasets. While no ground truth is available for these examples, they allow us to compare PTU and Isomap on inputs that may not even satisfy the (single-chart) manifold assumption.

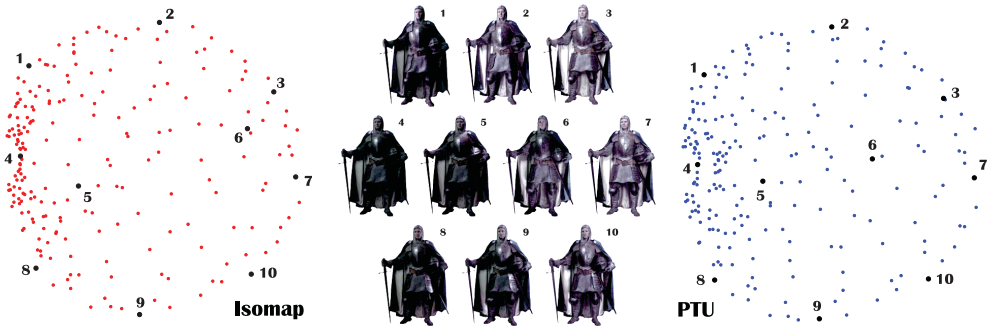


Figure 4.15: **Knights.** From 221 RGB images of 608×456 pixels (10 examples shown on the right) of an actor captured in full costume from different lighting directions, a 2D embedding is computed solely based on local pixel differences using Isomap and PTU (left). Both methods find a parameterization of the images corresponding to lighting direction and intensity (the knight images correspond to black dots).

Faces Dataset. On this classic set of 698 images of 64x64 pixels, PTU and Isomap recover the same two characteristic features of the data: Fig. 4.11 shows that both arrange the images based on the azimuth and elevation of the camera, with a fairly similar global structure (2D visualization, $d=3$, $k=6$, and $K=18$).

Digit zero. When applied to 3000 digit zero images (28x28 pixels) from the MNIST dataset, both PTU and Isomap create a parameterization of the different ways people write a zero, separating left-leaning from right-leaning and circular from oval zeros as shown in Fig. 4.7 (2D visualization, $d=4$, $k=K=10$).

Knights. We tried our approach on the dataset from Ch. 3 of reflectance fields captured using the Light Stage apparatus [117]. A static character (in a knight costume) was captured in 608×456 RGB images under 221 individual lighting directions covering a large sphere of illumination. From this pointset in $\mathbb{R}^{831744}$, Isomap and PTU learn a flat 2D manifold that best fits this high-dimensional dataset. The result for $d=2$ and $k=K=8$, shown in Fig. 4.15, recovers positions related to light angles without any knowledge of the setup; the highly distorted results of local approaches on this dataset can be found in Ch. 3. The black background of each image was removed for clarity.

Letter A. From an input set of 888 images with 120x120 pixels of a rotated and resized letter 'A' , the structure of this intrinsically 2-dimensional manifold in 14400 dimensional space revealed in PTU ($k=8$, $K=32$) and Isomap embeddings in

Fig. 4.5 (left) is very intuitive. However, when a section of the data is removed (right), Isomap suffers from its characteristic global distortion, caused by the presence of the hole; instead, the structure of PTU embedding remains basically the same.

4.6 Conclusion

The use of parallel transport on high-dimensional datasets remedies an important limitation of the Isomap approach: unfolding paths between pairs of points based on the Levi-Civita connection significantly improves the estimation of geodesic distances and removes the restriction for geodesic convexity of the input data. We demonstrated on a series of examples that our approach does indeed recover similar unfolding to Isomap for geodesically convex inputs of low- and high-dimensional data, but neither overestimates geodesic distances if large voids are present, nor suffers from large deformation in the case of non-geodesically convex inputs. This property is particularly crucial to the success of our landmark-based variant, Landmark-PTU, which can efficiently approximate the low-dimensional embedding of large datasets in $O(n^2 \log n)$, even in the presence of noise.

Chapter 5

OPERATOR-ADAPTED WAVELETS FOR DISCRETE DIFFERENTIAL FORMS

5.1 Contributions at a Glance

In this chapter, we adopt a finite-element perspective to the development of an operator-adapted multiresolution analysis. After formulating a series of axiomatic properties to ensure a proper Galerkin discretization tailored to a given continuous, linear, bijective, and self-adjoint positive-definite operator, we introduce a construction of operator-adapted basis functions and associated wavelets that matches the recent construction of gamblets introduced in [140]. However, our resolutely finite-element-based approach extends seamlessly from scalar-valued functions to *differential forms*, the building blocks of Cartan's Exterior Calculus [40] that extends differential and integral calculus to differentiable manifolds in coordinate-free matter. While Finite Element Exterior Calculus [11] and its underlying structure-preserving de Rham complex have been a recent topic of interest in practical computations, the construction of form-based wavelets in this context has received little to no attention, even if wavelets for discrete differential forms could be crucial to the discretization of nonlinear partial differential equations (PDEs) arising in, for instance, fluid dynamics. We thus propose a constructive approach to *multiresolution analysis of finite-element differential p-forms adapted to an exterior operator*. As a result, we introduce the first construction of vector-valued wavelets that block diagonalize a given operator. We also show how to derive operator-adapted wavelets restricted to a linear subspace satisfying a differential constraint, such as divergence-free wavelets. We demonstrate that for time-dependent nonlinear systems (involving complex interactions between multiple scales and domain geometry), our wavelets can be computed on the fly in quasilinear complexity through linearization of the dynamics. In addition to our algorithmic presentation of the general construction of operator-adapted wavelets, we discuss how an operator-adapted multiresolution analysis provides many opportunities to construct reduced models or perform fast numerical integration.

5.2 From Classical to Operator-adapted Wavelet-Based Finite Elements

Before delving into our contributions, we provide background for our work by briefly discussing the usual wavelet-Galarkin approach, and recent proposals on how to improve upon its foundations.

5.2.1 Wavelet-Galerkin Approach

Let H be a subspace of the Sobolev space H^s (defined as the closure of smooth functions compactly supported in an open bounded domain Ω, with respect to the L^2 norm and of total derivatives of order $s \in \mathbb{N}^*$), and let $\mathcal{L} : H \to H^* \subset H^{-s}$ be a continuous, linear, bijective, and self-adjoint positive-definite operator giving rise to a differential equation of the form

$$\mathcal{L}\, u = g. \tag{5.1}$$

A standard procedure to analyze and solve Eq. (5.1) involves expressing it in a weak form through the introduction of a bilinear form (often called "energy scalar product") $L : H^2 \to \mathbb{R}$ with $L(u, v) \coloneqq \langle u, v \rangle_{\mathcal{L}} = \int_\Omega u \cdot \mathcal{L}\, v$ (where product operation represents pointwise inner product) and a linear operator $G : H \to \mathbb{R}$ as $G(v) \coloneqq \langle g, v \rangle_{L^2} = \int_\Omega g \cdot v$. Since $\langle u, v \rangle_{\mathcal{L}}$ defines an inner product on H, the bilinear form is both continuous (i.e., $L(u, v) \leq C||u||_{\mathcal{L}}||v||_{\mathcal{L}}$) and coercive (i.e., $L(u, u) \geq C^{-1}||u||^2_{\mathcal{L}}$). As a result, the celebrated Lax-Milgram theorem implies that the weak problem

$$L(u, v) = G(v), \quad \forall v \in H \tag{5.2}$$

has a *unique* solution for any right-hand side $g \in H^*$ of Eq. (5.1).

The classical wavelet-Galerkin approach uses a compactly supported L^2-orthogonal functional basis [6] to solve Eq. (5.2). It proceeds by approximating the subspace H with a finite-dimensional solution space $\mathcal{V}^q$ and applying the multiresolution construction as follows. First, a nested sequence of lower-dimensional (coarser) functional spaces $\{\mathcal{V}^k\}_{k=1}^{q-1} \subset H$ are defined such that

$$\mathcal{V}^k \subset \mathcal{V}^{k+1} \text{ for } k = 1 \ldots q-1.$$

Complementary wavelet spaces $\{\mathcal{W}^k\}_{k=1}^{q-1} \subset H$ are then derived by enforcing

$$\mathcal{V}^{k+1} = \mathcal{V}^k \oplus \mathcal{W}^k \text{ for } k = 1 \ldots q-1, \tag{5.3}$$

where $\oplus$ denotes the direct sum of L^2-orthogonal subspaces. Consequently, $\mathcal{V}^q$ admits a multiresolution decomposition:

$$H \approx \mathcal{V}^q = \mathcal{V}^1 \oplus \mathcal{W}^1 \oplus \mathcal{W}^2 \oplus \ldots \mathcal{W}^{q-2} \oplus \mathcal{W}^{q-1}$$

On each resolution level k (corresponding to $\mathcal{V}^k$), suppose we have n_k functions $\{\varphi_i^k\}_{i=1}^{n_k}$ forming a basis of $\mathcal{V}^k$ and N_k (pre-)wavelets $\{\psi_j^k\}_{j=1}^{N_k}$ forming a basis of the wavelet space $\mathcal{W}^k$, with $N_k \coloneqq n_{k+1} - n_k$ due to Eq. (5.3). (Throughout this chapter, we will not require the wavelet basis elements of a fixed resolution k to be orthonormal; thus formally, $\{\psi_i^k\}_{i=1}^{N_k}$ are only pre-wavelets, although we will refer to them as wavelets for simplicity). The solution of Eq. (5.2) can now be approximated by a function u^q of the form:

$$u^q = \sum_{i=1}^{n_1} v_i^1 \varphi_i^1 + \sum_{k=1}^{q-1} \sum_{j=1}^{N_k} w_j^k \psi_j^k. \tag{5.4}$$

Let $N \coloneqq n_1 + \sum_{k=1}^{q-1} N_k = n_q$ be the total number of degrees of freedom associated with the wavelet decomposition of $\mathcal{V}^q$, and let $\mathbf{w}$ be the vector of coefficients $(v^1, w^1, \ldots, w^{q-1})$ of u^q of size N. The finite-dimensional weak problem can be formulated as a linear equation:

$$\mathbf{L}\mathbf{w} = \mathbf{g}, \tag{5.5}$$

where $\mathbf{L}$ is the global $N \times N$ stiffness matrix

$$\mathbf{L} = \begin{bmatrix} A^1 & M^{(1,2)} & \ldots & M^{(1,q)} \\ M^{(2,1)} & B^1 & \ldots & M^{(2,q)} \\ \vdots & \vdots & \ddots & \vdots \\ M^{(q-1,1)} & M^{(q,2)} & \ldots & B^{q-1} \end{bmatrix}$$

composed of an $n_1 \times n_1$ submatrix $A_{ij}^1 \coloneqq L(\varphi_i^1, \varphi_j^1)$ encoding the stiffness matrix of the coarsest basis functions, of a series of $N_k \times N_k$ submatrices $B_{ij}^k \coloneqq L(\psi_i^k, \psi_j^k)$ that are the stiffness submatrices of wavelets for each resolution level, and the off-diagonal block matrices $M^{(r,s)}$ describing interactions between bases of different resolution levels r and s:

$$M_{ij}^{(1,s)} \coloneqq L(\varphi_i^1, \psi_j^s) = M_{ji}^{(s,1)} \text{ for } s > 1,$$
$$M_{ij}^{(r,s)} \coloneqq L(\psi_i^r, \psi_j^s) \text{ for } r, s > 1.$$

The N-dimensional right-hand side vector $\mathbf{g}$ is defined in a similar fashion: if $g_i^1 \coloneqq G(\varphi_i^1)$ and $b_j^k \coloneqq G(\psi_j^k)$ for $k = 1 \ldots q-1$, then

$$\mathbf{g} = \begin{bmatrix} g^1 \\ b^1 \\ \vdots \\ b^{q-1} \end{bmatrix}.$$

This problem also admits a unique solution, inheriting this property from the continuous formulation since $\mathcal{V}^q \subset H$. Solving the linear system in Eq. (5.5), one obtains the set of coefficients $\mathbf{w}$ from which the approximate solution u^q is constructed through Eq. (5.4). This approach provides the same solution as if one had used a Galerkin approach at the finest resolution $\mathcal{V}^q$, just expressed in a different basis. The use of different resolutions brings, however, several advantages over the single-resolution system of equations: applications such as preconditioning and adaptive mesh refinement can selectively adapt the number of wavelet coefficients to utilize. Yet the matrix $\mathbf{L}$ is difficult to deal with in practice. It is a large matrix with typically deteriorating condition numbers for large values of q (i.e., fine meshes). This issue renders the wavelet-Galerkin discretization not substantially better than the classical Galerkin discretization: in essence, many of the core properties of wavelets that make them extremely useful in signal processing are not as relevant in the FEM context. In other words, the performance of wavelet-based methods based on a multiresolution L^2-orthogonal decomposition of the solution space [22, 33, 73] is affected by the regularity of wavelet coefficients because the decomposition is not inherently adapted to the underlying PDE.

5.2.2 Towards Operator-adapted Wavelets

Shortcomings of the wavelet-Galerkin approach have generated a large body of work aimed at alleviating the issues in solving Eq. (5.5) through the use of preconditioning, specific choices of wavelets to ease evaluation, small subsets/subbands of wavelets, etc. However, despite a series efforts to link wavelets to operators [56, 127], the foundations remained mostly unchanged in practical applications.

Operator-based Multiresolution Decomposition. In a sharp departure from the various palliative measures, Sudarshan [178] proposed to alter the nature of the multiresolution decomposition itself. He argued for what he called *scale-orthogonal wavelets*, i.e., FEM wavelets that should:

1. be operator-orthogonal to block diagonalize the operator at play;
2. produce well-conditioned stiffness matrices (i.e., A^1 and B^k for $k = 1 \ldots q-1$ in the previous example);
3. be localized or have fast decay for computational efficiency.

Property (a) implies a new multiresolution decomposition, no longer based on L^2 orthogonality, but on an operator-dependent notion of orthogonality: wavelets in $\mathcal{W}^k$ should be $\mathcal{L}$-orthogonal to the basis functions of their own resolution and to all other wavelets. This implies that we now need to enforce:

$$\mathcal{V}^{k+1} = \mathcal{V}^k \oplus_{\mathcal{L}} \mathcal{W}^k,$$

where $\oplus_{\mathcal{L}}$ denotes direct sum of $\mathcal{L}$-orthogonal subspaces; the induced multiresolution decomposition becomes:

$$H \approx \mathcal{V}^q = \mathcal{V}^1 \oplus_{\mathcal{L}} \mathcal{W}^1 \oplus_{\mathcal{L}} \dots \mathcal{W}^{q-2} \oplus_{\mathcal{L}} \mathcal{W}^{q-1}.$$

Numerical Consequence of an Operator-adapted Decomposition. A wavelet-Galerkin treatment using such a decomposition would result in a significantly simpler linear solve: discretizing the weak problem in Eq. (5.2) now produces a linear system with the same structure as shown in Eq. (5.5), but the off-diagonal "inter-level" stiffness matrices all *vanish*, i.e. $M^{(r,s)} = \mathbf{0}$ for $r \neq s$. In other words, the matrix $\mathbf{L}$ of the final linear system becomes *block diagonal*. Consequently, the full linear system reduces to a set of q small independent linear equations $B^k w^k = b^k$ for $k = 1 \dots q-1$ and $A^1 v^1 = g^1$, each corresponding to its own resolution level. Property (b) additionally guarantees that each of these linear systems can be solved easily and reliably (Riesz stability). Finally, the last property (c) ensures that within a given resolution level, only nearby basis elements have non-zero contributions, rendering the matrices B^k sparse, and allowing for highly efficient storage as well as fast computations. As with classical wavelets, one obtains a frequency-based decomposition of the solution u^q, except that now this representation is truly adapted to the operator: the coarser basis functions derive from finer scales, changing their shapes accordingly to capture the eigenspaces of the corresponding operator as well as possible while staying spatially localized.

Wavelet Construction. However, scale-orthogonal wavelets are not nearly as simple to construct as their use is desirable. A construction enforcing only (a) and (b) was proposed in [178]. A second construction was also offered in the same publication, this time by relaxing the exact $\mathcal{L}$-orthogonality of the decomposition, thus violating (a). However, this looser definition of wavelets leads to a construction involving Gram-Schmidt orthogonalizations of dense matrices, failing to deliver a fast algorithm except in restricted cases such as when the solution has very few non-zero wavelets coefficients. Recently, Owhadi [140] presented the first practical technique

to achieve all three properties (a)-(b)-(c), introducing operator adapted wavelets referred to as gamblets (due to their Bayesian and game theoretic interpretation). Gamblets have been used over the last year for alleviating the complexity bottleneck associated with parabolic and hyperbolic PDEs [141], dense kernel matrices [158], denoising PDEs [200] and eigenpairs computation [197].

5.2.3 Outline

In this paper, we revisit the recent work of Owhadi [140] and rederive his approach through an axiomatic derivation from a Finite Element Analysis point of view. We then demonstrate that this construction is general enough to handle functional spaces of differential forms based on Whitney basis functions [10, 25, 67] as well, thus offering a multiresolution analysis of arbitrary p-forms adapted to a given exterior operator. With this framework, one can construct novel computational tools such as vector-valued wavelets and divergence-free vector-valued basis functions adapted to any continuous, linear, bijective, and symmetric positive-definite operator. We also provide a detailed algorithmic description of our resulting construction of operator-adapted basis functions and wavelets of differential forms, and review how to use them for multiresolution analysis by leveraging the fact that each sub-band of the decomposition can be handled independently. Finally, we demonstrate the effectiveness of this fast computational procedure for numerical tasks such as coarse graining and model reduction of linear and nonlinear partial differential equations.

5.3 Axiomatic Construction of Operator-adapted Wavelets

We present an algorithmic construction of operator-adapted wavelets through an axiomatic derivation formulated from a finite element perspective. The approach proceeds in a fine-to-coarse fashion, with locality and operator-orthogonality imposed directly based on simple linear algebra. For completeness as well as to offer a deeper insight into the optimality properties of our construction, we also provide (in the Appendix) a short summary of the alternative interpretation from [140] derived from a game-theoretical view of functional approximation and conditional generalized Gaussian processes.

5.3.1 Canonical Multiresolution Analysis

Our approach starts from an existing set of refinable basis functions whose spans form multiscale approximation spaces. Each basis function in this sequence of q nested approximation spaces is associated with an element of a mesh $\mathcal{M}^k$ $(1 \leq k \leq q)$,

where meshes $\mathcal{M}^1, \mathcal{M}^2, ..., \mathcal{M}^q$ provide increasingly finer spatial discretizations of the domain Ω.

Refinable Basis Functions. Let $\{\mathcal{V}^k\}_{k=1}^q \subset H^*$ be a nested sequence of q functional spaces, where each space $\mathcal{V}^k$ is spanned by n_k compactly supported basis functions φ_i^k, i.e., $\mathcal{V}^k = \text{span}\,\{\varphi_i^k\}_{i=1}^{n_k}$. Each basis function φ_i^k is refinable for $1 \leq k < q$, that is to say, φ_i^k can be written as a weighted sum of finer basis functions $\{\varphi_j^{k+1}\}_{j=1}^{n_{k+1}}$:

$$\varphi_i^k = \sum_{j=1}^{n_{k+1}} \mathbf{C}_{ij}^k \, \varphi_j^{k+1}. \tag{5.6}$$

Note that this relation between basis functions of two successive levels implies a specific topological refinement of the mesh $\mathcal{M}^k$ at level k into a finer mesh $\mathcal{M}^{k+1}$. We do not make any special assumption on how these meshes are related (for instance, nested or not) as our construction is very general; as a consequence, we will rarely refer to these meshes, even if the definition of the basis functions are often expressed, explicitly or implicitly, in terms of barycentric coordinates over the elements of these meshes in practice. We do assume, however, that the refinement matrices $\mathbf{C}^k$ defining the basis functions are *sparse*—and as we are about to see, we also require that there exists a sparse parametrization of their *kernels*. To distinguish these (user-specified) canonical basis functions from the operator-adapted basis functions we will construct at each level of resolution, we refer to $\{\varphi_i^k\}_{i=1}^{n_k}$ as *test* (or measurement) *functions* in the remainder of this paper: they will only be used to integrate against at the finest level to kickstart our construction of an operator-adapted multiresolution analysis.

Remark 1: Such a nested, or hierarchical, representation of the functional space $\mathcal{V}^q$ is typical of Multiresolution Analysis (MRA) and wavelet-based adaptive solvers to allow for efficient handling of local refinements of the solution: our test functions are simply their 'scaling functions' (or 'father wavelets') — and any existing such function can be used in our construction. In practice, the finest functional space $\mathcal{V}^q$ giving rise to the hierarchy is formed by the span of n_q linearly independent functions in L^2. The L^2-orthogonal complement of $\mathcal{V}^k$ in $\mathcal{V}^{k+1}$ is typically called $\mathcal{W}^k$, which is spanned by N_k wavelets $\{\psi_i^k\}_{i=1}^{N_k}$, with $N_k \coloneqq n_{k+1} - n_k$ to capture the details at level k, i.e., the functions that are contained in $\mathcal{V}^{k+1}$, but not in $\mathcal{V}^k$. Note that only the test functions are needed in our construction, although their associated wavelets are implicitly present as well, as we will discuss when relevant.

Refinement Kernel. From a refinement matrix $\mathbf{C}^k$, we define a sparse, $N_k \times n_{k+1}$ *refinement kernel matrix* $\mathbf{W}^k$ satisfying:

$$\mathbf{C}^k \, \mathbf{W}^{k,T} = \mathbf{0}_{n_k \times N_k}.$$

In other words, the N_k rows of $\mathbf{W}^k$ form a basis for $\operatorname{Ker} \mathbf{C}^k$, hence our naming convention. This matrix should only have a few non-zero entries per row, corresponding to nearby test functions in the physical domain. In addition, the condition number of $\mathbf{W}^k$ should be as close to 1 as possible to achieve lower condition numbers for the operator-adapted stiffness matrices we will construct (corresponding to the B^k matrices in the previous section).

Remark 2: If the test (scaling) functions φ_i^k were L^2-orthonormal, the matrix $\mathbf{W}^k$ would correspond to the refinement relation of the *wavelets* associated to the scaling functions, i.e.,

$$\psi_i^k = \sum_{j=1}^{n_{k+1}} \mathbf{W}_{ij}^k \, \varphi_j^{k+1}.$$

Indeed, for a typical pair of mother and father wavelets, one has $\langle \varphi_i^k, \psi_j^k \rangle_{L^2} = \sum_{r,s} \mathbf{C}_{ir}^k \, \langle \varphi_r^{k+1}, \varphi_s^{k+1} \rangle_{L^2} \, \mathbf{W}_{sj}^{k,T}$. So if the mass matrix at level $k+1$ of the test functions is the identity, the wavelets form precisely the L^2-orthogonal subspace $\mathcal{W}^k$ mentioned in Rmk. 1. However, our construction does *not* require L^2-orthonormality of the test functions, so the refinement kernel matrices $\mathbf{W}^k$ do not directly define L^2-orthogonal wavelets: these matrices will be used instead to *parameterize the kernel of* $\mathbf{C}^k$ in order to construct our operator-adapted basis functions and wavelets.

5.3.2 Setup for Operator-adapted Basis Functions and Wavelets

From the refinement equations of the user-selected test functions and a choice of refinement kernel matrices, we wish to bootstrap a fine-to-coarse construction of operator-adapted basis functions and associated wavelets. We begin by listing a minimal set of basic, desirable axiomatic properties that are formulated to enforce that the new basis functions and wavelets define an operator-adapted multiresolution analysis of the solution space, and are easy to construct via linear algebra.

Cardinality of Degrees of Freedom. Our first requirement is that our operator-adapted multiscale decomposition must share the *same* cardinality as the canonical multiresolution analysis based on the test functions φ_i^k. Consequently, for each level

Table 5.1: Summary of the main symbols used in our paper and their definitions.

Symbols	Meaning
$\mathcal{V}^k$	Space of test functions on resolution level k, $\dim \mathcal{V}^k = n_k$.
$\{\varphi_i^k\}_{i=1}^{n_k}$	Basis of the space of test functions $\mathcal{V}^k$.
$\mathbf{C}^k$	Refinement $n_k \times n_{k+1}$ matrix for test functions, with $\varphi_i^k = \sum_{j=1}^{n_k} \mathbf{C}_{ij}^k \varphi_j^{k+1}$.
$\mathcal{V}^k$	Operator-adapted solution space on resolution level k, $\dim \mathcal{V}^k = n_k$.
$\{\varphi_i^k\}_{i=1}^{n_k}$	Operator-adapted basis functions spanning solution space $\mathcal{V}^k$.
$\mathbb{C}^k$	Refinement $n_k \times n_{k+1}$ matrix for operator-adapted basis functions, with $\varphi_i^k = \sum_{j=1}^{n_k} \mathbb{C}_{ij}^k \varphi_j^{k+1}$.
$\mathcal{W}^k$	Operator-adapted wavelet space on level k, with $\mathcal{V}^{k+1} = \mathcal{V}^k \oplus_{\mathcal{L}} \mathcal{W}^k$, where $\dim \mathcal{W}^k = N_k := n_{k+1} - n_k$.
$\{\psi_i^k\}_{i=1}^{N_k}$	Operator-adapted wavelet basis spanning wavelet space $\mathcal{W}^k$.
$\mathbf{W}^k$	Refinement kernel $N_k \times n_{k+1}$ matrices with $\mathbf{C}^k \mathbf{W}^{k,T} = 0$, defining operator-adapted wavelets as $\psi_i^k = \sum_{j=1}^{N_k} \mathbf{W}_{ij}^k \varphi_j^{k+1}$.
$\mathbb{A}^k$	Stiffness matrix of size $n_k \times n_k$ for operator-adapted basis functions on level k, defined as $\mathbb{A}_{ij}^k = L(\varphi_i^k, \varphi_j^k)$.
$\mathbb{B}^k$	Stiffness matrix of size $N_k \times N_k$ for operator-adapted wavelet basis on level k, defined as $\mathbb{B}_{ij}^k = L(\psi_i^k, \psi_j^k)$.
u^q	Finite-element solution of $\mathcal{L}u = g$ (Eq.(5.1)) using finest solution space $\mathcal{V}^q$: $u^q = \sum_{i=1}^{n_1} v_i^1 \varphi_i^1 + \sum_{p=1}^{q-1} \sum_{i=1}^{N_p} w_i^p \psi_i^p$
u^k	Level-k approximation of u^q: $u^k \in \mathcal{V}^k$ and $u^k = \sum_{i=1}^{n_1} v_i^1 \varphi_i^1 + \sum_{p=1}^{k-1} \sum_{i=1}^{N_p} w_i^p \psi_i^p$
ω^k	Contribution to FEM solution from level-k wavelets: $\omega^k \in \mathcal{W}^k$ and $\omega^k = u^{k+1} - u^k = \sum_{i=1}^{N_k} w_i^k \psi_i^k$

$k = 1 \ldots q$, we need to construct n_k operator-adapted basis functions, which we will refer to as $\{\varphi_i^k\}_{i=1}^{n_k}$, spanning a functional space that we will denote $\mathcal{V}^k$.

Collocation of Degrees of Freedom. One of the key properties of MRA is spatial localization: unlike operator eigenfunctions that are not attached to any particular spatial location, basis functions are always associated with a particular element (which, consequently, makes the wavelets localized as well). We exploit the existence of the spatially-localized canonical test functions φ_j^k by imposing a weak collocation of the operator-adapted basis functions (we use δ_{ij} to denote the Kronecker delta):

$$\langle \varphi_i^k, \varphi_j^k \rangle_{L^2} = \delta_{ij} \quad \text{for all } \; i = 1 \ldots n_k, \; j = 1 \ldots n_k, \; k = 1..q, \tag{5.7}$$

which will ensure that our operator-adapted basis functions φ_i^k are similarly located in the domain.

Operator-adapted Refinability. The new basis functions φ_i^k should be refinable to induce a hierarchy of nested functional spaces. We denote by $\mathbb{C}$ the corresponding (unknown) refinement matrices in order to impose the following refinement relation between levels:

$$\varphi_i^k = \sum_{j=1}^{n_{k+1}} \mathbb{C}_{ij}^k \, \varphi_j^{k+1}. \tag{5.8}$$

Operator-adapted Wavelets. We wish to construct wavelets as well, associated with the operator-adapted basis functions. Let $\{\psi_i^k\}_{i=1}^{N_k}$ be the $N_k \coloneqq n_{k+1} - n_k$ (pre-)wavelets defined at each level k, and let $\mathcal{W}^k$ the corresponding functional spaces they span. These wavelets should be $\mathcal{L}$-orthogonal to the operator-adapted basis functions of the same level, to enforce that $\mathcal{V}^{k+1} = \mathcal{V}^k \oplus_{\mathcal{L}} \mathcal{W}^k$; consequently, one must have

$$\langle \varphi_i^k, \psi_j^k \rangle_{\mathcal{L}} = 0 \quad \text{for all } i = 1 \dots n_k,\ j = 1 \dots N_k,\ k = 1 \dots q-1. \tag{5.9}$$

Wavelet Refinability. Since we assumed that $\mathcal{W}^k \subset \mathcal{V}^{k+1}$, our adapted wavelets at level k must also be a linear combinations of adapted basis functions at level $k+1$. We require that the corresponding wavelet refinement matrices are, in fact, the *refinement kernel matrices* $\mathbf{W}^k$, that is,

$$\psi_i^k = \sum_{j=1}^{n_{k+1}} \mathbf{W}_{ij}^k \, \varphi_j^{k+1}. \tag{5.10}$$

This particular choice of refinement matrix for our wavelets allows us to leverage the kernel of the canonical refinement on which we derive the operator-adapted decomposition.

Stiffness Matrices. By analogy to the L^2-orthogonal case described earlier, we denote by $\mathbb{A}^k$ the stiffness matrix of our new basis functions, and by $\mathbb{B}^k$ the stiffness matrix of the new wavelets. That is,

$$\mathbb{A}_{ij}^k = L(\varphi_i^k, \varphi_j^k), \qquad \mathbb{B}_{ij}^k = L(\psi_i^k, \psi_j^k)\,. \tag{5.11}$$

Note finally that our notational conventions were selected such that bold letters denote user-selected entities, while outlined letters refer to entities we must construct. In particular, the operator-adapted refinement matrices $\mathbb{C}^k$ are the key ingredients: computing them allows for the construction of adapted basis functions, which in turn

leads to the construction of their adapted wavelets, and thus of the induced stiffness matrices; see Table 5.1 for a review of the various symbols and their definitions.

5.3.3 Resulting Linear Algebraic Conditions

From the axiomatic characterization of our $\mathcal{L}$-adapted multiresolution analysis, we can easily derive linear algebra conditions on the matrices involved in our construction so that Properties (a)-(c) from Section 5.2.2 are satisfied. Indeed,

- the collocation conditions expressed in Eq. (5.7) directly imply that
$$\mathbb{C}^k \, \mathbf{C}^{k,T} = \mathbb{I}_{n_k}; \tag{5.12}$$
- due to Eq. (5.10), the $\mathcal{L}$-orthogonality conditions expressed in Eq. (5.9) can be rewritten as
$$\mathbb{C}^k \, \mathbb{A}^{k+1} \, \mathbf{W}^{k,T} = \mathbf{0}_{n_k \times N_k}; \tag{5.13}$$
- finally, the stiffness matrices defined in Eq. (5.11) are linked through $\mathbf{W}^k$ due to Eq. (5.10):
$$\mathbb{B}^k = \mathbf{W}^k \, \mathbb{A}^{k+1} \, \mathbf{W}^{k,T}. \tag{5.14}$$

From these conditions, we can establish a closed-form expression for $\mathbb{C}^k$ satisfying all of our requirements:

Proposition 5.1. *To enforce Eqs.* (5.12)*,* (5.13)*, and* (5.14)*, one must have*

$$\mathbb{C}^k = \mathbf{C}^{k,\dagger} \left[\mathbb{I}_{n_{k+1}} - \mathbb{A}^{k+1} \, \mathbf{W}^{k,T} \left(\mathbb{B}^k \right)^{-1} \mathbf{W}^k \right], \tag{5.15}$$

where $\mathbf{C}^{k,\dagger} = (\mathbf{C}^k \mathbf{C}^{k,T})^{-1} \mathbf{C}^k$ *is the pseudoinverse of* $\mathbf{C}^k$.

Proof. Given that $\mathbb{C}^k \, \mathbf{C}^{k,T} = \mathbb{I}_{n_k}$ (Eq. (5.12)), the refinement matrices $\mathbb{C}^k$ must be of the form

$$\mathbb{C}^k = \mathbf{C}^{k,\dagger} + \mathbb{Z} \, \mathbf{W}^k,$$

where $\mathbf{C}^{k,\dagger}$ denotes the pseudoinverse of $\mathbf{C}^k$ and $\mathbb{Z}$ is an arbitrary $n_k \times N_k$ matrix: indeed, we saw in Section 5.3.1 that the rows of $\mathbf{W}^k$ form a basis for $\operatorname{Ker} \mathbf{C}^k$ so that $\mathbf{W}^k \mathbf{C}^{k,T} = \mathbf{0}$. Additionally, we must have $\mathbb{C}^k \, \mathbb{A}^{k+1} \, \mathbf{W}^{k,T} = \mathbf{0}$ (Eq. (5.13)), implying that

$$(\mathbf{C}^{k,\dagger} + \mathbb{Z} \, \mathbf{W}^k) \, \mathbb{A}^{k+1} \, \mathbf{W}^{k,T} = \mathbf{0} \quad \Longrightarrow \quad \mathbb{Z} = -\mathbf{C}^{k,\dagger} \mathbb{A}^{k+1} \mathbf{W}^{k,T} \left(\mathbf{W}^k \mathbb{A}^{k+1} \mathbf{W}^{k,T} \right)^{-1}.$$

Substituting $\mathbb{B}^k$ for $\mathbf{W}^k \mathbb{A}^{k+1} \mathbf{W}^{k,T}$ (Eq. (5.14)) and plugging $\mathbb{Z}$ back into the expression of $\mathbb{C}^k$ leads to Eq. (5.15). □

5.3.4 Computational Procedure for Operator-adapted Approach

Based on the closed-form expression of the refinement matrices $\mathbb{C}^k$, we can now derive the computational procedure to turn the weak form in Eq. (5.5) on the finest level $\mathcal{V}^q$ into q independent and well-conditioned linear systems through our $\mathcal{L}$-adapted MRA.

Bootstrapping the Finest Scale. In order to start our construction, we need to select the basis functions $\{\varphi_i^q\}_{i=1}^{n_q}$ at the finest level. We pick

$$\varphi_i^q = \varphi_i^q \tag{5.16}$$

for simplicity, as it trivially enforces that these initial fine basis functions are attached to the same mesh elements as their canonical equivalents. Note that this choice implicitly amounts to rescaling the test functions at the finest scale through the inverse of their mass matrix; using such rescaled test functions as finest level basis functions is also a valid alternative as it formally enforces Eq. (5.7), but we follow the former approach because it does not involve the mass matrix inverse and is more straightforward to implement; one can also see our choice as a simple way to ensure that the finest basis functions are collocated with the test functions without actually having to enforce Eq. (5.7) explicitly. The resulting stiffness matrix $\mathbb{A}^q$ is then trivially evaluated as it is precisely the usual finite-element stiffness matrix $\mathbf{A}^q$, and the right hand side of the original problem is represented as a vector g^q obtained by integrating g against these finest basis functions, i.e., $g_i^q = G(\varphi_i^q)$.

Fine-to-coarse Evaluation. Given our setup and the closed-form expression of the key refinement matrix, we can now describe a transition to a coarser level k (from $k = q$, to $k = 2$) by evaluating relevant matrices and vectors in the following order:

$$\begin{cases} \mathbb{B}^{k-1} = \mathbf{W}^{k-1}\mathbb{A}^k\mathbf{W}^{k-1,T} \\ \mathbb{C}^{k-1} = \mathbf{C}^{k-1,\dagger}\left[\mathbb{I}_{n_k} - \mathbb{A}^k\mathbf{W}^{k-1,T}\left(\mathbb{B}^{k-1}\right)^{-1}\mathbf{W}^{k-1}\right] \\ \mathbb{A}^{k-1} = \mathbb{C}^{k-1}\mathbb{A}^k\mathbb{C}^{k-1,T} \\ g^{k-1} = \mathbb{C}^{k-1}g^k \\ b^{k-1} = \mathbf{W}^{k-1}g^k. \end{cases} \tag{5.17}$$

Resulting Weak Form. Once these evaluations have been done all the way to level 1, the weak form in Eq. (5.5) is, by construction, equivalent to q independent (and, as detailed later, well-conditioned) linear systems:

$$\begin{cases} \mathbb{A}^1 v^1 = g^1 \\ \mathbb{B}^k w^k = b^k \quad \text{for } k = 1 \dots q-1 \end{cases} \tag{5.18}$$

from which the *level-k FEM solution* $u^k \in \mathcal{V}^k$ for $k = 2 \dots q$ can then be evaluated via

$$u^k = \sum_{i=1}^{n_1} v_i^1 \varphi_i^1 + \sum_{p=1}^{k-1} \sum_{i=1}^{N_p} w_i^p \psi_i^p, \tag{5.19}$$

for $k = 2 \dots q$; compare to the large linear system in Eq. (5.5) using the usual L^2-orthogonal MRA. In particular, due to the independence of the linear systems in Eq. (5.18), one can easily compute approximations of the fine solution u^q by solving only the first k linear systems (instead of the full q linear systems) and returning u^k using Eq. (5.19). Observe also, that formally u^k is the $\mathcal{L}$-orthogonal projection of u^q onto $\mathcal{V}^k$.

5.3.5 Properties of the Construction

As mentioned earlier, the fine-to-coarse construction we derived above from basic desirable axioms is, in fact, equivalent to the work of [140], originally developed from a game-theoretical approach to optimal functional approximation. They argued that the optimal operator-adapted basis functions should satisfy a variational definition, formulated as

$$\varphi_i^k = \underset{\phi \in H}{\arg\min} \ ||\phi||_{\mathcal{L}}^2 \ \text{ s.t. } \langle \phi, \varphi_j^k \rangle_{L^2} = \delta_{ij} \text{ for } j = 1 \dots n_k. \tag{5.20}$$

This variational formulation implies that the basis functions φ_i^k can also be identified as optimal recovery splines in the sense of [128], and viewed as a generalization of energy minimizing splines in the sense of [186], variational multiscale (or LOD) basis functions in the sense of [106, 124] and polyharmonic splines in the sense of [74, 92, 142]. For completeness we provide a short summary of their arguments leading to this definition in App. C. From this variational definition, they derive a construction for operator-adapted basis functions and wavelets that mirrors ours, reinforcing the fact that our axiomatic characterizations are quite natural. As a consequence, we inherit the properties of the operator-adapted functions derived from Eq. (5.20). These properties include in particular:

- the condition numbers of stiffness matrices $\mathbb{A}^1$ and $\mathbb{B}^k$ are uniformly bounded for all $k = 1 \dots q-1$;
- the resulting basis functions φ_i^k decay exponentially fast under reasonably mild assumptions on the interactions between test functions and operator;
- operator-adapted basis functions allow for small approximation errors even if high frequencies (i.e., fine levels) are ignored, confirming their coarse-graining properties.

We provide a summary of the analysis leading to these conditions in App. C too, in order to make the underlying conditions and the actual properties more explicit, but interested readers should refer to the original paper [140] and to [139] for rigorous proofs. Note that we will numerically verify these properties in our context, and will explain the practical implications of their assumptions.

5.3.6 Practical Implementation

While the fine-to-coarse construction described in Sec. 5.3.4 is simple, it fails to be efficient to implement *as is* since many of the matrices involved are not sparse in general. Yet, the fact that operator-adapted basis functions are decaying exponentially fast can be leveraged to enforce sparsity throughout the construction. Additionally, we did not explicitly describe how the $\mathcal{L}$-adapted basis functions and wavelets are constructed in practice to recover the solution from Eq. (5.19), so we now go over implementation details of the fine-to-coarse construction procedure.

Basis Functions and Wavelets. Once the finest $\mathcal{L}$-adapted basis functions $\{\varphi_i^q\}_{i=1}^{n_q}$ have been defined (Eq. (5.16)), it is clear from Eq. (5.8) and Eq. (5.10) that all other basis functions and wavelets are simply *linear combinations* of these fine functions. Therefore, we can store them as just the coefficients of the fine basis functions they are made of. To simplify the notation, we assemble a (sparse) vector Φ^k of operator-adapted basis functions (with n_k rows) and a (sparse) vector Ψ^k of pre-wavelets (with N_k rows) for each resolution level k to keep track of these linear combinations:

$$\Phi^k = \begin{bmatrix} \varphi_1^k & \varphi_2^k & \dots & \varphi_{n_k}^k \end{bmatrix}^T, \qquad \Psi^k = \begin{bmatrix} \psi_1^k & \psi_2^k & \dots & \psi_{N_k}^k \end{bmatrix}^T,$$

where each φ_i^k and ψ_j^k is discretized as coefficients associated to all the n_q basis functions of the fine mesh. During the fine-to-coarse construction, these vectors are trivially assembled starting from the finest level basis $\{\varphi_i^q\}_{i=1}^{n_q}$ and onto coarser resolutions via the refinement relations of Eqs. (5.8) and (5.10). The pseudocode for computing adapted basis functions and corresponding wavelets is given in Alg. 5.1.

Algorithm 5.1 Operator-adapted Basis and Wavelet Construction

Input: Basis matrix $\mathbb{\Phi}^q$ and associated stiffness matrix $\mathbb{A}^q$ on finest resolution level, refinement matrices $\{\mathbf{C}^k, \mathbf{W}^k\}_{k=1}^{q-1}$.

for $k = q$ to 2 **do**

$\mathbb{\Phi}^{k-1}, \mathbb{\Psi}^{k-1}, \mathbb{A}^{k-1}, \mathbb{B}^{k-1} \leftarrow$ Per-level Coarsening Step $(\mathbb{\Phi}^k, \mathbb{A}^k, \mathbf{C}^{k-1}, \mathbf{W}^{k-1})$

end for

Output: Operator-adapted bases $\{\mathbb{\Phi}^k\}_{k=1}^q$, pre-wavelets $\{\mathbb{\Psi}^k\}_{k=1}^{q-1}$, and their stiffness matrices $\{\mathbb{A}^k\}_{k=1}^q$ and $\{\mathbb{B}^k\}_{k=1}^{q-1}$.

Algorithm 5.2 Per-level Coarsening Step

Input: Basis matrix $\mathbb{\Phi}^k$ and its stiffness matrix $\mathbb{A}^k$ on k-th resolution level, refinement matrices $\mathbf{C}^{k-1}$, $\mathbf{W}^{k-1}$.

1: Compute wavelet stiffness: $\mathbb{B}^{k-1} \leftarrow \mathbf{W}^{k-1}\mathbb{A}^k\mathbf{W}^{k-1,T}$

2: Compute pseudoinverse $\mathbf{C}^{k-1,\dagger} \leftarrow (\mathbf{C}^{k-1}\mathbf{C}^{k-1,T})^{-1}\mathbf{C}^{k-1}$

3: Compute intermediate matrix $\mathbb{Z}$ through $\mathbb{B}^{k-1}\mathbb{Z}^T = -\mathbf{W}^{k-1}\mathbb{A}^k\mathbf{C}^{k-1,\dagger\,T}$ (fast solve through localization)

4: Compute adapted basis refinement matrix: $\mathbb{C}^{k-1} \leftarrow \mathbf{C}^{k-1,\dagger} + \mathbb{Z}\mathbf{W}^{k-1}$

5: Compute adapted wavelets on level $(k-1)$: $\mathbb{\Psi}^{k-1} \leftarrow \mathbf{W}^{k-1}\mathbb{\Phi}^k$

6: Compute adapted basis on level $(k-1)$: $\mathbb{\Phi}^{k-1} \leftarrow \mathbb{C}^{k-1}\mathbb{\Phi}^k$

7: Assemble stiffness matrix for lower resolution level: $\mathbb{A}^{k-1} \leftarrow \mathbb{C}^{k-1}\mathbb{A}^k\mathbb{C}^{k-1,T}$ (sparsified through truncation)

Output: Operator-adapted basis matrix $\mathbb{\Phi}^{k-1}$, pre-wavelets $\mathbb{\Psi}^{k-1}$, stiffness matrices $\mathbb{A}^{k-1}$, $\mathbb{B}^{k-1}$ and refinement $\mathbb{C}^{k-1}$.

Sparsification. As a consequence of the exponential decay of operator-adapted basis functions discussed in Sec. 5.3.5, we can reduce computational complexity without sacrificing numerical precision by working with sparse matrices throughout the fine-to-coarse construction. Thresholding near-zero values will have no significant numerical effects, and will allow for efficient sparse computations; we refer the reader to [139, 140] for a thorough discussion of accuracy and error propagation across scales induced by this procedure. In the coarsening algorithm from level k to level $k-1$, we sparsify both matrix $\mathbb{C}^{k-1}$ and matrix $\mathbb{A}^{k-1}$ through simple truncation.

Fast Evaluation of $\mathbb{C}^{k-1}$. Seemingly, the most tedious part of the coarsening procedure is the computation of the $\mathcal{L}$-adapted refinement matrix $\mathbb{C}^k$ from Eq. (5.15). Evaluation is done in two stages: one linear solve to derive an intermediate matrix $\mathbb{Z}$ (used in the proof of the Prop. 5.1), then the final expression based on $\mathbb{Z}$, $\mathbf{C}^{k-1,\dagger}$, $\mathbb{A}^k$ and $\mathbf{W}^{k-1}$; see Alg. 5.2. In [140], $\mathbb{Z}$ is evaluated efficiently through localization, and we adopt the same approach here: in order to efficiently solve the linear system

$\mathbb{B}^{k-1}\mathbb{Z}^T = -\mathbf{W}^{k-1}\mathbb{A}^k\mathbf{C}^{k-1,\dagger\,T}$, we localize both $\mathbb{B}^{k-1}$ and the right hand side by exploiting the sparsity of $\mathbf{W}^{k-1}$. We compute each i-th column $\mathbb{Z}_i$ of $\mathbb{Z}^T$ by solving a small linear system as follows. Since the column $\mathbb{Z}_i$ corresponds to the i-th element of the coarser mesh, consider all mesh elements that are less than three element away[1], inducing a small region R of the domain. We first assemble a set of indices S such that $\ell \in S$ iff the ℓ-th row of $\mathbf{W}^{k-1}$ has at least one non-zero coefficient on one of the finer mesh elements in R. We then assemble a matrix $\mathbb{B}$ which is a submatrix of the stiffness $\mathbb{B}^{k-1}$ using only the rows and columns whose indices are in S (this is a reduced stiffness matrix of the wavelets around i whose support intersects R). We also assemble a vector $\mathbf{z}_i$ by subsampling the i-th column of $\mathbf{W}^{k-1}\mathbb{A}^k\mathbf{C}^{k-1,\dagger\,T}$ using the indices from S. Now entries of $\mathbb{Z}_i$ with indices from S, assembled in a vector $\mathbb{z}_i$, are evaluated by solving the small linear system: $\mathbb{B}\ \mathbb{z}_i = -\mathbf{z}_i$, while all other entries of $\mathbb{Z}_i$ are set to zero. This procedure to compute $\mathbb{Z}$ is significantly faster than a full-blown SparseLU or Preconditioned Conjugate Gradient solve, as it only involves linear solves of small size, and is less prone to inaccuracies than a hard truncation-based sparsification of the various matrices involved. In addition, it promotes sparsity of $\mathbb{Z}$ and of all subsequent operations, e.g., for the refinement matrix $\mathbb{C}^{k-1}$ and the basis functions stored as $\mathbb{\Phi}^{k-1}$.

Complexity. Exploiting sparsity makes the construction extremely efficient: in dimension d, it has computational complexity of $O(n_q \log^{2d+1} n_q)$ [139, 140] if one uses a fine grid with n_q degrees of freedom. Our timing tests confirm this expected complexity of our algorithm with fast evaluation of $\mathbb{C}^k$. However, we observed that our python implementation can produce slightly worse timings for $n_q < 10^3$ compared to regular sparse solvers combined with naive threshold-based sparsification: our fast evaluation only pays off for large enough problems.

5.3.7 Discussion

Pseudocode of the numerical procedure to compute an operator-adapted decomposition is given in Algs. 5.1 and 5.2. Notice that since the operator-adapted basis functions and wavelets are stored as linear combinations of the original test functions on the finest level $\mathcal{M}^q$, their evaluation at any point of the domain is fast as it only involves a few local evaluations of test functions. Consequently, all the regular finite-element approaches used to solve partial differential equations can be applied

[1] Note that this spatial extent can be adapted on a per-level basis to tailor the sparsity *vs.* accuracy balance of the decomposition.

Algorithm 5.3 Linear Solve

Input: Basis matrix $\mathbb{\Phi}^q$ and its stiffness matrix $\mathbb{A}^q$ on finest resolution level, refinement matrices $\{\mathbf{C}^k, \mathbf{W}^k\}_{k=1}^{q-1}$, right-hand side g^q in basis $\mathbb{\Phi}^q$.

for $k = q$ **to** 2 **do**

$\mathbb{\Phi}^{k-1}, \mathbb{\Psi}^{k-1}, \mathbb{A}^{k-1}, \mathbb{B}^{k-1}, \mathbb{C}^{k-1} \leftarrow$ Per-level Coarsening step $(\mathbb{\Phi}^k, \mathbb{A}^k, \mathbf{C}^{k-1}, \mathbf{W}^{k-1})$

Find wavelet coefficients w^{k-1} by solving $\mathbb{B}^{k-1} w^{k-1} = \mathbf{W}^{k-1} g^k$

Assemble level $(k-1)$ wavelet solution $\omega^{k-1} \leftarrow \mathbb{\Psi}^{k-1,T} w^{k-1}$

Coarsen right hand side $g^{k-1} \leftarrow \mathbb{C}^{k-1} g^k$

end for

Final coefficient solve $\mathbb{A}^1 v^1 = g^1$

Assemble final solution $u^q \leftarrow \mathbb{\Phi}^{1,T} v^1 + \omega^1 + \ldots + \omega^{q-1}$

Output: Solution to finite element problem u^q.

Algorithm 5.4 Nonlinear Solve

Input: Basis matrix $\mathbb{\Phi}^q$, refinement matrices $\{\mathbf{C}^k, \mathbf{W}^k\}_{k=1}^{q-1}$, stopping time *MaxIter* and basis recomputation rate m.

for $s = 0$ **to** *MaxIter* **do**

Compute stiffness $\mathbb{A}^q$ of linearized operator at current time step in basis $\mathbb{\Phi}^q$

Compute right hand side g^q at current time step in basis $\mathbb{\Phi}^q$

if $(s \bmod m == 0)$ **then**

Compute operator-adapted bases/wavelets $\mathbb{\Phi}^1$, $\{\mathbb{\Psi}^k\}_{k=1}^{q-1}$ and stiffness matrices $\mathbb{A}^1$, $\{\mathbb{B}^k\}_{k=1}^{q-1}$.

Optional: perform model reduction by selecting a subset of operator-adapted wavelets as active.

end if

Find solution $u^q(s)$ at current time step using most recent active operator-adapted wavelets.

$s \leftarrow s + 1$

end for

Output: Solution to finite element problem $u^q(s)$.

at any level of approximation—except that our "operator-aware" basis functions are used in lieu of the usual polynomial basis functions, just like in other works which derive tailored shape functions to improve accuracy on coarse grids [42]. The whole decomposition (or, for a fast approximation, part of the decomposition) can also be used to solve a linear differential equation like Eq. 5.1 by adapting the wavelet-Galerkin approach to our framework, as explicitly described in Alg. 5.3. Finally, nonlinear differential equations such as $\mathcal{L}(u) = g$ can also be solved efficiently with our approach: by linearizing the nonlinear operator at the current estimate of u and using the associated adapted basis functions and wavelets, one can iteratively improve the approximation of the solution as described in Alg. 5.4.

While we implicitly assumed that our test functions are traditional node-based and scalar-valued functions, we next describe how all the expressions given thus far actually apply to a much larger class of finite elements: our decomposition applies to differential forms and linear self-adjoint positive-definite exterior operators.

5.4 Operator-adapted Wavelets for k-forms

Finite element formulations have most commonly used node-based basis functions to discretize both scalar functions and the coordinates of vector fields. A growing trend in recent years consists in employing mixed finite element methods, where basis functions are associated with arbitrary mesh elements (vertices, edges, faces, and cells for 3D grids). These methods have been instrumental in promoting the use of finite dimensional spaces of *differential forms* for coordinate-free computations with scalar and vector fields. We demonstrate in this section that our wavelet construction extends to the *multiresolution analysis of differential p-forms adapted to a given continuous, linear, bijective, and self-adjoint positive-definite exterior operator* (see three examples in Fig. 5.1). We also exhibit basis functions and wavelets of p-forms adapted to common operators.

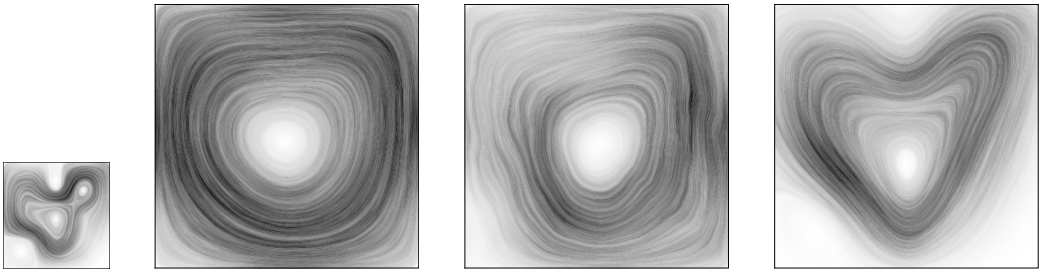


Figure 5.1: **Characteristic Solutions.** Elements of the solution spaces of the 1-form Laplacian (left), the modified 1-form Laplacian (middle, see Eq. (5.28)) and the advection-diffusion (right, see vector field of the advection in Fig. 5.6(top)) operators, computed by solving $\mathcal{L}u = g$ for the same right-hand side 1-form g (depicted far left).

5.4.1 Finite Element Differential Forms

In Cartan's "Exterior Calculus" [40], differential forms are used as the building blocks of differential and integral calculus in arbitrary dimension. For computational purposes, a number of efforts have been dedicated to derive a discrete version of this coordinate-free calculus through the use of finite dimensional approximation of differential forms.

Differential Forms for Computations. Mixed finite elements were introduced in the mid-1970s as an effort to build mixed discrete formulations for second order elliptic operators [133, 152]. The realization that they, in fact, were finite element spaces of differential forms came only later in [26]. At the root of these discrete differential forms is algebraic topology: cochains (resp., chains) provide a natural discretization of differential forms (resp., domain of integration) [132, 195]. Since a set of scalar values on vertices, edges, faces, and cells are proper discrete counterparts of respectively pointwise function evaluations, line integrals, surface integrals, and volume integrals [25], one can emulate exterior calculus on finite grids through the construction of finite-element differential forms [9]. This point of view naturally encompasses the use of volume integrals in finite volume methods and scalar functions in finite element methods, and also includes non-conforming "edge elements" and "facet elements" which are conforming in the spaces H_{div} and H_{curl} [133, 134]. Generalized basis functions that span the spaces of differential forms are particularly attractive, as many important scalar and vector calculus properties and invariants of the continuous setting directly carry over to the discrete world in a coordinate-free manner, culminating in a discrete Hodge theory [9, 11, 67]. The resulting finite-element framework for differential forms provides a powerful tool for solving a wide range of partial differential equations, which has been proven useful in a variety of applications such as computational electromagnetism [25, 176], fluid simulation [82, 102, 120, 129, 133], quadrangle meshing of curved manifolds [181], and computer vision [89] to mention a few.

Whitney Forms as Basis Functions of Differential Forms. The simplest basis of finite-element differential forms happen to correspond to a construct in algebraic topology known as Whitney forms [26]. Their initial purpose was to relate the de Rham sequence of smooth differential forms to simplicial cochain sequences in order to prove de Rham's theorem, which states that these sequences have isomorphic cohomology groups [194, 196]. But in a computational setting, they can be seen as low-order "basis functions" that turn values on mesh elements into differential forms: given a vector $U = (u_1, ..., u_n)^T$ of real values on all the n p-dimensional elements $\{\sigma_i\}_{i=1}^n$ of a mesh, one can construct a p-form $\mathbf{u}$ as a linear combination of the Whitney forms $(\phi_1^{[p]}, ..., \phi_n^{[p]})$ associated with these mesh elements through

$$\forall x \in \Omega, \quad \mathbf{u}(x) = \sum_{i=1}^n u_i \, \phi_i^{[p]}(x) = \mathbf{\Phi}^{[p]} U, \tag{5.21}$$

where $\mathbf{\Phi}^{[p]}$ stores all the Whitney p-forms as a row vector. In other words, there is one Whitney p-form for each p-dimensional element of a given mesh $\mathcal{M}$, and

their span generates a finite-dimensional space of differential p-forms. A Whitney p-form $\phi_i^{[p]}$ is expressed in terms of the barycentric coordinates of the associated p-element σ_i, and its integration over this (oriented) p-element is

$$\int_{\sigma_j} \phi_i^{[p]} = \delta_{ij}, \tag{5.22}$$

making it "histopolating"—an extension of the notion of pointwise interpolation; see, e.g., [153]. When a metric is introduced on the ambient space, Whitney forms can be expressed as either scalar-valued (for $p = 0$ and 3 in $\mathbb{R}^3$) or, via musical isomorphisms, vector-valued (for $p = 1$ and 2 in $\mathbb{R}^3$) functions with local support.

From Low-order to High-order Whitney Forms. While Whitney forms provide low-order interpolation of cochains (in particular, 0-forms are piecewise linear on a triangulation and trilinear on a cubical grid), a number of higher-order variants of these bases of differential forms have since then been proposed over simplicial or polytopal meshes [11, 35, 46, 84, 99, 100, 151, 191], some even with spectral accuracy [83, 157]. Their constructions vary greatly: a few are derived from tensor products of Whitney forms, while others follow the traditional approach of using higher "moments" to define various degrees of freedom associated to mesh elements in order to raise the order of the bases of differential forms while enforcing proper continuity across elements. Such high-order basis functions are carefully designed to span finite-dimensional differential p-form spaces Λ_p that form a *de Rham complex*,

$$\Lambda_0(\mathcal{M}) \xrightarrow[(\nabla)]{\mathbf{d}} \Lambda_1(\mathcal{M}) \xrightarrow[(\nabla\times)]{\mathbf{d}} \Lambda_2(\mathcal{M}) \xrightarrow[(\nabla\cdot)]{\mathbf{d}} \Lambda_3(\mathcal{M}) \to \mathbf{0}, \tag{5.23}$$

in which the fundamental differential operators gradient, curl, and divergence are unified as the exterior derivative $\mathbf{d}$ (which satisfies $\mathbf{d}\circ\mathbf{d} = \mathbf{0}$) [1], to offer structure-preserving finite element computations.

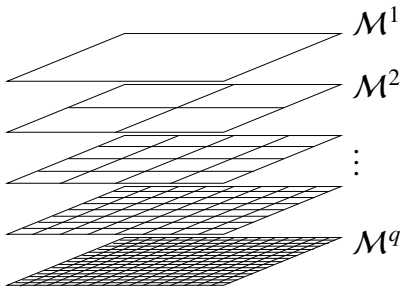
5.4.2 Refinable Test Functions for Differential Forms

The notion of refinability of basis functions [90] is crucial to the generation of wavelets, and even our multiresolution analysis requires an existing set of refinable test functions to bootstrap the bottom-up construction of our operator-adapted wavelets. Alas, refinability is a property rarely sought after or even discussed in the context of basis functions of differential p-forms: while the original Whitney forms are in fact refinable as pointed out in [191], most higher-order extensions are not known to have this property. As a consequence, the construction of wavelets for differential forms remains a challenge despite obvious applications for efficient

numerical computations in electromagnetism and other computational fields. To the authors' knowledge, the only arbitrary-order Whitney bases that are refinable by construction on both simplicial and polytopal meshes are the subdivision-based forms defined in [191] and extended in [62]. We thus review their construction, usage, and properties, as this particular family of refinable basis functions of differential forms and their associated mesh hierarchy will be used as test functions for differential forms in our multiresolution analysis.

Mesh Hierarchy

Due to the large body of known sudivision schemes for simplicial or polytopal meshes of arbitrary manifold domains [207], any (nested or non-nested) hierarchy of meshes for which subdivision schemes exist could be used for our construction in theory. However, regular (Cartesian) grids are undeniably simpler due to their tensor product nature, and often more amenable to efficient implementation. Consequently, for simplicity of presentation, we will only consider Cartesian meshes of simply connected domains in our explanations.



We thus assume that the computational domain is a d-dimensional cube in $\mathbb{R}^d$ (for $d = 1, 2,$ or 3), over which a nested hierarchy of regular grids $\{\mathcal{M}^k\}_{k=1}^q$ is defined, with $\mathcal{M}^{k+1}$ being twice as fine as $\mathcal{M}^k$ such that each p-dimensional mesh element of $\mathcal{M}^k$ is the union of 2^p p-dimensional mesh element of $\mathcal{M}^{k+1}$ for $p = 1, 2,$ or 3 (see inset for a 2D example of such a hierarchy). Each mesh element is given an arbitrary orientation, so that the boundary of any element is a linear combination of its faces with coefficients ± 1 based on whether the relative orientations match [101]. Considering only a simple d-dimensional cube domain is not a very restrictive assumption: we will show that our construction can not only homogenize operators, but also the spatial domain over which they are applied, making this simpler case sufficient to deal with complex domains (see Figs. 5.9 & 5.11)—but for accurate boundary handling, meshes conforming to curved boundaries (e.g., [78]) are obviously preferable.

(Low-order) Whitney Forms ϕ as Test Functions

Whitney bases for p-forms on cubical grids have been known for decades [134]: Whitney 0-forms $\phi^{[0]}$ associated to vertex values are the standard trilinear functions

over each hexahedral element, Whitney 1-forms $\phi^{[1]}$ associated with integrals over edges span the H_{curl}-conforming space, while Whitney 2-forms $\phi^{[2]}$ associated with integrals over faces span the H_{div}-conforming space. Finally, Whitney 3-forms $\phi^{[3]}$ associated with integrals over cells are equal to the indicator function of their associated cell. As noted in [190], these low-order Whitney forms are in fact refinable. Therefore, the Whitney forms $\phi^{[p]}$ for $p = 0, \ldots, 3$ can be used as test functions at any level of the mesh hierarchy, with one Whitney test function $\phi_i^{[p]}$ for each i-th p-element: the finite-dimensional differential p-form space Λ_p plays the role of the functional space of test functions at any level.

Design of Higher-order Whitney Forms through Subdivision

From these low-order original Whitney forms, higher-order refinable Whitney forms can be constructed as well using the concept of "subdivision" [190]. A subdivision scheme for p-forms maps scalar coefficients on p-elements from a coarse mesh $\mathcal{M}^k$ to the p-elements of a refined mesh $\mathcal{M}^{k+1}$ through a *subdivision matrix* $\mathbf{R}^{[p],k}$, converging to a p-form in the limit of refinement. The original Whitney forms on cubical grids correspond to a particular choice of subdivision matrix $\mathbf{R}^{[p],k}$ between two consecutive mesh levels $\mathcal{M}^k$ and $\mathcal{M}^{k+1}$ [190], as illustrated in 2D in Fig. 5.3(middle). Since a given subdivision scheme can be turned into a subdivision scheme of higher regularity through convolution along grid directions [191], subdivision-based high-order Whitney forms can be derived from the original low-order Whitney forms, as long as the associated p- and $(p+1)$-form subdivision schemes are treated in pairs to enforce that their subdivision matrices commute with the discrete exterior derivatives—thus ensuring a proper de Rham complex for the resulting finite element spaces spanned by these higher-order Whitney forms. For instance, in 2D, one such regularity-raising convolution results in bicubic splines for 0-forms and biquadratic splines for 2-forms as described in [62] (see Fig. 5.3(right)); the more convolutions one uses, the smoother the associated Whitney forms are the larger their supports become. The higher-order Whitney p-form $\phi_i^{[p],k}$ associated with the i-th p-element σ_i^k at level k can thus be conceived as the limit of repeated subdivisions of an initial assignment of scalar values δ_{ij} on the p-elements σ_j^k, i.e.,

$$\phi_i^{[p],k} = \lim_{m \to \infty} \mathbf{R}^{[p],m} \, \mathbf{R}^{[p],m-1} \ldots \mathbf{R}^{[p],k+1} \, \mathbf{R}^{[p],k} \, \mathbf{e}_i^k,$$

where the vector $\mathbf{e}_i^k$ of real values on elements of the mesh $\mathcal{M}^k$ contains a single 1 corresponding to element σ_i^k and zero on all other p-elements.

Properties of Subdivision-based High-order Whitney Forms

The resulting higher-order Whitney forms have also a few key distinguishing features that we summarize here, as these properties coming from their subdivision nature will be heavily exploited later on.

Simple Degrees of Freedom. While finite element exterior calculus methods use local polynomial shape functions associated with various degrees of freedom per mesh element [9, 84], our framework uses only the p-dimensional cells of a mesh to discretize a p-form. Consequently, the subdivision-based de Rham complex is simply the complex of the original low-order Whitney forms, i.e., the dual of the usual chain complex (see Fig. 5.2).

Finite-element Spaces of Differential Forms. High-order Whitney forms $\phi_i^{[p],k}$ can still be used to construct differential p-forms from mesh values through Eq. (5.21); the associated finite element spaces of differential forms on the level-k mesh $\mathcal{M}^k$ are, for $p = 0, \ldots, 3$,

$$\begin{aligned} \Lambda_p^k &= \operatorname{span} \left\{ \phi_i^{[p],k} \right\}_{i=1}^{n_k} \\ &= \left\{ \sum_{i=1}^{n_k} u_i \ \phi_i^{[p],k}(x) \ \mid u_i \in \mathbb{R} \ \text{ is associated to } p\text{-element } \sigma_i^k \in \mathcal{M}^k \right\}; \end{aligned} \tag{5.24}$$

that is, they define "basis functions" for p-forms that use only degrees of freedom associated with p-dimensional elements, but are of higher regularity than Whitney's original p-form basis, and with a larger spatial support. They can be used as test functions in our operator-adapted wavelet construction, i.e., we can set $\mathcal{V}^k := \Lambda_p^k$ for p-forms.

Refinement Property. The built-in refinablity of high-order Whitney p-forms implies

$$\phi_i^{[p],k} = \sum_{j=1}^{n_{k+1}} \mathbf{R}_{ji}^{[p],k} \ \phi_j^{[p],k+1}, \tag{5.25}$$

that is, if all the Whitney forms of level k are stored in a row vector $\mathbf{\Phi}^{[p],k}$, one has $\mathbf{\Phi}^{[p],k} = \mathbf{\Phi}^{[p],k+1} \mathbf{R}^{[p],k}$.

Commuting with Exterior Derivative. If we denote by $\boldsymbol{\partial}_p^k$ $(p = 0, \ldots, 2)$, the matrix of signed incidence between p- and $(p+1)$-elements of mesh $\mathcal{M}^k$ implementing the continuous boundary operator ∂ described in Fig. 5.2, then its transpose

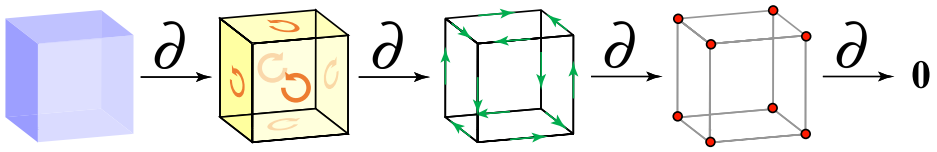


Figure 5.2: **Chain Complex.** Diagram depicting the chain complex of a cube element with the boundary operator ∂: from the cube, to its square oriented faces, to their oriented edges, and to their vertices.

$\mathbf{D}_p^k := \left(\partial_p^k\right)^T$ represents the discrete exterior derivatives acting on finite-element forms and implements the continuous derivative $\mathbf{d}$ [67], satisfying $\mathbf{D}_{p+1}^k \mathbf{D}_p^k = 0$ by construction. The subdivision matrices of high-order Whitney forms are chosen so that they commute with the discrete exterior derivatives [190]:

$$\mathbf{D}_p^{k+1} \mathbf{R}^{[p],k} = \mathbf{R}^{[p+1],k} \mathbf{D}_p^k.$$

That is, subdividing a discrete p-form on level k followed by application of the exterior derivative on level $k+1$ is the same as first applying the exterior derivative on level k followed by the subdivision scheme for $(p+1)$-forms.

Fast Pointwise and Integral Evaluations. Exact pointwise evaluation of the resulting high-order basis functions (which are piecewise polynomial for our regular meshes) at arbitrary parameter locations can be done efficiently [175]. Moreover, for each subdivision scheme based on a subdivision matrix $\mathbf{R}^{[p],k}$, there is an associated *limit stencil* operator $\mathbf{S}^{[p],k}$ defined through its matrix representation as

$$\mathbf{S}_{ij}^{[p],k} = \int_{\sigma_i^k} \phi_j^{[p],k}. \tag{5.26}$$

With this limit stencil, one can evaluate the integral of a finite-dimensional p-form on all the p-elements of a mesh: indeed, from a vector of real values U^k on the p-elements $\{\sigma_i^k\}_{i=1}^{n^k}$ of mesh $\mathcal{M}^k$, the p-form $\mathbf{u} \in \Lambda_p^k(\mathcal{M})$ defined through

$$\mathbf{u}(x) = \mathbf{\Phi}^{[p],k} U^k$$

integrated on a mesh element σ_ℓ^k is a local linear combination of the entries of U^k equal to the ℓ-th entry of the vector $\mathbf{S}^{[p],k} U^k$, that is,

$$\int_{\sigma_\ell^k} \mathbf{u} = \left[\mathbf{S}^{[p],k} U^k\right]_\ell.$$

Since the high-order Whitney forms are localized, $\mathbf{S}^{[p],k}$ is sparse; moreover, the non-zero values of this matrix can be evaluated in closed form through eigenanalysis of the subdivision matrix $\mathbf{R}^{[p],k}$ [62, 191], making the evaluation of integrals over

mesh elements of any finite-element differential forms particularly efficient. Note that the original Whitney forms satisfy $\mathbf{S}_{ij}^{[p],k} = \delta_{ij}$ (see Eq. (5.22)) due to their histopolation property; hence, the values of the vector U^k are directly the integrals of the continuous form. This histopolation property is no longer true for a higher-order Whitney form, but its limit stencil encodes precisely the map between degrees of freedom and local integrals of the corresponding finite-element p-form.

Discrete Stokes' Theorem. Due to the commutativity with discrete exterior derivatives and the existence of limit stencils, high-order Whitney forms induce a discrete variant of Stokes' theorem, $\int_{\mathcal{R}} \mathbf{du} = \int_{\partial\mathcal{R}} \mathbf{u}$ for any region $\mathcal{R}$ and p-form $\mathbf{u}$: for a vector of real values U^k on the p-elements $\{\sigma_i^k\}_i$ of mesh $\mathcal{M}^k$,

$$\mathbf{D}_p^k \, \mathbf{S}^{[p],k} U^k = \mathbf{S}^{[p+1],k} \, \mathbf{D}_p^k U^k. \tag{5.27}$$

In other words, Stokes' theorem holds over every mesh element in our finite-dimensional setting.

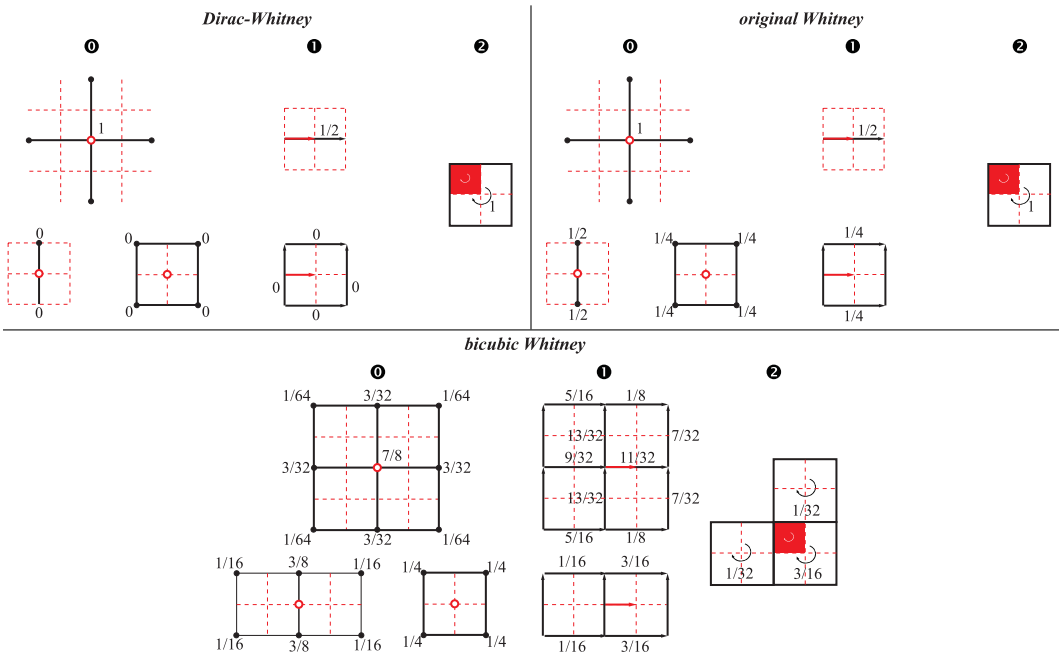


Figure 5.3: **Refinement Matrices for Subdivision-based Whitney Forms.** We provide the refinement rules in 2D and for 0-, 1- and 2-forms for (left) Dirac-Whitney, (right) original Whitney, and (bottom) "bicubic" Whitney forms. For a k-element on a twice-refined grid, the refinement rules use a simple linear combination of nearby coarse elements to define a refinable k-form basis.

In the remainder of our exposition, we will often omit the order $[p]$ of the Whiney forms or even the level k over which it is defined to improve legibility, when this does not lead to any ambiguity. Therefore, high-order Whitney forms will often be simply referred to as ϕ.

Finest Level Discretization through Subdivision-based Exterior Calculus

Consider a continuous, linear, and bijective *exterior* operator $\mathcal{L}$ that acts on differential p-forms and returns p-forms as well. In addition, we assume it is local, self-adjoint, and positive-definite with respect to L^2 inner product on differential forms (formally defined as $\langle \mathbf{u}, \mathbf{v} \rangle_{L^2} = \int_\Omega \mathbf{u} \wedge \star \mathbf{v}$). Note that the bijectivity condition is needed to ensure the inverse is well-defined: in practice a differential operator is combined with suitable boundary conditions to give rise to a well-defined problem. Given such an operator, the differential equation $\mathcal{L}u = g$ must first be discretized on the finest mesh $\mathcal{M}^q$. Many finite element based discretizations can be applied for this step. The computational foundations of Finite Element Exterior Calculus [9], Discrete Exterior Calculus [67], and Mimetic (or Compatible) Operators [23] differ in how they deal with the discretization of PDEs: for instance, some define a discrete operator corresponding to the Hodge star $\star$, while other prefer to bypass the creation of a discrete Hodge star by directly computing a discrete version of the codifferential $\boldsymbol{\delta} = (-1)^{n(k-1)+1} \star \mathbf{d} \star$ to enforce adjointness with the exterior derivative $\mathbf{d}$; boundary treatments are also often different as a consequence.

In our context, we remain agnostic vis-a-vis the discretization process of the weak form of the equation $\mathcal{L}u = g$. The only requirement is that our subdivision-based Whitney forms must be used as finite-element basis functions of differential forms, so that we can leverage their refinability with simple element-based degrees of freedom and structural properties (de Rham complex, Hodge decomposition, Stokes' theorem, etc.) to construct operator-adapted wavelets. For illustration purposes and numerical tests, we adopt the Subdivision-based Exterior Calculus framework (SEC [62]) based on the finite element spaces of differential forms $\{\Lambda_p^k\}_p$ defined in Eq. (5.24) to provide a discretization of exterior operators in this paper, but variants can be easily substituted.

Once this discretization is done on the finest level q, the rest of the operator-adapted multiresolution construction is mostly unchanged as we review next.

5.4.3 Operator-adapted Wavelets

While the subdivision-based Whitney forms naturally lead to a L^2-orthogonal notion of wavelets for any differential p-form by exploiting the difference between two consecutive approximation spaces $\mathcal{V}^k := \Lambda_p^k$ and $\mathcal{V}^{k+1} := \Lambda_p^{k+1}$ on two mesh levels $\mathcal{M}^k$ and $\mathcal{M}^{k+1}$, the induced multiresolution decomposition leads to the same shortcomings as for the wavelet-Galerkin approach—but this time in the context of finite-element differential forms. In this section, we discuss how to construct operator-adapted wavelets in order to block diagonalize a self-adjoint, positive-definite operator $\mathcal{L}$ acting on differential p-forms.

Bottom-up Construction of Wavelet p-forms. Because the construction provided in Sec. 5.3 made no assumption on the choice of mesh hierarchy or basis functions, it applies nearly *as is* for arbitrary p-forms. Because the notion of "refinement" and "subdivision" are in fact equivalent (inasmuch as they both define linear relationships between functions at two consecutive levels), one simply has to

1. set n_q to be the number of p-elements in the finest mesh $\mathcal{M}^q$;
2. bootstrap the construction by replacing the finest scale basis functions φ_i^q by Whitney p-forms ϕ_i^q (of any chosen smoothness order) on mesh $\mathcal{M}^q$, and compute the corresponding stiffness matrix $\mathbb{A}^q := \mathbf{A}^q$;
3. use refinement matrices $\mathbf{C}^k$ as the transpose of the subdivision matrices $\mathbf{R}^k$ associated with the selected Whitney p-forms, i.e., $\mathbf{C}^k := \mathbf{R}^{k,T}$, which in turn impose wavelet refinement matrices $\mathbf{W}^k$.

Everything else remains *unchanged* if the indices used in the procedure are understood to refer to an enumeration of the p-cells of each mesh; in particular, the number N_k of wavelet coefficients at level k remains equal to $n_{k+1} - n_k$ as it represents the dimension of the finite-element p-forms that are in $\mathcal{V}^{k+1}$ but not in $\mathcal{V}^k$. The resulting operator-adapted basis functions and wavelets will thus be linear combinations of Whitney p-forms on the finest grid, and can thus be stored as a sparse row vector per level as described in Sec. 5.3.6.

Choice of Refinement Matrix. As we just noted, a natural choice for a refinement matrix $\mathbf{C}^k$ to be used in our operator-adapted wavelet construction is (the transpose of) the subdivision matrix $\mathbf{R}^k$ associated to Whitney p-forms of a chosen regularity

order. This includes the subdivision matrices of the original Whitney p-forms (see Fig. 5.3(middle)), as well as any of the higher order Whitney subdivision matrices (see an example in Fig. 5.3(right)). Since the refinement matrix describes how a form basis element at level k can be written as a linear combination of the form basis elements at level k+1, the sparsity of $\mathbf{C}^k = \mathbf{R}^{k,T}$ decreases with the smoothness order of the corresponding Whitney forms, in turn impacting the sparsity and locality of $\mathbf{W}^k$.

Remark 3: Observe that the multiresolution construction could use a given subdivision-based Whitney p-form at the finest scale, whose regularity impacts the quality of the discretization $\mathbb{A}^q$ of operator $\mathcal{L}$, paired with an *arbitrary* refinement matrix $\mathbf{C}^k$. This choice of refinement instead impacts the regularity of test functions on coarser levels depending on the spatial size of the refinement stencil. As demonstrated in [139], approximation qualities of test functions in the weak H^{-s}-norm carry over to approximation qualities of the resulting operator-adapted wavelets in stronger energy norm. This means that using a simpler (i.e., sparser) refinement matrix $\mathbf{C}^k$ often turns out to be sufficient to obtain a high-quality operator adaptation, while offering a simplified computational procedure, faster decaying basis functions, and better condition numbers as we will demonstrate later. On the other hand, such simplified refinement rule obviously weakens the homogenization effect and thus reduces the efficiency of model reduction. Having $\mathbf{C}^k$ and $\mathbf{R}^k$ being decorrelated just offers added flexibility to our construction.

Dirac-Whitney Refinement Matrix. While the original (low-order) Whitney forms lead to the sparsest refinement matrices in the subdivision-based Whitney family, an even sparser choice of refinement matrix is worth mentioning: because our mesh hierarchy is nested, *the p-elements of mesh $\mathcal{M}^k$ can be written as linear combinations of p-elements of mesh $\mathcal{M}^{k+1}$*. This linear relationship can be seen as the refinement rule for the basis of *chains* or, equivalently, of *cochains*. One can thus see these refinement matrices as defining a dual basis to currents: they are measure-based Whitney forms that include, e.g., a Dirac delta function per vertex for 0-forms, and an indicator function of each 3-cell for 3-forms. For this reason, we refer to these sparsest refinement matrices as *Dirac-Whitney refinement matrices* (see Fig. 5.3(left)). While the corresponding (generalized) test functions are not quite useful in numerical applications (and would clearly be a poor choice of test functions), our construction can, in fact, use these refinement matrices: when paired with any subdivision-based Whitney forms on the finest level q to bootstrap

the construction as discussed in Rmk. 3 above, our resulting operator-adapted basis functions and wavelets will have very localized refinement matrices, but they will all be linear combinations of Whitney forms defined on mesh $\mathcal{M}^q$. As we will demonstrate in Sec. 5.5, this approach still leads to operator adaptation, with increased efficiency due to the sparsity of the Dirac-Whitney refinement matrices (see Fig. 5.4 vs. Fig. 5.5).

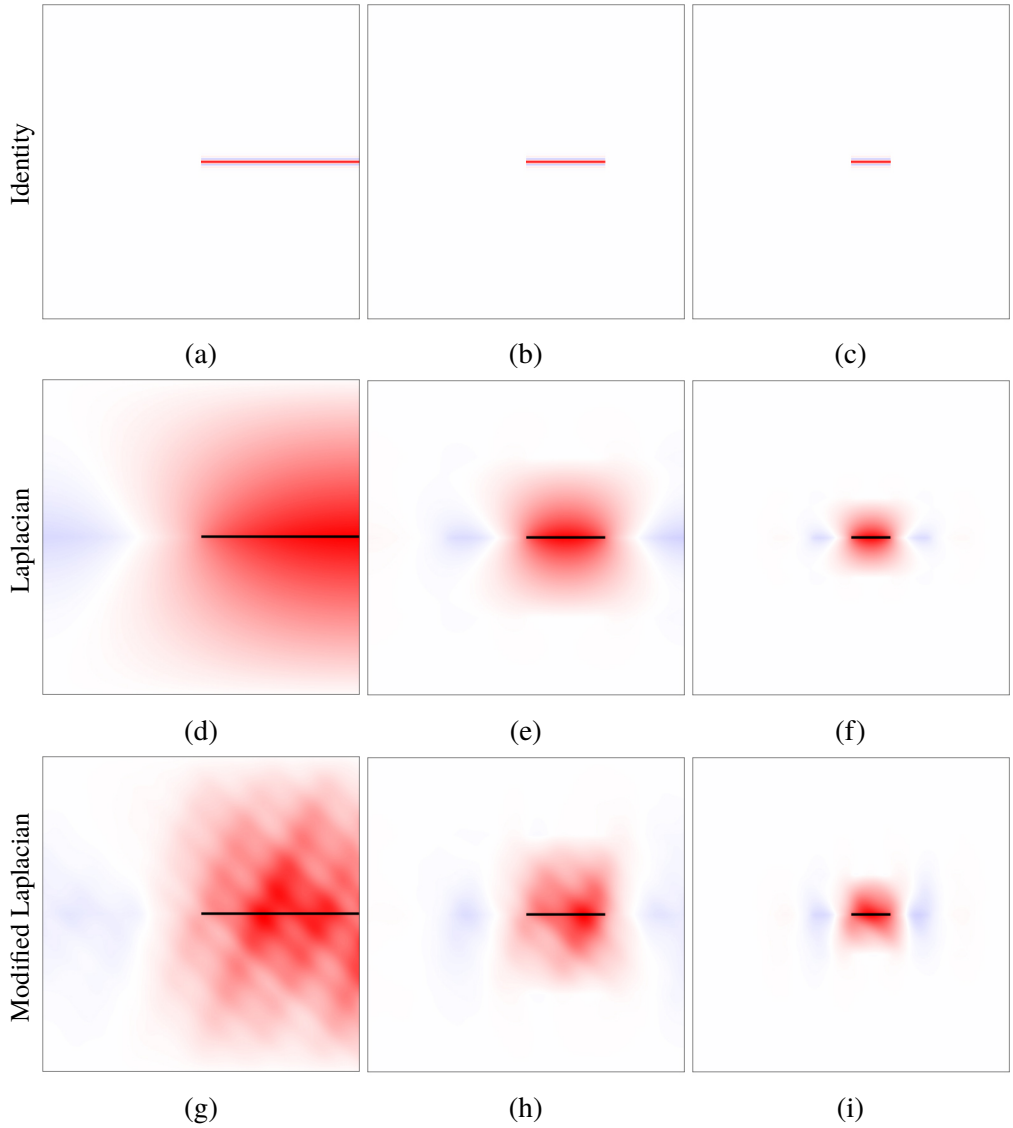


Figure 5.4: **Dirac-Whitney Basis Functions Adapted to Elliptic Operators.** Vertical components of 1-form basis functions φ_i^k adapted to different elliptic operators using the Dirac-Whitney refinement 1-form rule and associated with horizontal edges (in thick black) are visualized at three different resolution levels (2×2, 4×4, and 8×8 grids) using a linear red-to-blue colormap: (a)-(c) Identity operator; (d)-(f) 1-form Laplacian; (g)-(i) modified 1-form Laplacian (see Eq. (5.28)).

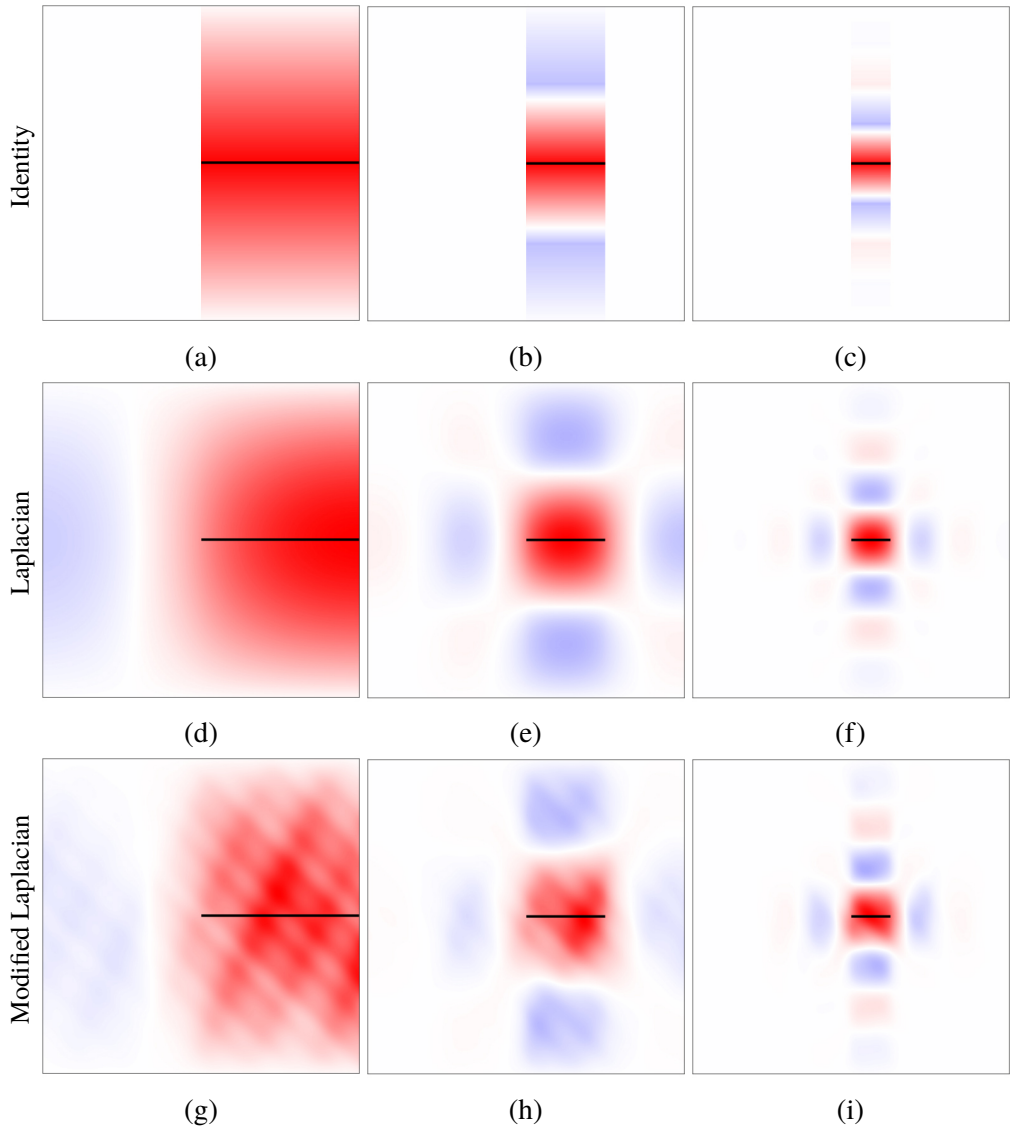
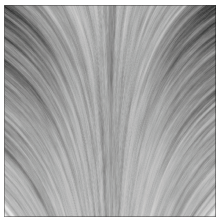
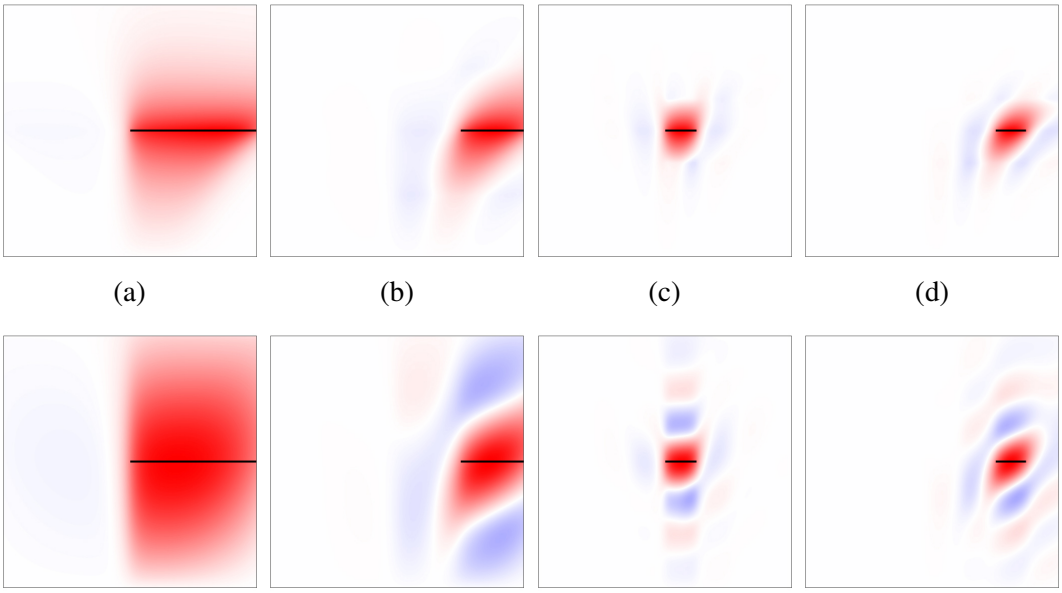


Figure 5.5: **Whitney Basis Functions Adapted to Elliptic Operators.** Vertical components of 1-form basis functions φ_i^k adapted to different elliptic operators using the original Whitney refinement 1-form rule and associated with horizontal edges (in thick black) are visualized at three different resolution levels (2×2, 4×4, and 8×8 grids) using a linear red-to-blue colormap: (a)-(c) Identity operator; (d)-(f) 1-form Laplacian; (g)-(i) modified 1-form Laplacian (see Eq. (5.28)).



Velocity field



(a) (b) (c) (d)

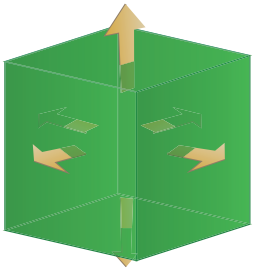
(e) (f) (g) (h)

Figure 5.6: **Advection-diffusion Adapted Basis Functions.** We visualize the vertical components of 1-form basis functions φ_i^k adapted to an (upwind-evaluated) advection-diffusion operator and associated to various horizontal edges (in thick black) from 3 resolution levels using a linear red-to-blue color ramp: using Dirac-Whitney refinement at (a) level $k = 1$, (b) level $k = 2$, (c)-(d) level $k = 3$; using the original Whitney refinement at (e) level $k = 1$, (f) level $k = 2$, (g)-(h) level $k = 3$. The advecting velocity field (left) is reflected in the shapes of the adapted edge basis functions.

5.4.4 Divergence-free Operator-adapted Wavelets

Oftentimes, differential equations of the form of Eq. (5.1) must be solved *within a given linear subspace*. This is the case, for instance, for incompressible fluids or solids where the additional incompressibility condition $\nabla \cdot u = 0$ plays a crucial role in the resulting dynamics. Enforcing this constraint has a long history in finite elements [47]. We show here that our operator-adapted wavelet construction can accommodate this case quite simply as well. We first focus on the case in which original Whitney forms are used, before extending it straightforwardly to higher-order Whitney forms.

Discrete Divergence. Stokes, Euler or Navier-Stokes equations in fluid dynamics are conveniently expressed with differential forms [82, 129, 134]. A typical discretization is to consider the velocity field as a 2-form in 3D, i.e., as a flux through each face of a 3D grid so that no-flux boundary conditions are trivially enforced [94]. A set of discrete fluxes $U^q = (u_1^q \dots u_{n_q}^q)^T$ on the finest resolution level $\mathcal{M}^q$ are converted into a continuous form $\mathbf{u}$ using the face-based low-order Whitney basis functions $\{\phi_i^q\}_{i=1}^{n_q}$, where n_q is the total number of 2-cells in $\mathcal{M}^q$. Note that the integral of the resulting continuous 2-form $\mathbf{u}$ on a face is precisely the flux initially stored on that face due to the histopolation property of linear Whitney forms (see Eq. (5.22)). The 2-form $\mathbf{u}$ (or, equivalently, the corresponding vector field $(\star\mathbf{u})^\sharp$ obtained by lowering the indices of the form once a metric is given) is divergence-free iff $\mathbf{du} = \mathbf{0}$, where $\mathbf{d}$ is the exterior derivative operator [1]. In our finite dimensional setting, it simply implies that the sum of the (oriented) fluxes leaving any 3-cell of the mesh (see inset) is zero, which is denoted as $\mathbf{D}_2^q U^q = \mathbf{0}$, where $\mathbf{D}_2^q$ is the discrete exterior derivative operator (the transpose of the boundary (incidence) operator acting on 3-cells; see [67] for instance) at level q.

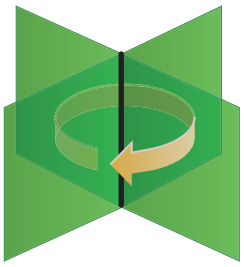


Divergence-free Test Forms Through Stream Functions. Instead of enforcing the linear divergence-free constraint on U^q, one can instead work in the reduced space of divergence-free vector fields. Because the whole setup of finite-element exterior calculus offers a discrete de Rham complex and since we assume our domain to be of trivial topology (no harmonic terms), we can consider instead a set of 2-forms, one per edge of $\mathcal{M}^q$, that spans this reduced space. This basis of divergence-free 2-form is defined as

$$\Xi^q = \mathbf{\Phi}^q \mathbf{D}_1^q,$$

where Ξ^q denotes the row vector of edge-based 2-forms $\{\xi_i^q\}_i$ and $\mathbf{\Phi}^q$ is, similarly, the row vector of all face-based Whitney 2-forms $\{\phi_i^q\}_i$. Hence, each 2-form ξ_i^q is a linear combination of local Whitney 2-forms with coefficients based on the discrete exterior derivative $\mathbf{D}_1^q$, a matrix whose rows and columns correspond to faces and edges of mesh $\mathcal{M}^q$ respectively.

For instance, for a regular 3D grid, each 2-form ξ_i^q associated with the i-th edge is simply the signed sum of the Whitney basis functions on the four faces adjacent to the edge (see inset). By construction, each edge-associated 2-form ξ_i^q



is divergence-free since $\mathbf{D}_1^q$ is in the kernel of $\mathbf{D}_2^q$ (i.e., $\mathbf{D}_2^q\mathbf{D}_1^q = \mathbf{0}$ corresponding to the continuous property $\mathbf{dd} = \mathbf{0}$). Hence, the finite dimensional vector space spanned by $\{\xi_i^q\}_i$ (they form a basis of this space due to their linear independence) represents the set of all 2-forms (or equivalently, vector fields) that are divergence-free in the discrete sense. This is nothing else but the differential form equivalent of defining the velocity through the curl of a *stream (vector-valued) function*: the degrees of freedom are thus edge-based values $\{s_i\}_i$ representing the local integrals of the stream function along edges, while $\sum_i s_i \xi_i$ is the discrete divergence-free 2-form.

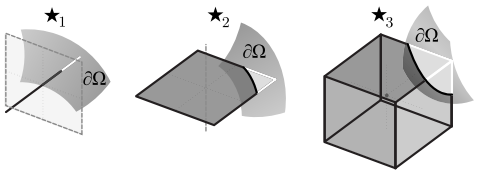
Operator-adapted Divergence-free Wavelets. The bottom-up approach described in Sec. 5.4.3 can now be applied directly with the divergence-free 2-form basis Ξ^q used as test functions at the finest scale to bootstrap the construction. The only difference is that n_q is now the number of *edges* (since it represents the number of degrees of freedom in this divergence-free case) and one must use a refinement matrix $\mathbf{C}^{[1],k}$ corresponding to 1-forms, even though we are constructing 2-form wavelets. With this approach, we work directly in the space of divergence-free fields without the need for reprojection steps: both operator-adapted bases and wavelets will be divergence-free by construction, since they are linear combinations of the elements of $\{\xi_i^q\}_{i=1}^{n_q}$. Note that in 2D, the stream function is a 0-form and fluxes through edges are 1-forms, so the construction must be altered to use linear combinations of edge-based basis functions per vertex instead, but the overall approach is identical otherwise.

Higher Order Bases. While the divergence-free construction we just described assumes that the refinement matrices corresponding to the original Whitney forms are used, the same construction applies to higher order Whitney forms as well. Indeed, when using subdivision-based Whitney bases, the integral of the reconstructed continuous 2-form $\mathbf{u}$ over an oriented face is always a local linear combination of the initial face values $\{u_i^q\}_i$, called the limit stencil $\mathbf{S}^{[2],q}$ (see Eq. (5.26)). Therefore, the vector of divergences over all 3-cells is given by $\mathbf{D}_2^q\,\mathbf{S}^{[2],q}\,U^q$, which by Eq. (5.27) (discrete Stokes' theorem), is also equal to $\mathbf{S}^{[3],q}\,\mathbf{D}_2^q\,U^q$. Enforcing $\mathbf{D}_2^q U^q = 0$ thus implies discrete divergence-freeness of the resulting 2-form, and the construction carries over as is, even in this high-order case.

Non-trivial Topology. If the domain has non-zero genus and requires a non-regular grid, the space of divergence-free forms/fields needs to account for harmonic forms. Thankfully, computing these additional basis functions is a well studied problem which can be achieved by leveraging the homology generators of the domain, see for instance [67]. As long as all the meshes $\mathcal{M}^k$ in the hierarchy have the same number of boundaries and the same genus (that is, they all capture the proper topology of the domain), these extra topological degrees of freedom are present at each level and are thus trivially added to the reduced bases of divergence-free forms at each level.

5.4.5 Embedding Complex Domains on Cartesian Grids

While unstructured meshes can conform to arbitrary domains, regular grids require simpler data structures and simpler refinement stencils which make them highly desirable when efficiency is paramount. However, handling complex domains clashes with the simplicity of regular grids.



Recently, in the context of DEC discretization of differential calculus, a local numerical homogenization of the diagonal Hodge star to capture sub-grid resolution was proposed in [120]. It extended to arbitrary k-forms the approach of [135], which offered a robust and second-order convergent pressure projection over an arbitrary domain Ω using a regular computational grid. While diagonal entries of the standard diagonal Hodge p-star are defined as the ratios of volumes of primal p-element and its dual element [26], the key idea of the improved boundary treatment is to account for the parts of the regular grid that are outside the domain Ω by altering the discrete diagonal Hodge star accordingly: partial lengths, areas, or volumes of primal elements that are within the domain Ω are used in the evaluation of the diagonal Hodge star, see inset. Topological operators like the exterior derivative remain unchanged. This method can be used in our context as well: one can construct arbitrary operator-adapted basis functions and wavelets on a regular grid while conforming to a smaller domain Ω by modifying the Hodge star operator on the finest mesh $\mathcal{M}^q$: as Figs. 5.9 and 5.11 demonstrate, an operator $\mathcal{L}$ involving Hodge stars in its exterior calculus expression will be properly handled over the actual domain Ω as long as it is covered by the computational grid. Note that this approach, which *homogenizes both space and operator* over the regular grid, is currently limited to a low-order approximations of the Hodge star. Future extensions to higher-order approximations of Hodge stars may provide more accurate sub-grid accuracy.

5.4.6 Discussion

A few remarks are in order about our extension to multiresolution analysis of differential forms.

L^2-adapted Case. When the identity operator is used along with Whitney p-forms at the finest level and their corresponding refinement matrices, one may expect that the resulting operator-adapted basis functions $\wp^k$ will simply be the same Whitney p-forms on each p-element of the mesh $\mathcal{M}^k$ at each level k. However, as Rmk. 2 in Sec. 5.3.1 pointed out, this would only be true if these Whitney forms were L^2-orthonormal in the first place, which is not the case. However, the induced L^2-orthonormal basis functions $\wp^k$ share a close resemblance to the Whitney forms from which the refinement matrix comes, as Fig. 5.5 demonstrates.

Impact of Refinement Matrix Choice. Having the finest basis functions used to bootstrap the construction and the actual refinement matrix $\mathbf{C}^k$ being decorrelated as noted in Rmk. 3 of Sec. 5.4.3 offers much flexibility in practical computations. Dirac-Whitney refinement matrices will have operator-adapted basis functions and wavelets with the smallest support, and this support will grow with the smoothness order of the Whitney forms that the refinement matrix corresponds to. Higher-order Whitney forms are advisable from a functional approximation point of view as they induce lower approximation errors for a given resolution level. If, instead, lowering computational cost is paramount, then low-order Whitney forms are preferable. In particular, Dirac-Whitney refinement matrices typically induce lower condition numbers of matrices $\mathbb{A}^k$ and $\mathbb{B}^k$: their associated refinement kernel matrix $\mathbf{W}^k$ have optimal condition numbers, i.e., $\operatorname{cond} \mathbf{W}^k = 1$, while the use of higher-order Whitney forms leads to higher (but still bounded) condition numbers.

Linear Differential Constraints. While we only gave details for the case of divergence-free vector fields, the construction of a basis of differential forms in a given linear subspace is possible for many typical differential constraints in physical systems: one only has to find a set of localized linear combinations of Whitney forms that creates a basis of the kernel of the linear operator at play in the constraint.

5.5 Numerical Tests and Applications

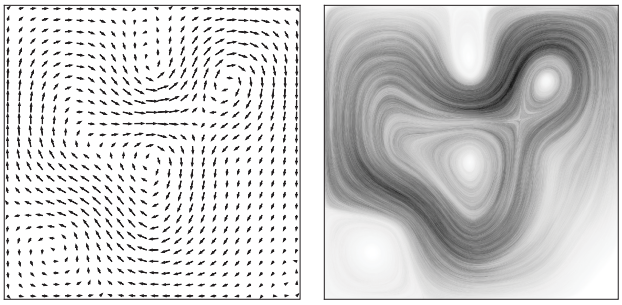


Figure 5.7: **Quiver plot vs. LIC visualization.**

In this section, we discuss potential applications of our operator-adapted decomposition of differential forms and present a number of numerical experiments to illustrate our contributions.

Since our work extends the recent work of Owhadi *et al.* [138–141, 158] on operator-adapted decompositions of scalar functions, we focus on examples involving one-forms (instead of zero-forms) to highlight the novelty of our work, even if our approach applies to arbitrary forms. We will depict one-forms either visualizing components of associated vector fields or using a variant of Line Integral Convolution (LIC [38], implemented in [156]), which offers a more expressive visualization than the traditional quiver plot of vectors (see Fig. 5.7) through the use of integral lines and greylevel coloring to encode direction and length respectively.

5.5.1 Elliptic Operators

We first show examples of operator-adapted basis functions and wavelets for typical linear operators. In these three examples, the original Whitney 1-forms are used at the finest level to bootstrap the decomposition.

Identity Operator. As a trivial example, we can run our algorithm on 1-forms for $\mathcal{L} = \mathbb{I}$. As expected, using a Dirac-Whitney refinement matrix leads to nearly edge-based Dirac basis functions, while using a low-order Whitney refinement matrix engenders an L^2-orthogonal basis of 1-forms resembling the edge-based Whitney forms; see Figs. 5.4 and 5.5 (top rows).

Laplace–de Rham Operator. For the 2D Laplacian $\mathcal{L} = \star_1 d_0 \star_0^{-1} d_0^T \star_1 + d_1^T \star_2 d_1$ of one-forms with zero Dirichlet boundary conditions (corresponding to the vector Laplacian $\nabla\nabla\cdot - \nabla\times\nabla\times$ in vector calculus), the adapted basis functions look much

smoother (see Figs. 5.4 and 5.5 (middle row)). This is to be expected due to the obvious link between the Dirichlet energy (a common measure of smoothness) and the Laplace–de Rham operator: as indicated by the variational definition in Eq. (5.20), they are, in fact, the *smoothest* localized functions.

General Elliptic Operator. Our approach applies to any elliptic operator, such as the general diffusion operator Δ_σ, where the vector Laplacian (with zero Dirichlet boundary condition) uses an arbitrary conductivity tensor σ. We demonstrate this property by using $\mathcal{L} = \tilde{\star}_1 d_0 \star_0^{-1} d_0^T \tilde{\star}_1 + d_1^T \star_2 d_1$ on one-forms, where the Hodge star of 1-forms is altered by making it dependent on a spatially-varying metric μ of the same form as proposed in [142]:

$$\begin{aligned} \mu(x,y) &= \begin{pmatrix} \alpha(x,y) & 0 \\ 0 & 1 \end{pmatrix}, \text{ where} \\ \alpha(x,y) &= \prod_{k=0}^{4} \left(1 + \tfrac{1}{5}\cos(2^k \pi(x+y))\right)\left(1 + \tfrac{1}{5}\sin(2^k \pi(x-2y))\right). \end{aligned} \tag{5.28}$$

We denote the resulting modified Hodge star as $\tilde{\star}_1$. Figs. 5.4 and 5.5 (bottom rows) show the resulting operator-adapted 1-form basis functions, reflecting the highly-oscillatory modification from the vector Laplacian that the metric μ creates. We will show in Sec. 5.5.5 that these basis functions allow proper homogenization of this elliptic operator and lead to bounded stiffness condition numbers. One can thus see our decomposition as an extension of the numerical methods for scalar-valued homogenization of the operator $\nabla \cdot \sigma \nabla$, such as [142].

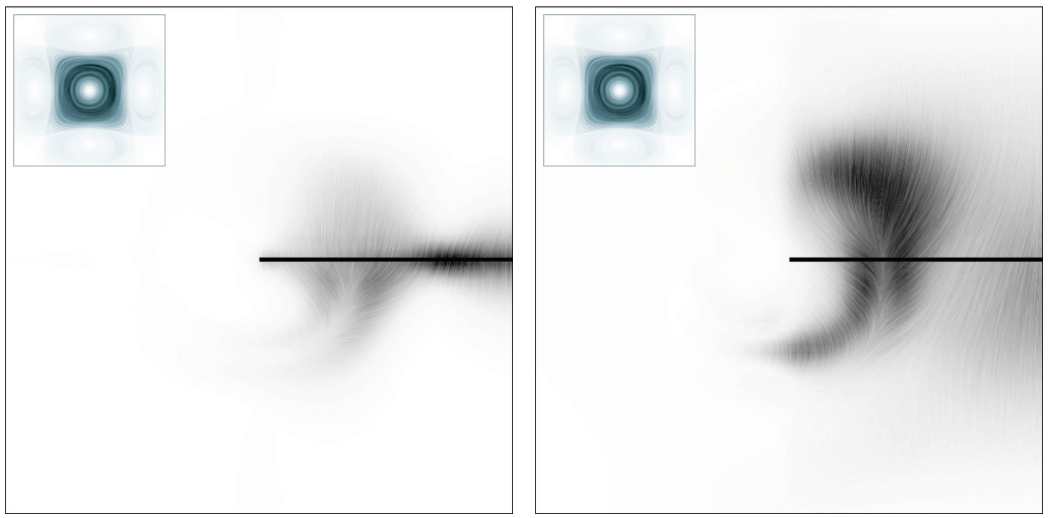
5.5.2 Advection-Diffusion Operator

Our approach can handle non-elliptic operators just as well: we demonstrate it by constructing a wavelet decomposition adapted to the well-studied advection-diffusion equation (with zero boundary flux and free-slip along the boundary):

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{a} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} = \mathbf{b},$$

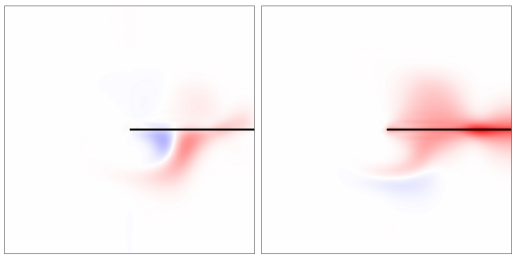
where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$ are given vector fields, $\nu \in \mathbb{R}_+$ is a viscosity parameter and $\mathbf{u} \in \mathbb{R}^2$ is the unknown, time-varying velocity field. In general, this equation describes the transport and diffusion of physical entities such as the ozone in the atmosphere or pollutants in oceans. Using its exterior calculus equivalent for one-forms (i.e., $u = \mathbf{u}^\flat$), a trapezoidal-based time discretization for a time step τ leads to the following implicit scheme (using $u_{n+\frac{1}{2}}$ as a shorthand for $\frac{1}{2}(u_{n+1} + u_n)$):

$$\frac{u_{n+1} - u_n}{\tau} + \pounds_{\mathbf{a}}\, u_{n+\frac{1}{2}} - \nu\, \Delta u_{n+\frac{1}{2}} = \mathbf{b}^\flat$$

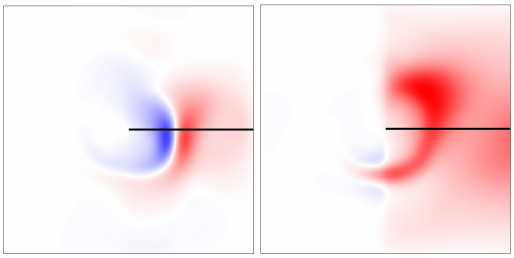


(a) Adapted basis (with Dirac)

(b) Adapted basis (with Whitney)



(c) x and y components



(d) x and y components

Figure 5.8: **Adapted Basis for Advection-diffusion.** We visualize a coarse-edge (in thick black) basis function φ_i^k adapted to advection-diffusion, where the advecting velocity field is shown in the top-left inset. Instead of the first-order upwind discretization used in other figures, an energy-preseving discretization of the advection operator [146] is employed here.

or, equivalently,

$$\left[\mathbb{I} + \tfrac{1}{2}\tau\ \pounds_{\mathbf{a}} - \tfrac{1}{2}\tau\nu\ \Delta\right] u_{n+1} = \left[\mathbb{I} - \tfrac{1}{2}\tau\ \pounds_{\mathbf{a}} + \tfrac{1}{2}\tau\nu\ \Delta\right] u_n + \mathbf{b}^\flat,$$

where the operator $\pounds_{\mathbf{a}}$ encodes the (Lie) advection of a form by the vector field $\mathbf{a}$. Observe that since the continuous form of the advection-diffusion equation mixes the skew-symmetric advection operator and the self-adjoint diffusion operator, the operator $\ell = \left[\mathbb{I} + \frac{1}{2}\tau\ \pounds_{\mathbf{a}} - \frac{1}{2}\tau\nu\ \Delta\right]$ is not necessarily symmetric and positive-definite. We thus compute the basis functions adapted to the symmetrized operator $\mathcal{L} = \ell^T\ell$ instead. In our results, we discretize the advection-diffusion operator ℓ using a first-order upwind approach [131] for most numerical tests involving advection-diffusion; the only exception is in Fig. 5.8 where we use the antisymmetric and conservative discretization provided in [146], which leads to similar results once symmetrized:

the adaptation to the operator $\mathcal{L}$ manifests itself in the "advected" shape of the resulting basis functions; see Figs. 5.6, 5.8, and 5.10.

5.5.3 Embedded Domains

By exploiting the domain adaptation of the Hodge star explained in Sec. 5.4.5, we can construct operator-adapted basis functions over complex domains while still using Cartesian computational grids. Figs. 5.9 and 5.11 show vector-Laplacian-adapted edge functions over fish-shaped and US-shaped domains. Far away from the boundary, the edge functions are virtually unchanged, but they nicely adapt to the convoluted boundary for surrounding edges. While more precisely conforming edge functions could be constructed from a triangle or quadrangle mesh approximation of the shape, our sub-grid accurate treatment of arbitrarily-shaped domains over regular grids is trivial to implement and properly converges as the grid is refined [120, 135].

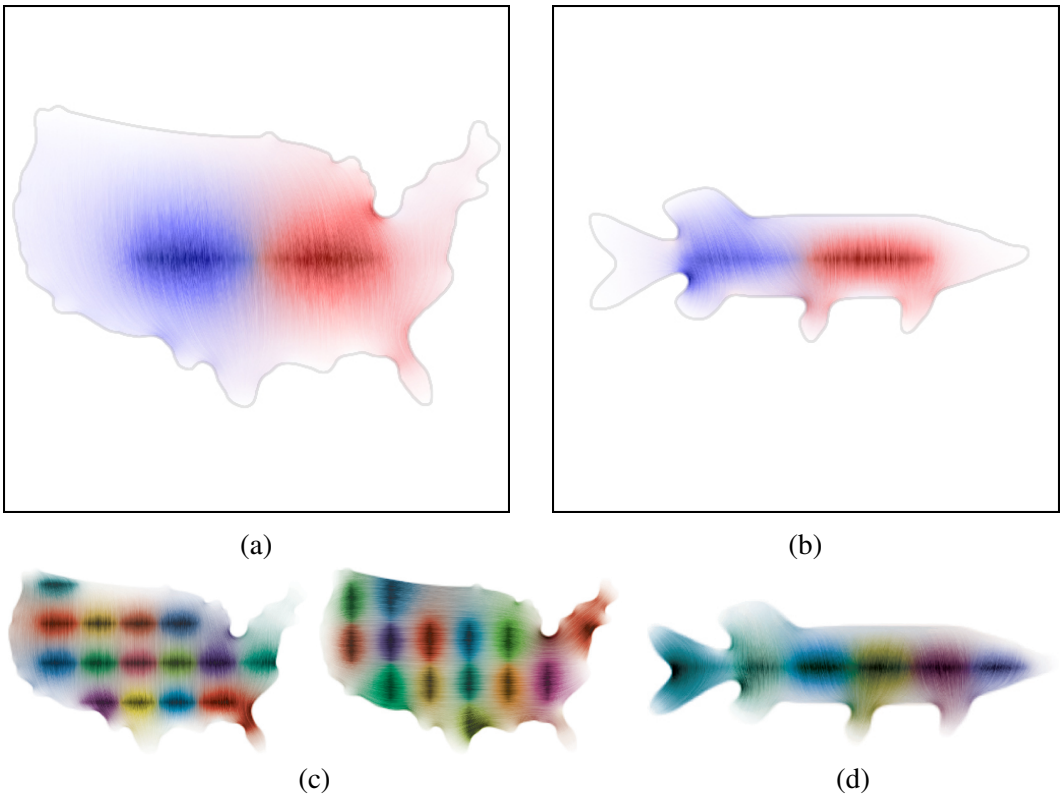


Figure 5.9: **Laplace-adapted basis functions on grid-embedded domains.** Discretization of the 1-form Laplace operator on a regular grid can be adjusted to account for an arbitrary computational domain, like a US map or a fish: using the finest grid of 256×256 elements, two resulting coarse (4×4 level) operator-adapted basis functions φ_i^k (one in red, one in blue) are displayed in (a-b), and all the basis functions at the level for which the grid is 8×8 are shown using different colors in (c-d).

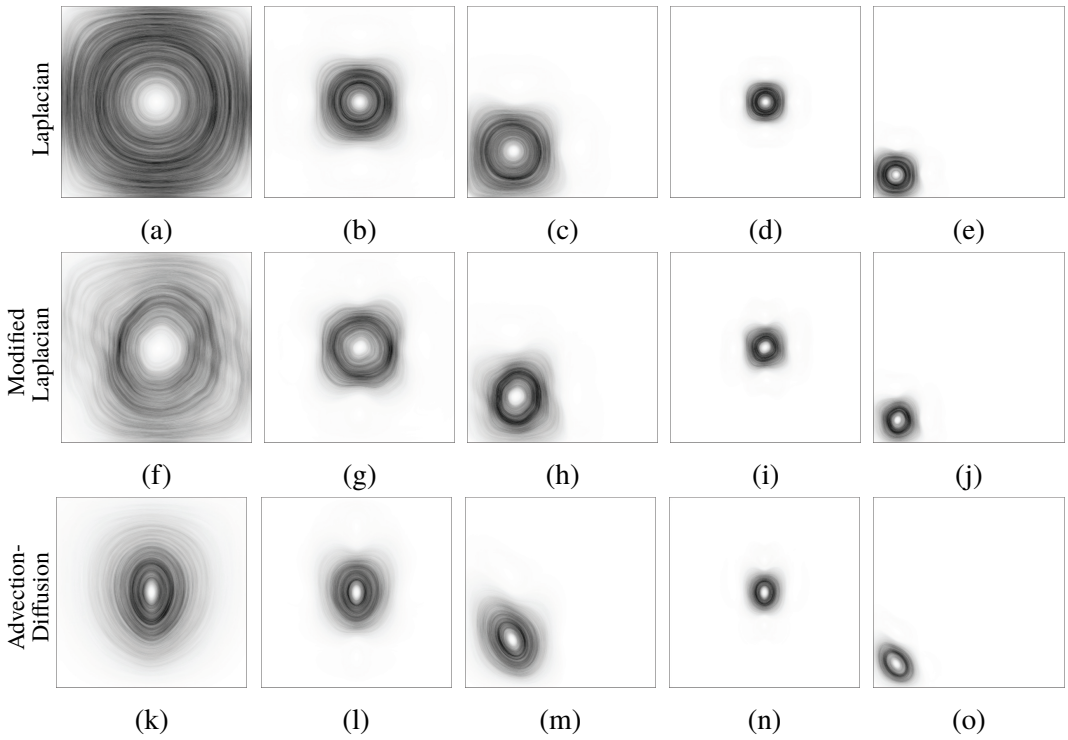


Figure 5.10: **Operator-adapted Divergence-free Bases.** We visualize vector fields corresponding to divergence-free 1-form basis functions φ_i^k adapted to various operators and associated to several spatial locations on three different resolution levels. Divergence-free basis functions retain a characteristic vortical appearance; however, the differences in their shapes reflect their ability to capture representative features of the associated solution spaces: they are smooth for the 1-form Laplacian (top), have high frequency oscillations for the modified 1-form Laplacian (middle, see Eq. (5.28)), and are stretched along the advecting velocity field for the advection-diffusion operator (bottom; see advecting field in Fig. 5.6(left)).

5.5.4 Divergence-Free Basis Functions

The construction of divergence-free vector-valued basis functions is an important computational need in many practical applications. While multiscale analysis approaches using divergence-free L^2-orthogonal wavelets have been proposed (in both the tensor and non-tensor product cases) [65, 183], none have been offered in the more general case of operator-orthogonality. As detailed in Sec. 5.4.4, our approach lends itself quite straightforwardly to this case. We demonstrate the divergence-free basis functions adapted to three differential operators in Fig. 5.10 on a simple domain. We also leverage the domain adaptation of the Hodge star in Fig. 5.11 to provide divergence-free bases on complex domains: using only a Cartesian grid, we construct a hierarchy of divergence-free vector-Laplacian-adapted basis functions that are restricted to a spatial domain with complex boundaries and with zero normal

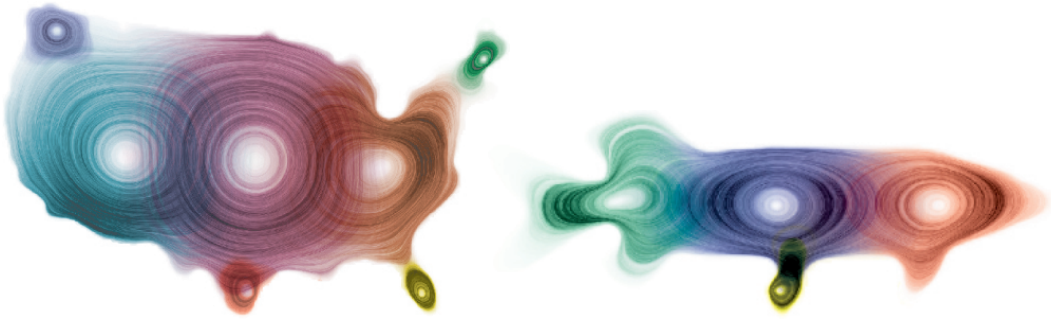


Figure 5.11: **Divergence-free Adapted Basis Functions on Grid-embedded Domains.** Using the same setup as Fig. 5.9, we now show the divergence-free basis function adapted to the 1-form Laplacian restricted to complex embedded domains. Various basis functions on a coarse (4×4) and a finer (8×8) level are shown using different colors in: three coarse and four dine basis functions for the US map, and three coarse and one fine basis functions for the fish.

components at the boundary. One can clearly see the effects of the boundary in the way the basis functions associated to edges of various hierarchy levels are shaped through the domain.

5.5.5 Approximation properties

We demonstrate the coarse-graining properties of our operator-adapted basis functions through the analysis of approximation errors. The notion of numerical homogenization is characterized by the following bound on the energy norm of the difference between a solution u to Eq. (5.1) and its operator-adapted level k approximation u^k (see Table 5.1 for definition):

$$||u - u^k||_{\mathcal{L}} \leq 2^{-ks} C \tag{5.29}$$

for the case of our 2D regular mesh hierarchy, where the constant $C > 0$ does not depend on the choice of operator (see App. D or [140] for the general case).

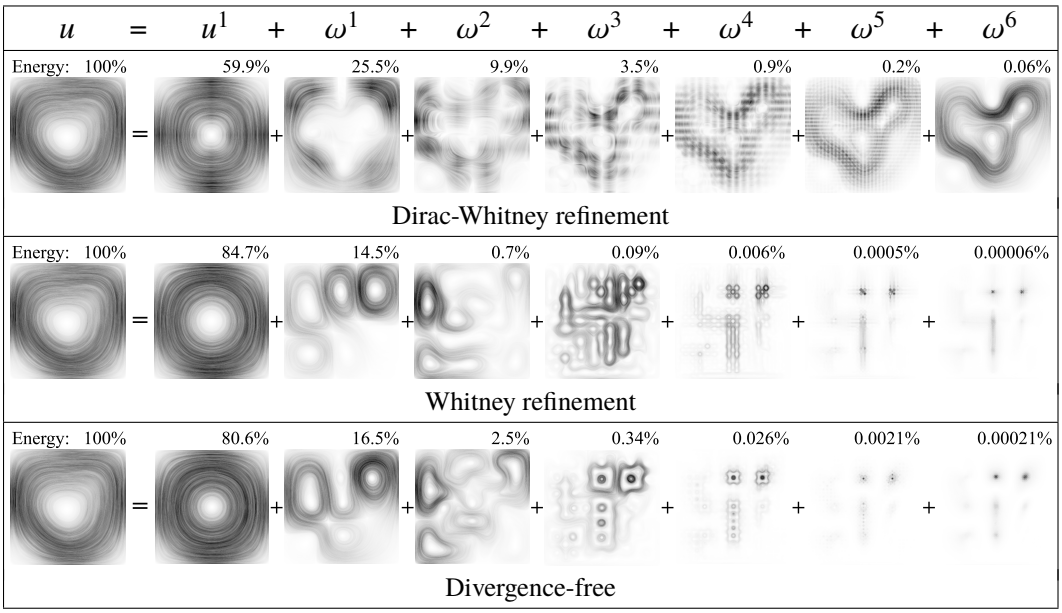
Using high resolution FEM solution of Eq. (5.1) as a proxy for u, and the right hand side of Eq. (5.29) as a reference slope, we plot the decay of the energy norm of approximation error as a function of k in Fig. 5.13 (top row) for several examples. The slopes of resulting curves follow the theoretical bound of Eq. (5.29) for three test operators, with original Whitney refinement rule leading to a faster error decay compared to Dirac-Whitney. If non-adapted wavelets are used instead (see Fig. 5.13 (bottom row)), the approximation errors deteriorate significantly and even display some growth with k for Dirac-Whitney refinement rule. For completeness, we also provide a similar plot in Fig. 5.14, where now the L^2-norm is employed to measure

approximation error instead of the energy norm. Operator-adapted construction still exhibits much faster decay compared to non-adapted version.

The coarse-graining properties of our basis-functions imply that only a subset of adapted wavelets can be used to produce a high-quality approximation of the solution space of their associated operator, thus naturally opening possibilities for efficient model reduction. The visualization of multiresolution decomposition of u^q for the 1-form Laplace and advection-diffusion operators in Fig. 5.12 also confirms that insight and reiterates that finer resolution levels correspond to high-frequency components of the solution, albeit they look quite different for the original Whitney vs. the Dirac-Whitney refinement rules: the former has better homogenizing properties due to larger support, which results in its ability to carry over more information to coarser resolutions. Indeed, Laplace-adapted multiresolution decomposition (Fig. 5.12(a)) shows that using only the 3 coarsest levels of Whitney wavelets leads to a smaller approximation error than using the 5 coarsest levels of the Dirac-Whitney construction; the Whitney-based construction allows to capture 99.9% of energy content with only 112 adapted wavelets, while Dirac-Whitney needs 1984 adapted wavelets to achieve the same energy content; for reference, a full solution u^q has 32512 degrees of freedom.

We finally note that our level k approximations u^k are in fact optimal both in Galerkin and game-theoretical sense; see App. C for more details.

(a) Multiresolution Decomposition for Laplace operator.



(b) Multiresolution Decomposition for Advection-diffusion operator.

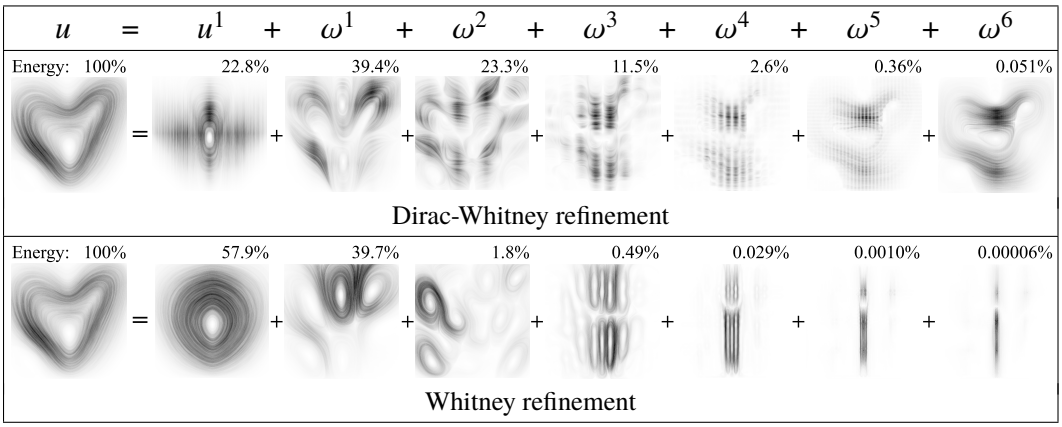


Figure 5.12: **Operator-adapted Multiresolution Decomposition.** Given a 1-form differential operator, an element of its solution space 1-form u defined via edge values on a 128×128 grid (left, LIC-visualized as its equivalent vector field) can be efficiently decomposed into a sum of a coarse 1-form u^1 described via edge values on a 2×2 grid, and all the wavelet contributions $\omega^1, \omega^2, \omega^3, \omega^4, \omega^5, \omega^6$ of the mesh hierarchy. Since our vector visualization does not convey relative magnitude, we also indicate the energy content (as a percentage) of each component. All operator-adapted bases exhibit homogenization properties with most of the energy concentrating on coarser resolution levels; however, this effect is significantly more pronounced for the Whitney refinement rule. We omitted operator-adapted divergence-free decomposition in (b), because advection-diffusion solution element u used in this figure is not divergence-free.

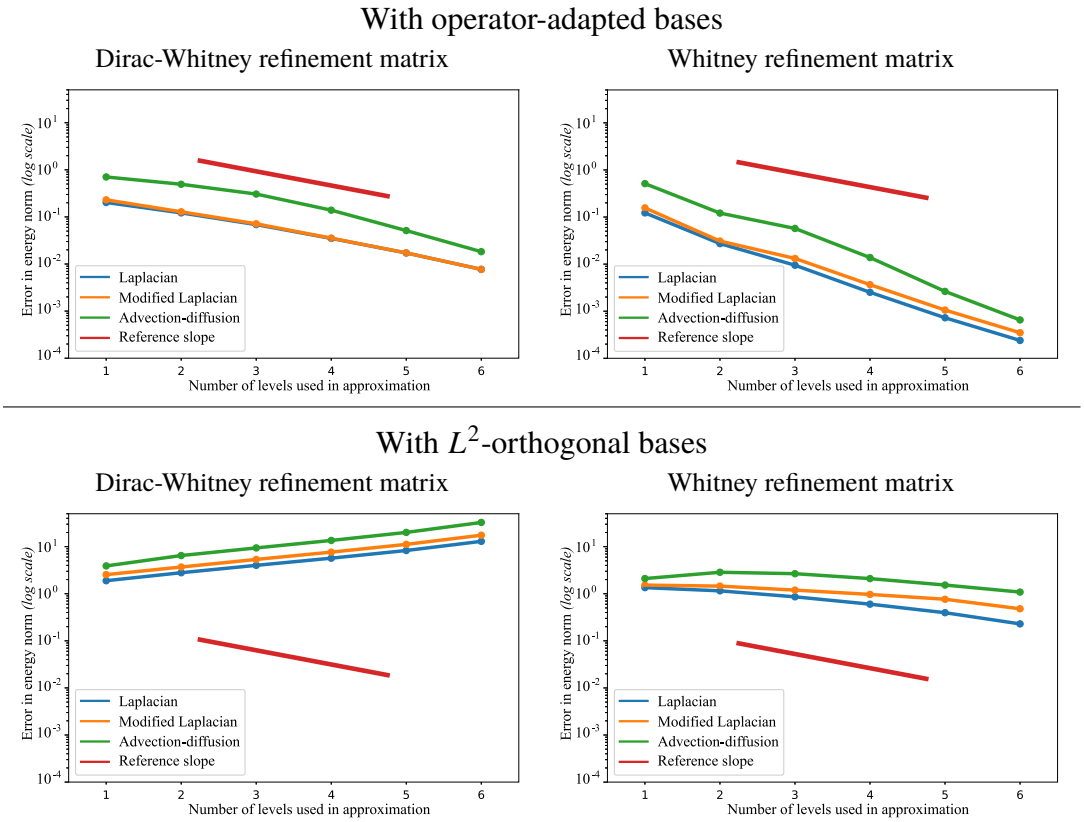


Figure 5.13: **Homogenization Effect.** We demonstrate decay of the homogenization error — i.e., the energy norm of the approximation error between the fine FEM solution u and its level-k operator-adapted approximation u^k (see Eq. (D.1)) — as a function of k; for each of the test operators (1-form Laplacian, modified 1-form Laplacian and advection-diffusion), the solution u is visualized in Fig. 5.1, to which the u^k approximant is compared. The slope of red line depicts the theoretical bound on typical numerical homogenization error behavior, assuming a regular 2D grid is used. Our operator-adapted Dirac-Whitney (top left) and original Whitney (top right) wavelets lead to exponential decay of the error for the three operators, demonstrating the expected numerical homogenization effect. In contrast, operator-independent wavelets based on Dirac-Whitney (bottom left) and original Whitney (bottom right) refinement rules fare significantly worse, leading to growing energy norm errors and unconvincing decay respectively (here we use wavelets that are L^2-orthogonal between the scales, i.e., adapted to Identity operator; see Figs. 5.4 & 5.5).

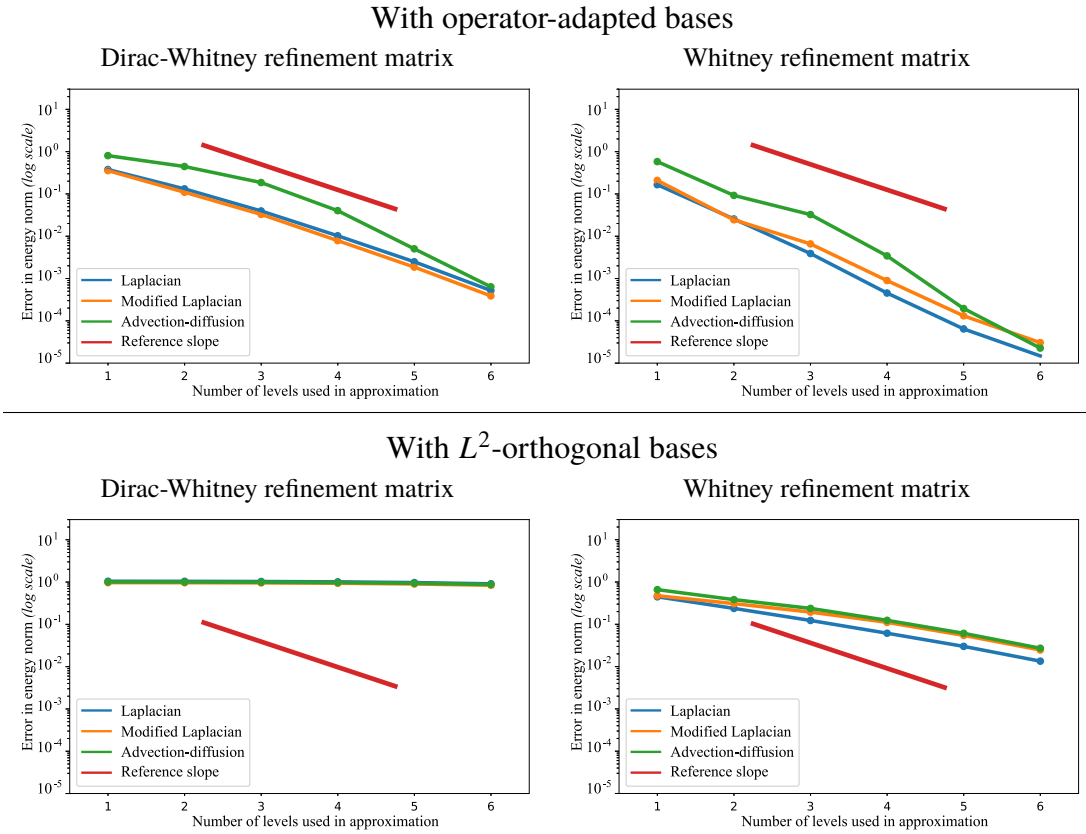


Figure 5.14: **Approximation Errors.** Using the same setup as Fig. 5.13, we now measure the approximation error using the L^2 norm instead of the energy norm. With this norm, it is known that non adapted bases may also exhibit convergence, but their rate of convergence can be arbitrarily bad [14]. While the decay rate is improved for all curves compared to Fig. 5.13, operator-adapted wavelets (top row) still perform significantly better than non-adapted ones (bottom row); Dirac refinement rules (left column) lead to slower decay than Whitney (right column), as expected due to smaller stencil. Note that the red curve representing the theoretical rate of convergence here is two times steeper than before.

5.5.6 Eigenvalue Ranges and Condition Numbers

An important measure of performance for our multiscale construction involves a spectral analysis of the resulting wavelet stiffness matrices $\mathbb{B}^k$. In Fig. 5.15, we show that the condition numbers of the stiffness matrices $\mathbb{B}^k$ are much smaller than their counterparts $\mathbb{A}^k$–from two to six orders of magnitude for high frequencies. Moreover, in Fig. 5.16 we also show that the ranges of eigenvalues of the matrices $\mathbb{B}^k$ overlap for consecutive values of k, reconstructing the full eigenvalue range of the input stiffness matrix $\mathbb{A}^q$ (evaluated with non-adapted basis on the finest resolution q), confirming the proper capture of the operator solution space at various spatial resolutions. Note that performing the same numerical tests for divergence-free bases lead to very similar results; see Fig. 5.17.

5.5.7 Exponential decay of basis functions

As we discussed in Sec. 5.3.5, our operator-adapted basis functions decay exponentially fast as long as some mild assumptions on the interaction between test functions and operator hold (see App. D). Figs. 5.18 and 5.19 corroborate this statement in the case of exterior operators, even if divergence-free constraints are imposed.

5.5.8 Stiffness Sparsity Patterns

We finally provide an evaluation of the sparsity of the stiffness matrices resulting from our multiresolution construction. In Fig. 5.20, we visualize diag$[\mathbb{A}^1, \mathbb{B}^1, ..., \mathbb{B}^6]$, a principal submatrix of the global multiresolution matrix $\mathbf{L}$ (Eq. (5.5)) using the Laplace operator on 1-forms. A straighforward implementation of Alg. 5.2 using an off-the-shelf linear solver combined with a pruning of all entries of $\mathbb{B}^k$ that are less than $1e^{-15}$ in absolute value results in already sparse matrices. However, our fast solve through spatial localization produces even sparser matrices with a limited impact on accuracy: the relative L^2 error in the resulting operator-adapted basis functions ψ_i^k introduced by this sparsification does not exceed 0.75% (as discussed in Sec. 5.3.6, we used 3-element-wide neighborhoods localization for Dirac-Whitney refinements, and 4-element wide for Whitney to account for larger support). Observe that in this example using 1-forms, each $\mathbb{B}^k$ is represented by two blocks corresponding to horizontal and vertical edges respectively.

Figure 5.15: **Condition Numbers.** We compare the condition numbers (in log scale) of stiffness matrices $\mathbb{A}^k$ of 1-form basis functions (in blue) and their associated wavelets $\mathbb{B}^k$ (in orange) for our operator-adapted multiresolution decomposition with $q = 7$ resolution levels (finest grid is 128×128 cells) for three operators: 1-form Laplacian (top row), modified 1-form Laplacian (middle row, see Eq. (5.28)), and advection-diffusion (right row). The Dirac-Whitney refinement rule (left column) tends to produce slightly better condition numbers for matrices $\mathbb{B}^k$ compared to the original Whitney rule (right column) as expected, but in both cases they remain bounded and often peak on intermediate resolution levels. In particular, condition numbers of the wavelet stiffness matrices $\mathbb{B}^k$ are up to 3 orders of magnitude smaller than those of $\mathbb{A}^k$ for large k (i.e., on finer scales).

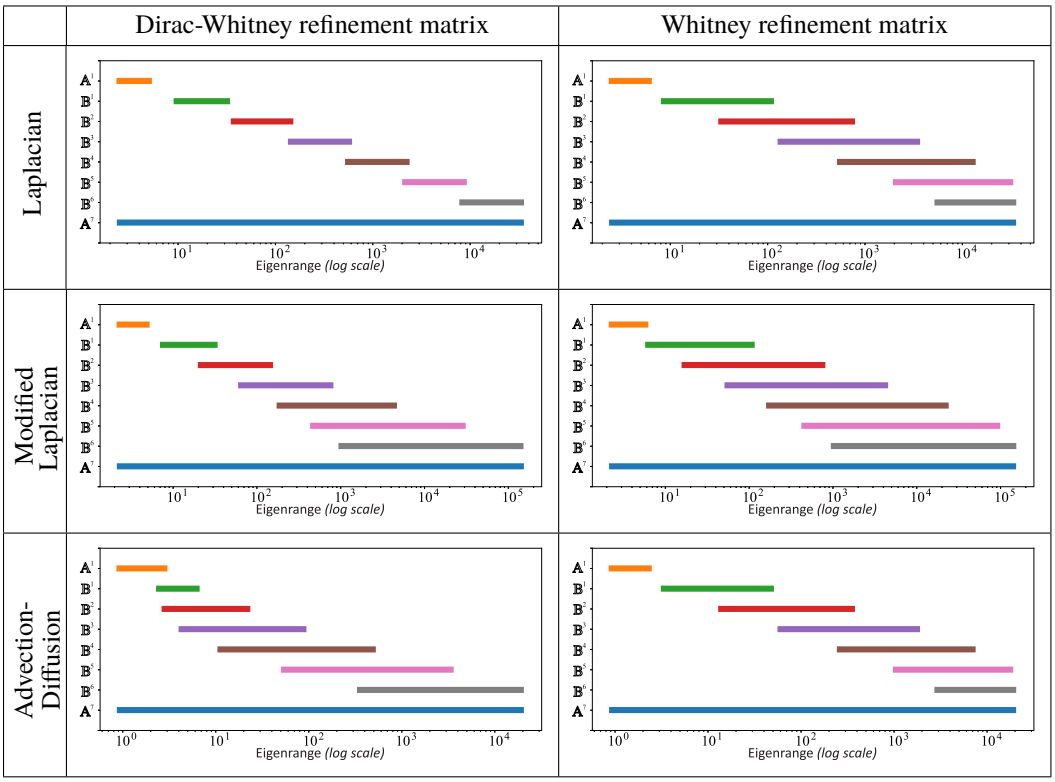


Figure 5.16: **Eigenranges.** We show that the eigenvalue ranges of 1-form wavelet stiffness matrices $\mathbb{B}^k$ are (overlapping) subbands of the eigenvalue range of the input stiffness matrix $\mathbb{A}^q = \mathbf{A}^q$ (evaluated with non-adapted basis on the finest resolution) for three test operators: 1-form Laplacian (top row), modified 1-form Laplacian (middle row, see Eq. (5.28)) and advection-diffusion (bottom row); each horizontal segment covers the spectrum of the associated stiffness matrix ($\mathbb{A}^1, \mathbb{B}^1, \mathbb{B}^2, \mathbb{B}^3, \mathbb{B}^4, \mathbb{B}^5, \mathbb{B}^6$, and $\mathbf{A}^7$.). Both Dirac-Whitney (left column) and Whitney (right column) refinement rules produce operator-adapted wavelets that capture frequencies associated with respective resolution levels; the degree of overlap generally depends on the operator and tends to be smaller for Dirac-Whitney refinement due to its small spatial support.

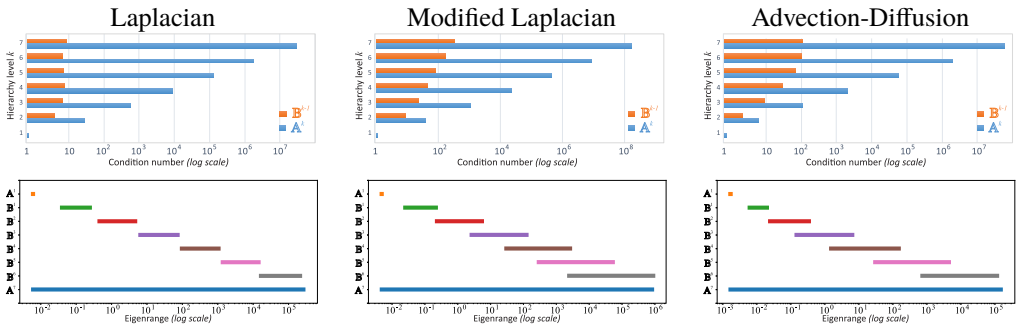


Figure 5.17: **Eigenanalysis of Operator-adapted Divergence-free Basis.** We demonstrate spectral properties of stiffness matrices corresponding to the 1-form divergence-free wavelets and basis functions adapted to 1-form Laplacian (left colum), modified 1-form Laplacian (middle column, see Eq. (5.28)) and advection-diffusion (right column): condition numbers (top row) and eigenrange subband structure (bottom row) have the same qualitative properties as for unconstrained operator-adapted multiresolution constructions (see Figs. 5.15&5.16). Note that the range of eigenvalues for the top level is degenerate (hence the short orange bar): since there's only one vertex at the coarsest level, the stiffness matrix becomes just a scalar.

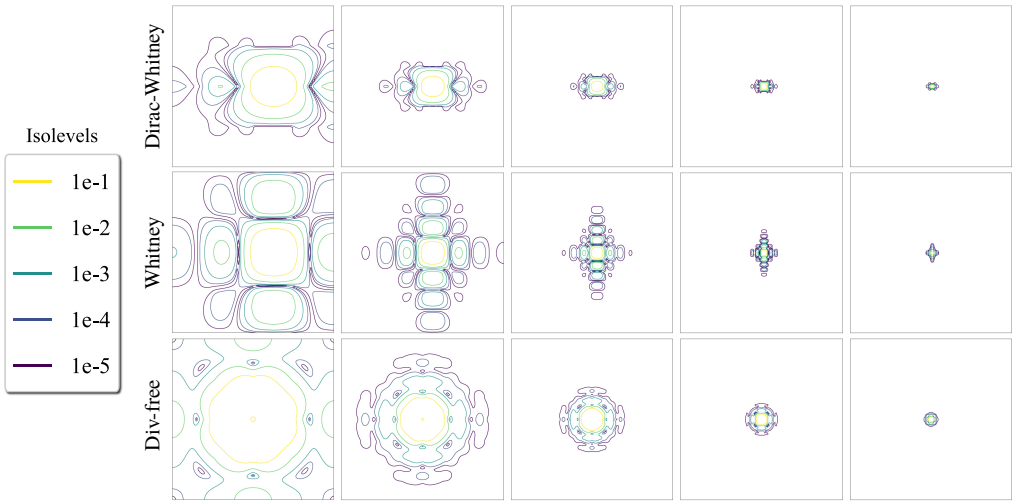


Figure 5.18: **Localization of Laplacian-adapted Bases.** We demonstrate the exponential decay of basis functions φ_i^k adapted to the 1-form Laplacian using Dirac-Whitney (top), original Whitney (middle), and divergence-free (bottom) refinements, normalized to have unit maximum value, using log-scale contour plots. Examples of edge-based basis functions from five different levels of the mesh hierarchy are shown.

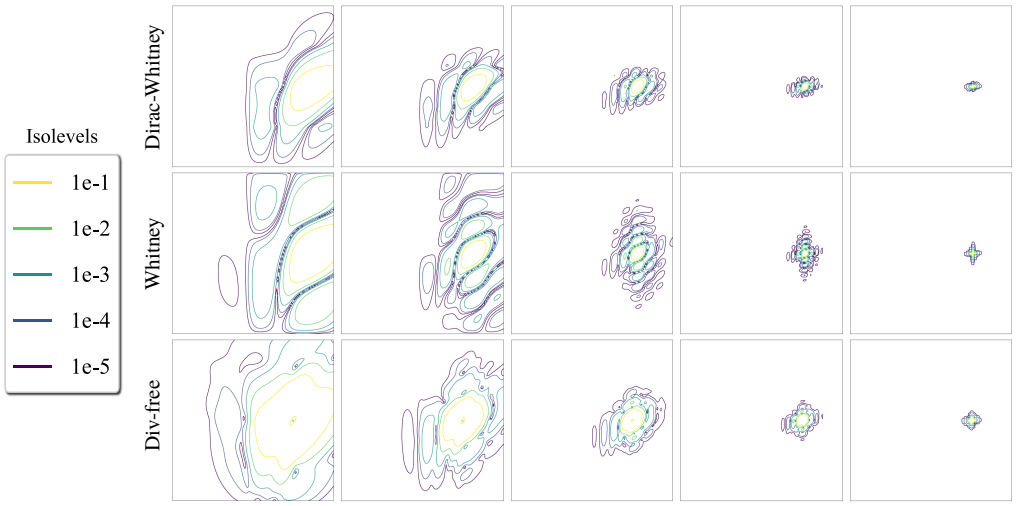


Figure 5.19: **Localization of Advection-diffusion-adapted Bases.** We demonstrate the exponential decay of basis functions φ_i^k adapted to the advection-diffusion (discretized through first-order upwind approximation) using Dirac-Whitney (top), original Whitney (middle), and divergence-free (bottom) refinements, normalized to have unit maximum value, using log-scale contour plots. Examples of edge-based basis functions from five different levels of the mesh hierarchy are shown.

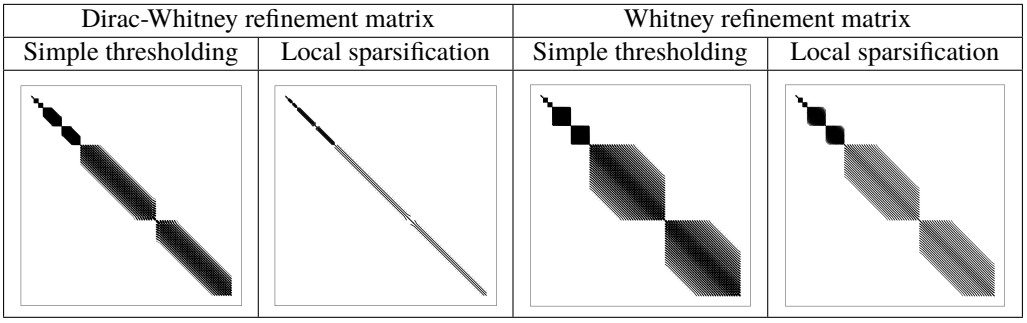


Figure 5.20: **Sparsity.** We visualize the sparsity pattern of the block diagonal stiffness matrix $\mathrm{diag}[\mathbb{A}^1, \mathbb{B}^1, ..., \mathbb{B}^6]$ obtained via Alg. 5.1 when a simple linear solver with threshold-based ($< 1e^{-15}$) pruning (left) vs. fast localized solve (right) is used, for a 1-form Laplace operator adapted decomposition using Dirac-Whitney (top) and original Whitney (bottom) refinement rules. Using fast localized solves allows to decrease sparsity (proportion of non-zeros) from 6.58% to 0.51% for Dirac-Whitney, and from 10.96% to 3.63% for original Whitney refinement.

5.5.9 Examples of Applications

Our construction of fine-to-coarse operator-adapted basis functions and wavelets have many potential applications in numerical contexts. We cite a few important examples next.

Model Reduction. As we discussed early on, our operator-adapted basis functions offer a finite-dimensional basis of localized functions that captures the eigensubspaces of the corresponding operator, associated with different frequency subbands, as well as possible. Therefore, they are particularly relevant in the context of model reduction and adaptive refinement, much more so than classical wavelets. We next discuss a few approaches to leverage our operator-adapted multiresolution analysis to either improve accuracy for the same number of degrees of freedom, or reduce the number of degrees of freedom for a given accuracy.

- *Resolution reduction.* The simplest way to reduce the number of degrees of freedom in the discretization of a differential equation with limited impact on accuracy is to omit high frequencies (i.e., the "details" of the solution corresponding to finest operator-adapted wavelets) while retaining wavelets spanning the p coarsest resolution levels for $p < q$. Since our bottom-up construction accumulates information from finer scales to derive coarser basis functions, this simple approach allows to capture most of the solution without a large amount of fine scale wavelets.

- *Goal-driven model reduction.* While the previous approach is intuitive and simple, it fails at providing a clear assessment of the error induced by the removal of the finest details. Goal-oriented adaptive refinement techniques aim, instead, at selecting a subset of wavelets from each resolution level leading to a computationally efficient and high quality estimation of a particular output functional [36], typically defined as a linear functional Q. This is easily achieved in our framework: after computing wavelet coefficients w^k on level k via a sparse linear solve, the least contributing wavelets $\{\psi_j^k\}_{j=1}^{N_k}$ are found by checking if
$$|Q(w_j^k \psi_j^k)| \leq \varepsilon \max_{\substack{1\leq i\leq N_k \\ 1\leq p\leq k}} |Q(w_i^p \psi_j^p)|$$
for a small $\varepsilon > 0$. These wavelets can be safely eliminated to reduce the number degrees of freedom and make the matrices $\mathbb{B}^k$ sparser while insuring that the quantity of interest Q can still be well approximated. Note that the error in Q

incurred by removing the weakly contributing wavelets can be evaluated on the fly.

- *Norm-driven reduction.* Yet another strategy to drop degrees of freedom is to remove those that least contribute to the operator-norm of the solution. Since our wavelets are well localized, interactions between them tend to be limited. Therefore, a wavelet ψ_j^k at level k can be considered negligible in its impact on the solution if

$$||w_j^k \psi_j^k||_{\mathcal{L}}^2 \leq \varepsilon \max_{\substack{1\leq i\leq N_k \\ 1\leq \ell\leq k}} ||w_i^\ell \psi_i^\ell||_{\mathcal{L}}^2$$

for a small $\varepsilon > 0$. However, since the operator norm of the solution

$$||u^q||_{\mathcal{L}}^2 = \sum_{i,j=1}^{n_1} u_i^1 u_j^1 \mathbb{A}_{ij}^1 + \sum_{k=1}^{q-1} \sum_{i,j=1}^{N_q} w_i^k \, w_j^k \, \mathbb{B}_{ij}^k$$

has non-negligible wavelet interaction terms associated with off-diagonal elements of the stiffness matrices, this simple heuristic can be made more robust by measuring the importance of a particular wavelet ψ_j^k through its full contribution to the norm, expressed as

$$\left| w_j^k \sum_{i=1}^{N_q} w_i^k \, \mathbb{B}_{ij}^k \right|.$$

Thresholding these contributions results in a well-approximated solution in the operator norm with typically very few coefficients.

Efficient Navier-Stokes Simulation. Our approach also applies to nonlinear equations such as the Navier-Stokes equations for incompressible fluids:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} = 0$$
$$\nabla \cdot \mathbf{u} = 0.$$

Multiresolution analysis of incompressible fluids is an important topic in computational fluid dynamics, but an efficient construction of wavelets that are truly adapted to the fluid dynamics has remained elusive. Our construction offers such a tool, and could be extremely useful to turbulence analysis and efficient simulation.

For simple integrators where the update rule is linear in the next velocity $\mathbf{u}^{n+1}$, our approach can be used *as is* where the operator corresponds to the update rule just like we showed in Sec. 5.5.2 in the case of advection-diffusion. For instance, using a

trapezoidal temporal discretization with operator splitting, we obtain the following Implicit-Explicit numerical scheme involving the usual pressure projection step:

$$\left[\mathbb{I} + \tfrac{1}{2}\tau\,\mathrm{Adv}(\mathbf{u}_n) - \frac{1}{2}\tau\nu\Delta\right]\mathbf{u}^*_{n+1} = \left[\mathbb{I} - \frac{1}{2}\tau\,\mathrm{Adv}(\mathbf{u}_n) + \frac{1}{2}\tau\nu\Delta\right]\mathbf{u}_n$$
$$\Delta p_{n+1} = \nabla\cdot\mathbf{u}^*_{n+1}$$
$$\mathbf{u}_{n+1} = \mathbf{u}^*_{n+1} - \nabla p_{n+1}.$$

If the velocity $\mathbf{u}$ is treated as a two-form (flux $\star\mathbf{u}^\flat$) and the pressure p as a three-form (integral per cell), the advection terms Adv can be discretized to produce the traditional Harlow-Welsh scheme [94] as discussed in [146]. Alas, this type of simplistic integration scheme is only relevant for low Reynolds numbers. For more involved nonlinear integrator schemes, our construction of operator-adapted basis functions can still pay off significantly: one can linearize the integrator, use only a reduced basis of divergence-free velocity fields (in order to bypass the pressure projection altogether), and construct operator-adapted basis functions at the current time step in almost linear time; from these basis functions, the nonlinear update rule can be solved efficiently by picking only the most relevant degrees of freedom as discussed earlier, which cuts down on the computational time of the simulation significantly (see Alg. 5.4). In addition, other simple heuristics can be applied; for instance, one could omit wavelets that are far enough from certain areas of interest in physical space. Finally, note that fluid simulations generally require high resolutions to accurately resolve interactions between small and large scale structures. They often benefit from subgrid models, aimed at modeling small scale effects without explicitly resolving them, thus reducing the dimensionality of the system. Our operator-adapted bases also provide a framework to directly modulate particular frequency subbands (with localization in physical space if needed) by appropriately varying the corresponding wavelet coefficients in order, for instance, to reproduce Kolmogorov's law of energy cascading. The same idea can also be applied in the more general context of fluid or smoke simulations in graphics by locally exciting or repressing certain frequencies of a simulation to produce a visually richer and/or desired look: the tools afforded by our operator-adapted basis functions and wavelets can help with a large spectrum of numerical tasks. A thorough investigation of operator-adapted wavelets for fluid dynamics is left to future work.

Image Processing. A variety of modern image processing tools that perform denoising, filtering, high dynamic range tone mapping, and contrast enhancement are based on the multiresolution concept: they decompose an input image into a

base and *detail* layers (that correspond to coarser and finer scales respectively), and process them separately before assembling an output image, see [77, 95, 96, 116, 143, 144]. The base-detail decomposition is often constructed using either non-adapted wavelets, or image-dependent anisotropic diffusion (e.g., through the edge-aware Perona-Malik operator [147], or an anisotropic Laplace operator in which the metric is controlled by variations in log-luminance [76]).

Our work provides a new approach to perform multiresolution decomposition adapted to anisotropic diffusion operators and has a lot of potential in improving and accelerating traditional image processing techniques thanks to its quasi-linear complexity. We hope to have results soon to confirm this expectation (see Fig. 5.21 for a preliminary test).

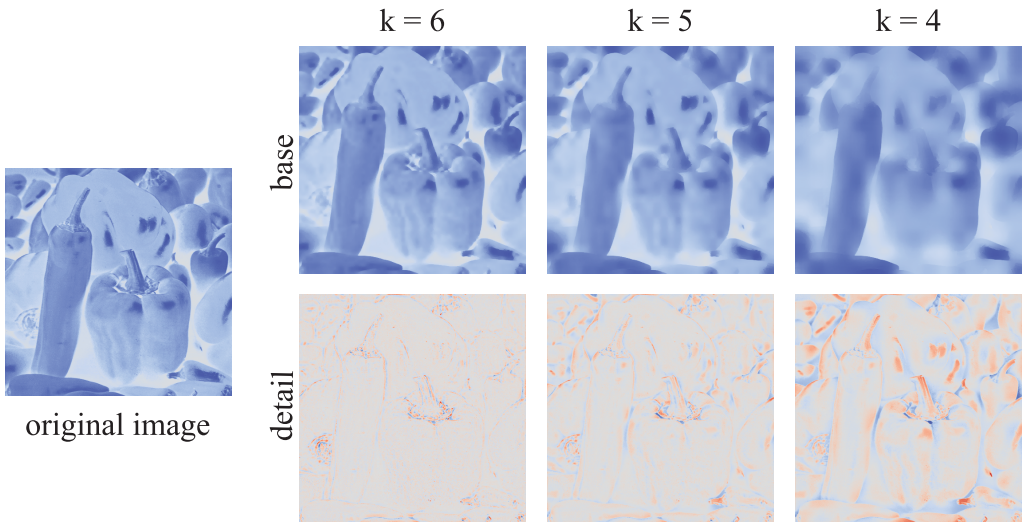


Figure 5.21: **Base-detail decomposition.** For an input image of 512×512 pixels (left) we perform a multiresolution decomposition adapted to Perona-Malik operator acting on 2-forms using $q = 9$ resolution levels. This complete decomposition can be directly leveraged in image processing tasks, but it can also be used to obtain a reduced base-detail representation. In particular, the base layer can be computed as a projection of the original image onto the k-th resolution level $\mathcal{V}^k$ (top row); observe the strong preservation of edges even for $k = 4$. The detail level, obtained as the difference between the original image and the base level (bottom row), can also be seen as a projection of the original image onto the wavelet space $\mathcal{W}^k \oplus_{\mathcal{L}} \ldots \oplus_{\mathcal{L}} \mathcal{W}^q$.

5.6 Conclusion

In this paper, we have introduced an operator-adapted multiresolution analysis for differential forms. Building over the recent notion of "gamblets", we presented an efficient fine-to-coarse construction, in log-linear time, of a hierarchy of basis

functions for differential forms and associated wavelets tailored to a given differential operator. Our numerical procedure involves only linear algebra, and can thus be easily implemented. This construction can be leveraged in finite element analysis for model reduction, numerical homogenization, or simply to provide a Galerkin discretization of an operator such that the resulting stiffness matrix becomes block diagonal, with uniformly well-conditioned and sparse blocks. Our method also leads to what we believe is the first constructive approach to obtain vector-valued wavelets that block diagonalize a given continuous, linear, bijective, positive-definite, and self-adjoint operator, where vector fields are expressed through their covariant counterparts, i.e., one-forms. Our construction is very general: it applies to various types of computational grids, different smoothness orders of refinable basis functions, many strategies of sparsification, etc. Depending on the precise application that one targets, this flexibility is beneficial, as it can accommodate a variety of specific numerical requirements.

Chapter 6

CONCLUSION

In this chapter we summarize the contributions presented in this thesis before discussing possible directions for future research on geometric model reduction.

6.1 Summary of Contributions

Our main contributions can be summarized as follows:

- We first formulated a local geometry processing method suitable for both manifold learning and nonlinear shape deformation. In the manifold learning setting, our proposed approach produces an embedding whose coordinate functions are as affine as possible within the local isometric parameterization of each small neighborhood. Formally, the embedding is obtained as a minimizer of the novel "multi-Laplacian" quadratic form, which characterizes geometry features of the data through all local relative positions of input points with respect to each other, subject to conditions of non-degeneracy. Computationally, our spectral affine-kernel embedding framework is based on local eigenanalysis of each neighborhood followed by a partial eigendecomposition of the global, sparse multi-Laplacian quadratic form. Our approach offers linear precision and is significantly more robust to irregular sampling and to reasonable amounts of noise than previous methods. In addition, we leverage the multi-Laplacian, augmented with user-prescribed constraints on select vertices, to formulate first spectral as-rigid-as-possible nonlinear deformation tool. It does not suffer from artifacts typical to alternative nonlinear deformation approaches and provides better convergence guarantees due to its eigen-based nature.

- Relying on the concepts of Levi-Civita connection and Cartan development, we also proposed a novel method for the efficient estimation of geodesic distances over point sets sampling connected compact manifolds through discrete parallel transport. This contribution was leveraged to formulate a global manifold learning method, based on partial eigenanalysis of a dense matrix and inspired by the Isomap algorithm. We demonstrated that our approach removes the restriction for geodesic convexity of the input data needed by the

Dijkstra graph-based shortest path estimator and significantly improves the accuracy of geodesic distance estimates in practice, while retaining the same computational complexity. Our parallel transport unfolding inherits resilience to strong noise characteristic of Isomap, recovering similar embeddings for geodesically convex input data. However, unlike Isomap, it does not suffer from large spurious deformations in the case of highly irregular sampling and non-geodesically convex inputs. This property is particularly important to the success of a subsequent landmark-based approximation of parallel transport unfolding, which can efficiently compute the low-dimensional embedding of large datasets in $O(n^2 \log n)$, even in the presence of noise.

- Finally, we have introduced an operator-adapted multiresolution analysis for differential forms, extending the recent notion of gamblets and providing an alternative axiomatic derivation of the gamblet transform motivated by finite-element considerations. Our efficient fine-to-coarse construction computes a hierarchy of basis functions for differential forms and associated wavelets adapted to a given continuous, linear, bijective, positive-definite, and self-adjoint differential operator using simple linear algebra in log-linear time. In particular, our method leads to the first constructive approach to obtain vector-valued operator-adapted wavelets, where vector fields are expressed through natural covariant counterparts – 1-forms. This multiresolution analysis can be leveraged in finite element framework to produce a Galerkin discretization with block diagonal stiffness matrix, composed of uniformly well-conditioned and sparse blocks, and further, to perform model reduction and numerical homogenization. Our construction can be applied to both linear and nonlinear differential equations, various types of computational grids, different smoothness orders of underlying refinable basis functions, and finite-element spaces satisfying linear differential constraints (e.g., divergence-freeness).

6.2 Future Work

Having introduced a number of new methods and concepts, we note that this work could be further extended in several interesting ways. Next, we discuss a few promising directions for future research.

High-dimensional Data Processing. With the increasing interest in data-driven computations, further extending current geometry processing tools to analyze, encode, and edit high dimensional datasets as we started in this thesis is an interesting

direction of future research. While we have assumed that the input data lie on a manifold of a given dimension d, approaches that estimate the right dimensionality from the data or from local scale selection [30, 51, 86, 115, 148] could prove valuable additions to our proposed approaches. However, the real data is sometimes better captured by a CW complex rather than a manifold, i.e., a set made out of regions of different dimensionality combined together. Developing a method for high accuracy estimation of geodesic distances in this case, or being able to compute low-dimensional parametrization of a CW complex with local or global geometric dimensionality reduction techniques would be a nice and useful extension. In addition, targeting geometric parametrization properties other than quasi-isometry (e.g., measure-preservation or conformality) could prove fruitful in certain applications. Out of sample extensions [18] and consolidation of high-dimensional datasets (where high dimensional points are denoised, resampled, and even inserted based on the voids in the reduced embedding as is commonly used for 3D pointsets [5]) may be worth exploring as well.

Even though differential geometry is rarely used in data analysis (a notable exception being in unsupervised domain adaptation [167]), its concepts may offer new insights addressing common data science problems or improve existing algortihms. In particular, using the notions of arbitrary non-Euclinean metrics, metric connections, parallel transport and geodesic distances can be further explored in the context of semi-supervised learning, graph-based clustering, and generalized manifold learning (CW complex learning).

Operator-adapted Multiresolution Analysis. Besides the large number of potential practical applications that our efficient construction of operator-adapted wavelets could impact, including model reduction in fluid simulation and image processing, a few obvious theoretical developments are worth investigating. For instance, current approach assumes that the operator $\mathcal{L}$ is self-adjoint and positive-definite – otherwise, we use $\mathcal{L}^T\mathcal{L}$. This fallback symmetrized operator results in squaring the condition number, which may not be optimal. Also, our treatment of arbitrary domain shapes within the computational grid is currently restricted to a low-order approximation of the Hodge star on the finest level. Finding higher-order spatial homogenizations of the Hodge star may be valuable to allow for broader applications. Finally, while we described our approach on "flat" domains, the notion of forms carries over naturally to non-flat domains (defined for instance through subdivision of cell complexes) as well, and a proper numerical treatment of the resulting Rie-

mannian metric induced by the embedding space can be efficiently achieved via a push-pull quadrature of the Hodge star at the finest level [62]. This should allow an even more general treatment of operator-adapted wavelets. In particular, the construction of scale spaces (multiresolution descriptions) of geometric shapes based on the Laplace-Beltrami operator for instance could also have important applications for geometric encoding [109], without requiring a global parameterization.

BIBLIOGRAPHY

[1] R. Abraham, J. E. Marsden, and R. Ratiu. *Manifolds, Tensor Analysis, and Applications: 2nd Edition*. Springer-Verlag, 1988.

[2] A. Agarwal, A. Anandkumar, P. Jain, and P. Netrapalli. Learning sparsely used overcomplete dictionaries via alternating minimization. *SIAM Journal on Optimization*, 26(4):2775–2799, 2016.

[3] A. Agarwal, A. Anandkumar, and P. Netrapalli. A clustering approach to learning sparsely used overcomplete dictionaries. *IEEE Transactions on Information Theory*, 63(1):575–592, 2017. ISSN 0018-9448.

[4] Michal Aharon, Michael Elad, Alfred Bruckstein, et al. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006.

[5] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Trans. Vis. Comput. Graphics*, 9(1):3–15, 2003.

[6] Kevin Amaratunga, John R. Williams, Sam Qian, and John Weiss. Wavelet-Galerkin solutions for one-dimensional partial differential equations. *International Journal for Numerical Methods in Engineering*, 37(16):2703–2716, 1994.

[7] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, 22(4):81–504, 1999.

[8] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[9] D. N. Arnold, R. S. Falk, and R. Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155, 2006.

[10] Douglas N. Arnold. Spaces of finite element differential forms. In U. Gianazza, F. Brezzi, P. Colli Franzone, and G. Gilardi, editors, *Analysis and Numerics of Partial Differential Equations*, pages 117–140. Springer, 2013.

[11] Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Finite element exterior calculus: from Hodge theory to numerical stability. *Bull. Amer. Math. Soc. (N.S.)*, 47:281–354, 2010.

[12] Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. 2015.

[13] Jackie Assa, Yaron Caspi, and Daniel Cohen-Or. Action synopsis: Pose selection and illustration. *ACM Trans. Graph.*, 24(3):667–676, July 2005.

[14] Ivo Babuška and John E. Osborn. Can a finite element method perform arbitrarily badly? *Mathematics of Computation*, 69(230):443–462, 2000.

[15] E. Bacry, S. Mallat, and G. Papanicolaou. A wavelet space-time adaptive scheme for partial differential equations. In *Progress in Wavelet Analysis and Applications (Toulouse, 1992)*, pages 677–682. Frontières, Gif-sur-Yvette, 1993.

[16] Boaz Barak, Jonathan A Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 143–151. ACM, 2015.

[17] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, pages 585–591, 2001.

[18] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In *Neural Information Processing Systems*, pages 177–184, 2003.

[19] S. Bertoluzza, Y. Maday, and J.-C. Ravel. A dynamically adaptive wavelet method for solving partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 116(1-4):293–299, 1994. ISSN 0045-7825. ICOSAHOM'92 (Montpellier, 1992).

[20] G. Beylkin. On multiresolution methods in numerical analysis. *Doc. Math., Extra*, 3:481–490, 1998.

[21] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Communications on Pure and Applied Mathematics*, 44(2): 141–183, 1991.

[22] Gregory Beylkin and Nicholas Coult. A multiresolution strategy for reduction of elliptic PDEs and eigenvalue problems. *Applied and Computational Harmonic Analysis*, 5(2):129–155, 1998.

[23] Pavel B. Bochev and James M. Hyman. Principles of mimetic discretizations of differential operators. In Douglas N. Arnold and et al., editors, *Compatible Spatial Discretizations*, pages 89–119. Springer New York, 2006.

[24] Ingwer Borg and Patrick J. F. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 3rd edition, 2005.

[25] A. Bossavit. Whitney forms: a class of finite elements for three-dimensional computations in electromagnetism. *IEEE Proceedings A*, 135(8):493–500, 1988.

[26] Alain Bossavit. Mixed finite elements and the complex of whitney forms. *The mathematics of finite elements and applications VI*, pages 137–144, 1988.

[27] Mario Botsch and Leif Kobbelt. An intuitive framework for real-time freeform modeling. *ACM Trans. Graph.*, 23(3):630–634, 2004.

[28] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE Trans. Vis. Comp. Graph.*, 14(1):213–230, 2008.

[29] Mario Botsch, Mark Pauly, Markus Gross, and Leif Kobbelt. PriMo: Coupled prisms for intuitive surface modeling. In *Symp. Geom. Proc.*, pages 11–20, 2006.

[30] Matthew Brand. Charting a manifold. In *Neural Information Processing Systems*, pages 985–992, 2003.

[31] Matthew Brand. Fast low-rank modifications of the thin Singular Value Decomposition. *Linear Algebra and its Applications*, 415(1):20–30, 2006.

[32] Ulrik Brandes and Christian Pich. Eigensolver methods for progressive multidimensional scaling of large data. In Michael Kaufmann and Dorothea Wagner, editors, *International Symposium on Graph Drawing*, pages 42–53. 2007.

[33] M.E. Brewster and G. Beylkin. A multiresolution strategy for numerical homogenization. *Applied and Computational Harmonic Analysis*, 2(4):327–349, 1995.

[34] M. R. Brito, E. L. Chávez, A. J. Quiroz, and J. E. Yukich. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, 35(1):33–42, 1997.

[35] A. Buffa, J. Rivas, G. Sangalli, and R. Vázquez. Isogeometric discrete differential forms in three dimensions. *SIAM Journal on Numerical Analysis*, 49(2):818–844, 2011.

[36] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224(2):880–896, 2007.

[37] James R. Bunch, Christopher P. Nielsen, and Danny C. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31 (1):31–48, 1978.

[38] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. In *Proc. ACM SIGGRAPH Conf.*, pages 263–270, 1993.

[39] J. M. Carnicer, W. Dahmen, and J. M. Peña. Local decomposition of refinable spaces and wavelets. *Applied and Computational Harmonic Analysis*, 3(2): 127–153, 1996.

[40] Élie Cartan. Les systèmes différentiels extérieurs et leurs applications géométriques. 1945.

[41] Lawrence Cayton and Sanjoy Dasgupta. Robust euclidean embedding. In *International Conference on Machine Learning*, pages 169–176, 2006.

[42] Jiong Chen, Hujun Bao, Tianyu Wang, Mathieu Desbrun, and Jin Huang. Numerical coarsening using discontinuous shape functions. *ACM Trans. Graph.*, 37(4):Art. 117, 2018.

[43] Zhonggui Chen, Ligang Liu, Zhengyue Zhang, and Guojin Wang. Surface parameterization via aligning optimal local flattening. In *ACM Symposium on Solid and Physical Modeling*, pages 291–296, 2007.

[44] G. Chiavassa and J. Liandrat. A fully adaptive wavelet algorithm for parabolic partial differential equations. *Appl. Numer. Math.*, 36(2-3):333–358, 2001. ISSN 0168-9274.

[45] Heeyoul Choi and Seungjin Choi. Robust kernel isomap. *Pattern Recognition*, 40(3):853–862, 2007.

[46] Snorre H. Christiansen and Francesca Rapetti. On high order finite element spaces of differential forms. *Math. Comp.*, 85:517–548, 2016.

[47] P. Ciarlet. *The Finite Element Method for Elliptic Problems*. SIAM, 2002.

[48] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992.

[49] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods for elliptic operator equations: convergence rates. *Math. Comp.*, 70(233):27–75, 2001. ISSN 0025-5718.

[50] Ronald R. Coifman, Stéphane Lafon, Ann B. Lee, Mauro Maggioni, F Frederick J. Warner, and Steven W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In *National Academy of Sciences*, volume 102(21), pages 7426–7431, 2005.

[51] J. A. Costa and A. O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. Signal Process.*, 52(8): 2210–2221, 2004.

[52] Keenan Crane, Mathieu Desbrun, and Peter Schröder. Trivial connections on discrete surfaces. *Comp. Graph. Forum*, 29(5):1525–1533, 2010.

[53] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph.*, 32(5):Art. 152, October 2013.

[54] G. M. Crippen and T. F. Havel. Stable calculation of coordinates from distance information. *Acta Crystallographica Section A*, 34(2):282–284, 1978.

[55] S. Dahlke and I. Weinreich. Wavelet-Galerkin methods: an adapted biorthogonal wavelet basis. *Constr. Approx.*, 9(2-3):237–262, 1993. ISSN 0176-4276.

[56] S. Dahlke and I. Weinreich. Wavelet bases adapted to pseudodifferential operators. *Appl. Comput. Harmon. Anal.*, 1(3):267–283, 1994. ISSN 1063-5203.

[57] W. Dahmen and A. Kunoth. Adaptive wavelet methods for linear-quadratic elliptic control problems: convergence rates. *SIAM J. Control Optim.*, 43(5): 1640–1675, 2005. ISSN 0363-0129.

[58] W. Dahmen, H. Harbrecht, and R. Schneider. Compression techniques for boundary integral equations—asymptotically optimal complexity estimates. *SIAM Journal on Numerical Analysis*, 43(6):2251–2271, 2006.

[59] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Symposium on Computational Geometry*, pages 253–262, 2004.

[60] Fernando de Goes, Pierre Alliez, Houman Owhadi, and Mathieu Desbrun. On the equilibrium of simplicial masonry structures. *ACM Trans. Graph.*, 32 (4):Art. 93, 2013.

[61] Fernando de Goes, Beibei Liu, Max Budninskiy, Yiying Tong, and Mathieu Desbrun. Discrete 2-tensor fields on triangulations. *Comput. Graph. Forum*, 33(5):13–24, 2014.

[62] Fernando de Goes, Mathieu Desbrun, Mark Meyer, and Tony DeRose. Subdivision exterior calculus for geometry processing. *ACM Trans. Graph.*, 35 (4):Art. 133, 2016.

[63] Vin De Silva and Joshua B. Tenenbaum. Global vs. local methods in nonlinear dimensionality reduction. In *NIPS*, pages 705–712, 2002.

[64] Vin De Silva and Joshua B Tenenbaum. Sparse multidimensional scaling using landmark points. Report, Technical report, Stanford University, 2004.

[65] Erwan Deriaz and Valérie Perrier. Divergence-free and curl-free wavelets in two dimensions and three dimensions: application to turbulent flows. *Journal of Turbulence*, 7:N3, 2006.

[66] Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, 21(3):209–218, 2003.

[67] Mathieu Desbrun, Eva Kanso, and Yiying Tong. Discrete differential forms for computational modeling. In Alexander Bobenko and et al., editors, *Discrete Differential Geometry*, pages 287–324. Birkhäuser Basel, 2008.

[68] Mathieu Desbrun, Eva Kanso, and Yiying Tong. Discrete differential forms for computational modeling. In A. Bobenko et al., editor, *Discrete Differential Geometry*, pages 287–324. Birkhäuser Basel, 2008.

[69] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[70] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995.

[71] David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

[72] David L. Donoho and Iain M. Johnstone. Minimax estimation via wavelet shrinkage. *Ann. Statist.*, 26(3):879–921, 06 1998.

[73] Mihai Dorobantu and Björn Engquist. Wavelet-based numerical homogenization. *SIAM Journal on Numerical Analysis*, 35(2):540–559, 1998.

[74] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *Rev. Francaise Automat. Informat. Recherche Operationnelle Ser. RAIRO Analyse Numerique*, 10(R-3):5–12, 1976. ISSN 0399-0516.

[75] B. Engquist, S. Osher, and S. Zhong. Fast wavelet based algorithms for linear evolution equations. *SIAM Journal on Scientific Computing*, 15(4):755–775, 1994.

[76] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 67:1–67:10, New York, NY, USA, 2008. ACM.

[77] Raanan Fattal. Edge-avoiding wavelets and their applications. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 22:1–22:10, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-726-4.

[78] Leman Feng, Pierre Alliez, Laurent Busé, Hervé Delingette, and Mathieu Desbrun. Curved optimal delaunay triangulation. *ACM Trans. Graph.*, 37(4): Art. 61, 2018.

[79] Theodore Frankel. *The Geometry of Physics: An Introduction*. Cambridge University Press, 2011.

[80] J. Fröhlich and K. Schneider. An adaptive wavelet Galerkin algorithm for one- and two-dimensional flame computations. *European J. Mech. B Fluids*, 13(4):439–471, 1994. ISSN 0997-7546.

[81] T. Gantumur and R. Stevenson. Computation of differential operators in wavelet coordinates. *Mathematics of Computation*, 75(254):697–709, 2006.

[82] E.S. Gawlik, P. Mullen, D. Pavlov, J.E. Marsden, and M. Desbrun. Geometric, variational discretization of continuum theories. *Physica D: Nonlinear Phenomena*, 240(21):1724–1760, 2011.

[83] Marc Gerritsma. Edge functions for spectral element methods. In *Spectral and High Order Methods for Partial Differential Equations*, pages 199–207. Springer, 2011.

[84] Andrew Gillette, Tyler Kloefkorn, and Victoria Sanders. Computational Serendipity and Tensor Product Finite Element Differential Forms. *ArXiv e-prints*, math.NA 1806.00031, May 2018.

[85] D. Gines, G. Beylkin, and J. Dunn. LU factorization of non-standard forms and direct multiresolution solvers. *Applied and Computational Harmonic Analysis*, 5(2):156–201, 1998.

[86] Simon Giraudot, David Cohen-Steiner, and Pierre Alliez. Noise-adaptive shape reconstruction from raw point sets. *Comput. Graph. Forum*, 32(5): 229–238, 2013.

[87] Yair Goldberg and Ya'acov Ritov. LDR-LLE: LLE with low-dimensional neighborhood representation. In *Symposium on Visual Computing*, pages 43–54, 2008.

[88] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd ed.)*. Johns Hopkins University Press, 1996.

[89] L. J. Grady and J. R. Polimeni. Springer-Verlag London Limited, 2010.

[90] Eitan Grinspun. *The basis refinement method*. PhD thesis, California Institute of Technology, Engineering and Applied Science, 2003.

[91] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[92] R. L. Harder and R. N. Desmarais. Interpolation using surface splines. *J. Aircraft*, 9:189–191, 1972.

[93] David Harel and Yehuda Koren. Graph drawing by high-dimensional embedding. In *International Symposium on Graph Drawing*, pages 207–219, 2002.

[94] Francis H. Harlow and J. Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, 1965.

[95] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, June 2013.

[96] Charles Hessel. *The automatic decomposition of an image in base and detail : Application to contrast enhancement*. PhD thesis, Université Paris-Saclay, May 2018.

[97] Klaus Hildebrandt, Christian Schulz, Christoph Von Tycowicz, and Konrad Polthier. Interactive surface modeling using modal analysis. *ACM Trans. Graph.*, 30(5):Art. 119, 2011.

[98] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. ISSN 0036-8075.

[99] R. Hiptmair. Higher order whitney forms. *Progress In Electromagnetics Research*, 32:271–299, 2001.

[100] R. Hiptmair. Finite elements in computational electromagnetism. *Acta Numer.*, 11:237–339, 2002.

[101] Anil Hirani. *Discrete Exterior Calculus*. PhD thesis, Computer Science, California Institute of Technology, 2003.

[102] Anil N. Hirani, Kalyana B. Nakshatrala, and Jehanzeb H. Chaudhry. Numerical method for Darcy flow derived using Discrete Exterior Calculus. *International Journal for Computational Methods in Engineering Science & Mechanics*, 16(3):151–169, 2015.

[103] Kai Hormann, Konrad Polthier, and Alia Sheffer. Mesh parameterization: Theory and practice. In *ACM SIGGRAPH ASIA Course Notes*, pages 12:1–12:87, 2008.

[104] Jin Huang, Xiaohan Shi, Xinguo Liu, Kun Zhou, Li-Yi Wei, Shang-Hua Teng, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Subspace gradient domain mesh deformation. *ACM Trans. Graph.*, 25(3):1126–1134, 2006.

[105] Jin Huang, Yiying Tong, Kun Zhou, Hujun Bao, and Mathieu Desbrun. Interactive shape interpolation through controllable dynamic deformation. *IEEE Trans. Vis. Comput. Graphics*, 17(7):983–992, 2011.

[106] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method—a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1-2):3–24, 1998.

[107] K. Hui, C. Wang, and B. Xiao. Globally-preserving based locally linear embedding. In *International Conference on Pattern Recognition*, pages 531–534, 2010.

[108] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.

[109] Andrei Khodakovsky, Peter Schröder, and Wim Sweldens. Progressive geometry compression. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 271–278, 2000.

[110] Marc Khoury, Yifan Hu, Shankar Krishnan, and Carlos Scheidegger. Drawing large graphs by low-rank stress majorization. *Computer Graphics Forum*, 31 (3.1):975–984, 2012.

[111] Sung-Ho Kim. Generation of expression space for realtime facial expression control of 3d avatar. In André Gagalowicz and Wilfried Philips, editors, *Proceedings of MIRAGE*, pages 318–329. 2007.

[112] W. Kohn. Analytic properties of Bloch waves and Wannier functions. *Physical Review*, 115(4):809, 1959.

[113] Ralf Kornhuber and Harry Yserentant. Numerical homogenization of elliptic multiscale problems by subspace decomposition. *Multiscale Modeling & Simulation*, 14(3):1017–1036, 2016.

[114] Artiom Kovnatsky, Klaus Glashoff, and Michael M. Bronstein. MADMM: A generic algorithm for non-smooth optimization on manifolds. In *European Conference on Computer Vision*, pages 680–696, 2016.

[115] Elizaveta Levina and Peter J Bickel. Maximum likelihood estimation of intrinsic dimension. In *NIPS*, pages 777–784, 2004.

[116] Yuanzhen Li, Lavanya Sharan, and Edward H. Adelson. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. Graph.*, 24(3):836–844, July 2005. ISSN 0730-0301.

[117] LightScape. Light Stage Data Gallery, University of Southern California's Institute for Creative Technologies; `http://gl.ict.usc.edu/Data/LightStage/`, 2016.

[118] Binbin Lin, Xiaofei He, Chiyuan Zhang, and Ming Ji. Parallel vector field embedding. *Journal of Machine Learning Research*, 14:2945–2977, 2013.

[119] Yaron Lipman, Olga Sorkine, David Levin, and Daniel Cohen-Or. Linear rotation-invariant coordinates for meshes. *ACM Trans. Graph.*, 24(3):479–487, 2005.

[120] Beibei Liu, Gemma Mason, Julian Hodgson, Yiying Tong, and Mathieu Desbrun. Model-reduced variational fluid simulation. *ACM Trans. Graph.*, 34(6):Art. 244, October 2015.

[121] Beibei Liu, Yiying Tong, Fernando De Goes, and Mathieu Desbrun. Discrete connection and covariant derivative for vector field analysis and design. *ACM Trans. Graph.*, 35(3):23:1–23:17, 2016.

[122] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics (TOG)*, 16(1):34–73, 1997.

[123] Van Der Maaten and Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 01 2008.

[124] A. Målqvist and D. Peterseim. Localization of elliptic multiscale problems. *Mathematics of Computation*, 83(290):2583–2603, 2014.

[125] N. Marzari and D. Vanderbilt. Maximally localized generalized Wannier functions for composite energy bands. *Physical Review B*, 56(20):12847, 1997.

[126] Eivind Lyche Melvær and Martin Reimers. Geodesic polar coordinates on polygonal meshes. *Computer Graphics Forum*, 31(8):2423–2435, 2012.

[127] Y. Meyer. *Wavelets and Operators*, volume 1. Cambridge university press, 1995.

[128] C. A. Micchelli and T. J. Rivlin. A survey of optimal recovery. In *Optimal Estimation in Approximation Theory*, pages 1–54. Springer, 1977.

[129] Mamdouh S. Mohamed, Anil N. Hirani, and Ravi Samtaney. Discrete exterior calculus discretization of incompressible navier–stokes equations over surface simplicial meshes. *Journal of Computational Physics*, 312:175–191, 2016.

[130] Patrick Mullen, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. Spectral conformal parameterization. In *Symp. Geom. Proc.*, pages 1487–1494, 2008.

[131] Patrick Mullen, Alexander McKenzie, Dmitry Pavlov, Luke Durant, Yiying Tong, Eva Kanso, Jerold E. Marsden, and Mathieu Desbrun. Discrete Lie advection of differential forms. *Foundations of Computational Mathematics*, 11(2):131–149, 2011.

[132] James R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Menlo Park, CA, 1984.

[133] J.-C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numer. Math.*, 35(3):315–341, 1980.

[134] J.-C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numer. Math.*, 35:315–341, 1980.

[135] Yen Ting Ng, Chohong Min, and Frédéric Gibou. An efficient fluid-solid coupling algorithm for single-phase flows. *J. Comput. Phys.*, 228(23):8807–8829, 2009.

[136] Katsumi Nomizu. Kinematics and differential geometry of submanifolds. *Tohoku Math. J.*, 30(4):623–637, 1978.

[137] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381 (6583):607, 1996.

[138] H. Owhadi and C. Scovel. Universal scalable robust solvers from computational information games and fast eigenspace adapted multiresolution analysis. *arXiv:1703.10761*, 2017.

[139] H. Owhadi and C. Scovel. *Operator adapted wavelets, fast solvers, and numerical homogenization from a game theoretic approach to numerical approximation and algorithm design*. 2018. To appear.

[140] Houman Owhadi. Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games. *SIAM Review*, 59(1):99–149, 2017.

[141] Houman Owhadi and Lei Zhang. Gamblets for opening the complexity-bottleneck of implicit schemes for hyperbolic and parabolic ODEs/PDEs with rough coefficients. *Journal of Computational Physics*, 347:99–128, 2017.

[142] Houman Owhadi, Lei Zhang, and Leonid Berlyand. Polyharmonic homogenization, rough polyharmonic splines and sparse super-localization. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(2):517–552, 2014.

[143] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Fŕedo Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1):1–73, 10 2009.

[144] Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *Commun. ACM*, 58 (3):81–91, February 2015.

[145] JinHyeong Park, Zhenyue Zhang, Hongyuan Zha, and R. Kasturi. Local smoothing for manifold learning. In *Computer Vision and Pattern Recognition*, volume 2, pages 452–459, 2004.

[146] Dmitry Pavlov, Patrick Mullen, Yiying Tong, Eva Kanso, Jerrold E. Marsden, and Mathieu Desbrun. Structure-preserving discretization of incompressible fluids. *Physica D: Nonlinear Phenomena*, 240(6):443–458, 2011.

[147] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[148] Karl W. Pettis, Thomas A. Bailey, Anil K. Jain, and Richard C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Trans. Pattern Anal. Mach. Intell*, 1(1):25–37, 1979.

[149] Robert Pless. Image spaces and video trajectories: using isomap to explore video sequences. In *IEEE International Conference on Computer Vision*, pages 1433–1440, 2003.

[150] Y Qiu, Gael Guennebaud, and Jitse Niesen. Spectra. http://yixuan.cos.name/spectra, 2015.

[151] Francesca Rapetti and Alain Bossavit. Whitney forms of higher degree. *SIAM J. Numer. Anal.*, 47(3):2369–2386, 2009.

[152] P.-A. Raviart and J. M. Thomas. A mixed finite element method for 2nd order elliptic problems. 606:292–315, 1977.

[153] N. Robidoux. Polynomial hitopolation, superconvergent degrees of freedom, and pseudospectral discrete hodge operators, 2008. Unpublished.

[154] Guy Rosman, Michael M. Bronstein, Alexander M. Bronstein, and Ron Kimmel. Nonlinear dimensionality reduction by topologically constrained isometric embedding. *International Journal of Computer Vision*, 89:56–68, 2010.

[155] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[156] Dzhelil Rufat. Licpy: Line integral convolution with python, 2018. `https://github.com/drufat/licpy`.

[157] Dzhelil Rufat, Gemma Mason, Patrick Mullen, and Mathieu Desbrun. The chain collocation method: A spectrally accurate calculus of forms. *Journal of Computational Physics*, 257:1352–1372, January 2014.

[158] Florian Schäfer, Timothy John Sullivan, and Houman Owhadi. Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. *CoRR*, abs/1706.02205, 2017.

[159] R. Schmidt. Stroke parameterization. *Computer Graphics Forum*, 32(2): 255–263, 2013.

[160] Ryan Schmidt and Karan Singh. Approximate conformal parameterization of point-sampled surfaces. Report, Technical report CSRG-605, Computer Science Department, University of Toronto, 2009.

[161] Ryan Schmidt, Cindy Grimm, and Brian Wyvill. Interactive decal compositing with discrete exponential maps. *ACM Trans. Graph.*, 25(3):605–613, 2006.

[162] Scikit. Scikit Learn Machine Learning in Python, sklearn.manifold module; `http://scikit-learn.org/`, 2016.

[163] Bl. Sendov. Adapted multiresolution analysis and wavelets. In *Functions, series, operators (Budapest, 1999)*, pages 23–38. János Bolyai Math. Soc., Budapest, 2002.

[164] R.W. Sharpe. *Differential Geometry: Cartan's Generalization of Klein's Erlangen Program*, volume 166 of *Graduate Texts in Mathematics*. Springer-Verlag New York, 1997.

[165] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell*, 22(8):888–905, 2000.

[166] Xiaohan Shi, Kun Zhou, Yiying Tong, Mathieu Desbrun, Hujun Bao, and Baining Guo. Mesh puppetry: Cascading optimization of mesh deformation with inverse kinematics. *ACM Trans. Graph.*, 26(3):Art. 81, 2007.

[167] Ashish Shrivastava, Sumit Shekhar, and Vishal M. Patel. Unsupervised domain adaptation using parallel transport on Grassmann manifold. In *IEEE Winter Conference on Applications of Computer Vision*, pages 277–284, 2014.

[168] A. Singer and H.-T. Wu. Vector diffusion maps and the connection laplacian. *Communications on Pure and Applied Mathematics*, 65(8):1067–1144, 2012.

[169] Yong Sheng Soh and Venkat Chandrasekaran. Learning semidefinite regularizers. *Foundations of Computational Mathematics*, 2018.

[170] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Symp. Geom. Proc.*, pages 175–184, 2004.

[171] Olga Sorkine. Laplacian mesh processing. In *Eurographics State of the Art Reports*, 2005.

[172] Olga Sorkine. Differential representations for mesh processing. *Comput. Graph. Forum*, 25:789–807, 2006.

[173] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symp. Geom. Proc.*, pages 109–116, 2007.

[174] Michael Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 1. Publish of Perish, Inc., second edition, 1979.

[175] Jos Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Proceedings of SIGGRAPH Conference*, pages 395–404, 1998.

[176] Ari Stern, Yiying Tong, Mathieu Desbrun, and Jerrold E. Marsden. Variational integrators for Maxwell's equations with sources. *PIERS Online*, 4(7): 711–715, 2008.

[177] R. Stevenson. Adaptive wavelet methods for solving operator equations: an overview. In *Multiscale, Nonlinear and Adaptive Approximation*, pages 543–597. Springer, 2009.

[178] Raghunathan Sudarshan. *Operator-adapted Finite Element Wavelets: theory and applications to a posteriori error estimation and adaptive computational modeling*. PhD thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, 2005.

[179] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998.

[180] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319–2323, 2000.

[181] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. Designing quadrangulations with discrete harmonic forms. In *Symposium on Geometry Processing*, pages 201–210, 2006.

[182] Warren S. Torgerson. Multidimensional scaling of similarity. *Psychometrika*, 30(4):379–393, 1965.

[183] Karsten Urban. On divergence-free wavelets. *Advances in Computational Mathematics*, 4(1):51–81, 1995.

[184] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative review. *J. Mach. Learn. Res.*, 10:66–71, 2009.

[185] O. V. Vasilyev and S. Paolucci. A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. *J. Comput. Phys.*, 125(2):498–512, 1996. ISSN 0021-9991.

[186] P. S. Vassilevski. General constrained energy minimization interpolation mappings for AMG. *SIAM J. Sci. Comput.*, 32(1):1–13, 2010. ISSN 1064-8275.

[187] P. S. Vassilevski and J. Wang. Stabilizing the hierarchical basis by approximate wavelets. I. Theory. *Numer. Linear Algebra Appl.*, 4(2):103–126, 1997. ISSN 1070-5325.

[188] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[189] Jiaping Wang, Xin Tong, Stephen Lin, Minghao Pan, Chao Wang, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Appearance manifolds for modeling time-variant appearance of materials. *ACM Trans. Graph.*, 25(3):754–761, July 2006.

[190] Ke Wang. *A subdivision approach to the construction of smooth differential forms*. PhD thesis, Applied Mathematics, California Institute of Technology, 2008.

[191] Ke Wang, Weiwei, Yiying Tong, Mathieu Desbrun, and Peter Schröder. Edge subdivision schemes and the construction of smooth vector fields. *ACM Trans. Graph.*, 25(3):1041–1048, 2006.

[192] Yu Wang, Alec Jacobson, Jernej Barbič, and Ladislav Kavan. Linear subspace design for real-time shape deformation. *ACM Trans. Graph.*, 34(4):Art. 57, July 2015.

[193] G. H. Wannier. Dynamics of band electrons in electric and magnetic fields. *Reviews of Modern Physics*, 34(4):645, 1962.

[194] André Weil. Sur les théorèmes de de Rham. *Comment. Math. Helv.*, 26: 119–145, 1952.

[195] H. Whitney. *Geometric integration theory*. Princeton University Press, 1957.

[196] Hassler Whitney. *Geometric integration theory*. Princeton University Press, 1957.

[197] Hehu Xie, Lei Zhang, and Houman Owhadi. Fast eigenpairs computation with operator adapted wavelets and hierarchical subspace correction. *arXiv preprint arXiv:1806.00565*, 2018.

[198] Xianglei Xing, Sidan Du, and Kejun Wang. Robust Hessian locally linear embedding techniques for high-dimensional data. *Algorithms*, 9(2):36, 2016.

[199] Qiang Ye and Weifeng Zhi. Discrete Hessian Eigenmaps method for dimensionality reduction. *Journal of Computational and Applied Mathematics*, 278:197–212, 2015.

[200] Gene Ryan Yoo and Houman Owhadi. De-noising by thresholding operator adapted wavelets. *arXiv preprint arXiv:1805.10736*, 2018.

[201] Hao (Richard) Zhang and Daniel Cohen-Or. Dimensionality reduction. In *A Sampler of Useful Computational Tools for Applied Geometry, Graphics, and Image Processing*, chapter 9, pages 131–145. CRC Press, 2015.

[202] Teng Zhang and Gilad Lerman. A novel M-estimator for robust PCA. *J. Mach. Learn. Res.*, 15(1):749–808, January 2014.

[203] Zhenyue Zhang and Jing Wang. MLLE: Modified locally linear embedding using multiple weights. In *NIPS*, pages 1593–1600, 2007.

[204] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Sci. Comput.*, 26(1):313–338, 2004.

[205] Kun Zhou, Jin Huang, John Snyder, Xinguo Liu, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Large mesh deformation using the volumetric graph Laplacian. *ACM Trans. Graph.*, 24(3):496–503, 2005.

[206] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans. Vis. Comp. Graph.*, 8 (2):198–207, Apr 2002.

[207] Denis Zorin and Peter Schröder. Subdivision for modeling and animation, 2000. Course notes at ACM SIGGRAPH Conference.

Appendix A

PROOF OF PROPOSITION 4.3

Before formulating a precise error bound on our geodesic approximation to quantify Proposition 4.3, we first review existing results and state a few reasonable assumptions on the sampling $\mathcal{S}$ of the manifold $\mathcal{M}$.

A similar notion of discrete parallel transport was considered by the authors of [168] in order to define and prove convergence of a connection Laplacian operator. We will make use of one of the theorems they proved towards their goal, with a proof (omitted here) relying on geometric properties of parallel transport and probabilistic guarantees (obtained through Bernstein's inequality) on the quality of tangent bases approximation. Following their notations, let $\varphi : \mathcal{M} \hookrightarrow \mathbb{R}^D$ be the embedding of a smooth and compact Riemannian d-manifold $\mathcal{M}$ with its metric induced from the embedding space $\mathbb{R}^D$. Points from $\mathcal{S}$ are considered to be sampled from $\mathcal{M}$ according to a probability density function $p \in C^3(\mathcal{M})$. Denote the tangent bundle of $\mathcal{M}$ by $T\mathcal{M}$, the tangent space at point $\mathbf{x}_i$ by $T_i\mathcal{M}$, the differential of the embedding at $\mathbf{x}_i$ by $d\varphi_i : T_i\mathcal{M} \mapsto \mathbb{R}^D$ and the parallel transport operator from $\mathbf{x}_j$ to $\mathbf{x}_j$ along a geodesic connecting them by $\mathbf{P}_{i,j} : T_j\mathcal{M} \mapsto T_i\mathcal{M}$. Let ε be the maximum radius of the geodesic K-neighborhoods used to construct our approximate tangent bases $\{\mathbb{T}_i\}_{i=1}^n$ (see Eq. (4.1) and its description) and finally, let $\varepsilon_g > 0$.

Theorem *(from [168]). Assume $\varepsilon = O(n^{-\frac{1}{d+2}})$ and consider two points $\mathbf{x}_i$ and $\mathbf{x}_j$ of $\mathcal{M}$, such that the geodesic distance between them is $O(\varepsilon_g)$. Then for any vector $\mathbf{u}_j \in T_j\mathcal{M}$, with high probability:*

$$\mathbf{R}_{i,j}\mathbb{T}_j^T d\varphi_j \left[\mathbf{u}_j\right] = \mathbb{T}_i^T d\varphi_i \left[\mathbf{P}_{i,j}\mathbf{u}_j\right] + O(\gamma_{i,j}), \tag{A.1}$$

where $\gamma_{i,j} = \varepsilon^3 + \varepsilon_g^3$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are at least ε-away from the boundary of $\mathcal{M}$ and $\gamma_{i,j} = \varepsilon + \varepsilon_g^3$ otherwise.

In addition to this theorem, we will use two more assumptions:

I. Let $\mathbf{c}_p = (\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_m})$ be the shortest polyline connecting $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_m}$ in the proximity graph G. First, we assume that the polyline is included in an ε_d-thickening of the real geodesic $\mathbf{c}_g$ connecting the same endpoints, where ε_d is a positive constant such that $\varepsilon_d^2 \kappa_s \ll 1$, with κ_s denoting the maximum absolute value of intrinsic sectional curvature of $\mathcal{M}$. Note that this condition

is less stringent than the one used to prove convergence of Dijkstra polylines to geodesic curves [180]: sampling voids of maximum geodesic diameter ε_d are allowed in the input pointset $\mathcal{S}$.

II. Let $\mathbf{g}_{i_s}$ be a tangent vector in $T_{i_s}\mathcal{M}$ that connects the endpoints of the Cartan development of the geodesic curve between $\mathbf{x}_{i_s}$ and $\mathbf{x}_{i_{s+1}}$ onto $T_{i_s}\mathcal{M}$. Denoting $\mathbf{e}_{i_s} = (\mathbf{x}_{i_{s+1}} - \mathbf{x}_{i_s})$, we will also assume:

$$\mathbb{T}_{i_s}^T d\varphi_{i_s} \mathbf{g}_{i_s} = \mathbb{T}_{i_s}^T \mathbf{e}_{i_s} + O(\gamma_{i_s,i_{s+1}}) \tag{A.2}$$

This condition links local curvature and sampling density: the projection of $\mathbf{e}_{i_s}$ onto the approximate tangent basis $\mathbb{T}_{i_s}$ must be close to the unwrapped geodesic in the same basis. Note that the length rescaling step (Eq. (4.2.4)) can help in practice to tighten the error bound $O(\gamma_{i_s,i_{s+1}})$.

Proposition 4.3 revisited: Under assumptions I, II and the assumptions of the theorem, the PTU estimate d_{PTU} of the geodesic distance between points $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_m}$, based on a Dijkstra shortest polyline $\mathbf{c}_p = (\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_m})$ on G, provides an approximation of the length d_g of the geodesic curve $\mathbf{c}_g$ with the same endpoints in the following sense:

$$d_{\mathrm{PTU}}(\mathbf{x}_{i_1}, \mathbf{x}_{i_m}) = d_g(\mathbf{x}_{i_1}, \mathbf{x}_{i_m}) + O(\delta),$$

where $O(\delta)$ is between $O(m^2(\varepsilon + \varepsilon_g^3 + \varepsilon_g \varepsilon_d^2 \kappa_s))$ and $O(m^2(\varepsilon^3 + \varepsilon_g^3 + \varepsilon_g \varepsilon_d^2 \kappa_s))$ depending on how many polyline segments are close to the manifold boundary.

Proof: First, observe that condition (A.2) implies

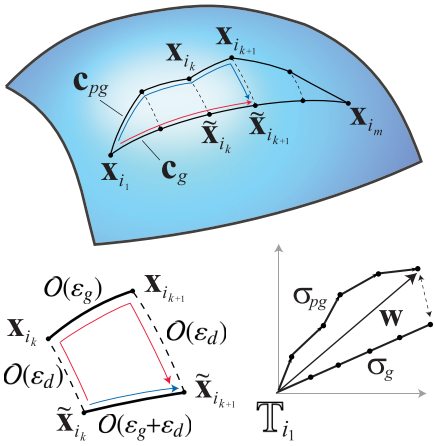
$$\mathbf{v}_{i_s} = \left(\prod_{j=1}^{r-1} \mathbf{R}_{i_j,i_{j+1}} \right) \left[\mathbb{T}_{i_r}^{\ t} \mathbf{e}_{i_s} \right] = \left(\prod_{j=1}^{r-1} \mathbf{R}_{i_j,i_{j+1}} \right) \left[\mathbb{T}_{i_r}^{\ t} \mathbf{g}_{i_s} \right] + O(\varepsilon_g^3).$$

Using Eq. (A.1) repeatedly, we obtain that $\mathbf{v}_{i_s}$ is approximately equal to $\mathbf{g}_{i_s}$ parallel-transported to $\mathbb{T}_{i_1}$ along a piecewise-geodesic curve $\mathbf{c}_{pg}$ passing through the points $(\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_m})$, i.e.

$$\mathbf{v}_{i_s} = \mathbb{T}_{i_1} d\varphi_{i_1} \left(\prod_{j=1}^{s-1} \mathbf{P}_{i_j,i_{j+1}} \right) \left[\mathbf{g}_{i_s} \right] + O\left(\sum_{j=1}^{s-1} \gamma_{i_j,i_{j+1}} \right) + O(\varepsilon_g^3).$$

Denoting the first term of the right-hand side by $\mathbf{w}_{i_s}$ and calling $\mathbf{w} = \sum_{s=1}^{m} \mathbf{w}_{i_s}$, we obtain

$$\sum_{s=1}^{m} \mathbf{v}_{i_s} = \sum_{s=1}^{m} \mathbf{w}_{i_s} + O(\gamma) \text{ , i.e., } \quad \mathbf{v} = \mathbf{w} + O(\gamma),$$



where $O(\gamma)$ takes value between $O(m^2(\varepsilon+\varepsilon_g^3))$ and $O(m^2(\varepsilon^3+\varepsilon_g^3))$ depending on how many polyline segments are close to the boundary; note that $\mathbf{w}$ is the vector connecting the endpoints of Cartan's development $\sigma_{pg} \in T_{i_1}\mathcal{M}$ of the *piecewise-geodesic* curve $\mathbf{c}_{pg}$ interpolating the polyline vertices (see inset, bottom right). To show that the curve σ_{pg} stays close to Cartan's development σ_g of the geodesic $\mathbf{c}_g$, recall that for a patch of $\mathcal{M}$ with diameter $O(\varepsilon_d)$, the change in direction of a vector parallel-transported along different paths within the patch is bounded by $O(\kappa_s\varepsilon_d^2)$ (this result relies on a bounded *curvature transformation*, which in turn follows from our bound on absolute value of *sectional curvature* from Assumption I). Using Assumption II, we can cut the geodesic $\mathbf{c}_g$ connecting $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_m}$ into curved segments with vertices $\tilde{\mathbf{x}}_{i_k}$, such that $d_g(\tilde{\mathbf{x}}_{i_k}, \mathbf{x}_{i_k}) = O(\varepsilon_d)$ for $k = 2, \ldots, m-1$ (we set $\tilde{\mathbf{x}}_{i_1} = \mathbf{x}_{i_1}$ and $\tilde{\mathbf{x}}_{i_m} = \mathbf{x}_{i_m}$ for notational consistency). Because the geodesic length of $\mathbf{x}_{i_{k+1}}\mathbf{x}_{i_k}$ is $O(\varepsilon_g)$, the diameter of a patch enclosed by four geodesic segments $\mathbf{x}_{i_k}\mathbf{x}_{i_{k+1}}$, $\tilde{\mathbf{x}}_{i_{k+1}}\tilde{\mathbf{x}}_{i_k}$, $\tilde{\mathbf{x}}_{i_k}\mathbf{x}_{i_k}$ and $\mathbf{x}_{i_{k+1}}\tilde{\mathbf{x}}_{i_{k+1}}$ is $O(\varepsilon_g+\varepsilon_d)=O(\varepsilon_d)$ assuming $\varepsilon_g \lesssim \varepsilon_d$ (see inset). Thus the result of parallel transporting a vector along 3 geodesic segments $\tilde{\mathbf{x}}_{i_k}\mathbf{x}_{i_k}$, $\mathbf{x}_{i_k}\mathbf{x}_{i_{k+1}}$ and $\mathbf{x}_{i_k}\tilde{\mathbf{x}}_{i_k}$ deviates in direction from parallel transporting the same vector along geodesic segment $\tilde{\mathbf{x}}_{i_k}\tilde{\mathbf{x}}_{i_{k+1}}$ by $O(\kappa_s\varepsilon_d^2)$. Summing up contributions from all the patches for $k = 1, \ldots, \ell$, and taking into account cancellations from segments $\mathbf{x}_{i_k}\tilde{\mathbf{x}}_{i_k}$ and $\tilde{\mathbf{x}}_{i_k}\mathbf{x}_{i_k}$ for $k = 2, \ldots, \ell-1$, the directional error in parallel transporting a vector along the piecewise geodesic curve $\mathbf{x}_{i_1}\mathbf{x}_{i_2}\ldots\mathbf{x}_{i_\ell}\tilde{\mathbf{x}}_{i_\ell}$ compared to its transport along the true geodesic segment $\mathbf{x}_{i_1}\tilde{\mathbf{x}}_{i_\ell}$ becomes $O(\ell\kappa_s\varepsilon_d^2)$. As a result, performing Cartan's development patch by patch, the final discrepancy between the endpoint positions of σ_{pg} and σ_g incurred by developing them using parallel transport along piecewise-geodesic curve $\mathbf{c}_{pg}$ vs. the true geodesic $\mathbf{c}_g$ is $O(m^2\kappa_s\varepsilon_g\varepsilon_d^2)$, as it combines the cumulative directional errors of parallel transported tangent vectors and the lengths of the corresponding segments.

Given that Cartan's development of a geodesic curve is a straight segment, we conclude that $||\mathbf{w}||_2 = d_g(\mathbf{x}_{i_1}, \mathbf{x}_{i_m}) + O(m^2\kappa_s\varepsilon_g\varepsilon_d^2)$. By construction, we have $d_{\mathrm{PTU}}(\mathbf{x}_{i_1}, \mathbf{x}_{i_m}) = ||\mathbf{v}||_2$, implying that

$$d_{\mathrm{PTU}}(\mathbf{x}_{i_1}, \mathbf{x}_{i_m}) = d_g(\mathbf{x}_{i_1}, \mathbf{x}_{i_m}) + O(\delta),$$

where $O(\delta) = O(m^2\varepsilon_g\varepsilon_d^2\kappa_s+\gamma)$. Note that our discrete unfolding of *any* polyline

converges to the corresponding Cartan's development; however, in general it has to be a (nearly-)shortest polyline for its extremities to be at a distance approximating the proper geodesic length, as geodesic curves are only locally (and not globally) shortest. □

Finally, we note that while our assumptions are weaker that the ones used to prove convergence of Dijkstra polylines to geodesic curves [180], the error for graph-based distance approximations has linear dependence on the number of polyline segments, while our bound depends quadratically on the number of segments. However, in practice our parallel transport based method consistently outperforms Dikstra path approximation (see Fig. 4.4), potentially pointing to the existence of a tighter bound.

Appendix B

PARALLEL TRANSPORT UNFOLDING: SUPPLEMENTAL MATERIAL

This appendix contains additional results to provide further tests of the Parallel Transport Unfolding (PTU) approach and comparisons to other existing manifold learning algorithms.

B.1 PTU vs. Local Methods

One of the key advantages of Parallel Transport Unfolding is its resilience to noise: like Isomap [180], PTU uses *all* geodesic distances to embed a dataset into a low-dimensional space, which allows for a greatly increased robustness to noise and outliers. In this section, we provide further numerical tests to confirm this statement.

Petals Dataset

First off, the noiseless Petals dataset makes clear that Isomap fails due to the obvious geodesic non-convexity while PTU has no problem finding the proper four petals, as demonstrated in Fig. B.1. In this noiseless case, it turns out that most local methods, such as Modified LLE [203], Hessian LLE [71], and SAKE, do also remarkably well (see Fig. B.2), as convexity (or lack thereof) plays basically no role in their embeddings.

However, if a bit of noise is added to this example, local methods fail (at times spectacularly): for a moderate Gaussian noise with a standard variation of 3% of the radius of the sphere on which the petals lie, Fig. B.3 shows that local methods all fail while PTU keeps a very similar embedding since it relies on all pairwise geodesic distances—Isomap too to a certain extent, even if it is clearly more deformed than in the noiseless case.

Study of Noise Effects on Local and Global Methods

In order to better demonstrate the robustness of global methods to noise, we use the simple (and very widely used) Swiss Roll dataset, and use both local and global methods on this dataset with an increasing amount of Gaussian noise along the normal of this roll (varying the standard deviation from 0 to 2.8% of the bounding box size). The number of neighbors is set to 10 for all methods to offer a fair comparison.

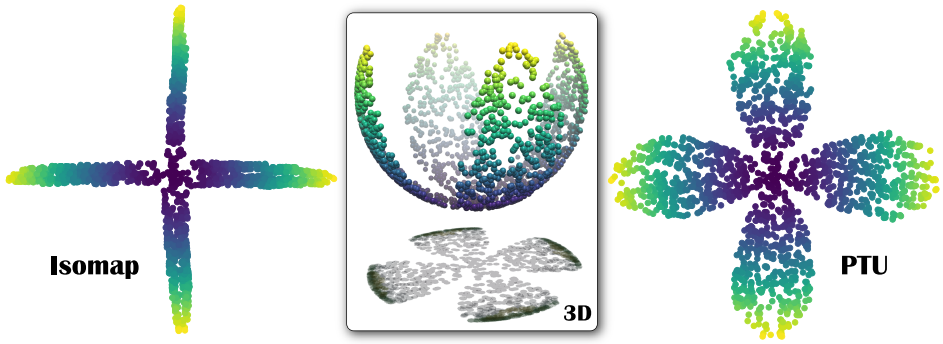


Figure B.1: **Unfolding Petals.** From a 3D sampling of 4-petal shaped portion of a sphere (middle), Isomap (a staple of manifold learning) fails to find a near isometric 2D parameterization (left) due to the non-convexity of the intrinsic geometry. Our parallel transport approach, instead, deals with this case perfectly (right). Lifting the pointset to 100D and applying random rotations and reflections does not change our result.

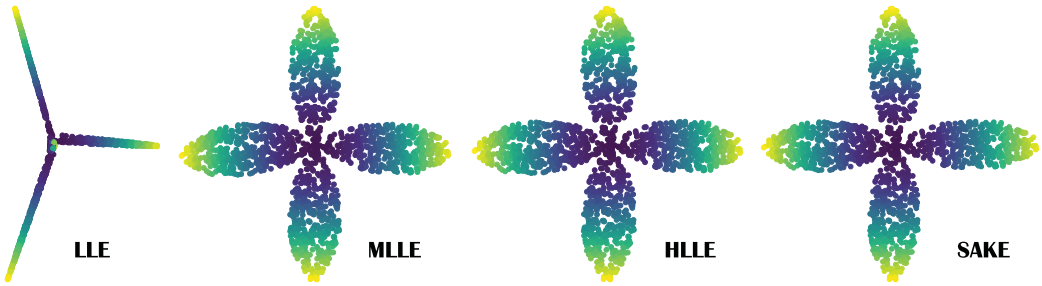


Figure B.2: **Local Methods for Noiseless Petals.** From a 3D sampling of 4-petal shaped portion of a sphere (see Fig. 4.1), local methods such as Modified LLE [203], Hessian LLE [71], or SAKE have no issue with the non-convexity of the intrinsic geometry (unlike Isomap), and give results nearly equivalent to PTU. A notable exception is LLE [155], which returns a near degenerate solution.

As Table B.4 clearly shows, the best local method (SAKE on this example, see Image SAKE/σ = 1.2%) starts failing at half of the maximum standard deviation that global methods can handle (see Images Isomap/σ = 2.2% and PTU/σ = 2.2%). When the standard deviation of the noise reaches 2.8%, even global methods fail as the proximity graph starts having numerous connections across branches: pairwise geodesic distances will have too many incorrect values to be able to recover a decent embedding. Note that at this level of noise (see side view of Swiss Roll), the dataset is far from the manifold assumption we are making about input data.

B.2 Landmark-PTU

Fig. B.5 provides more results for Landmark-PTU: in order to complement Fig. 17 from the submission, we also provide the results for landmarks on the *noiseless* Petals

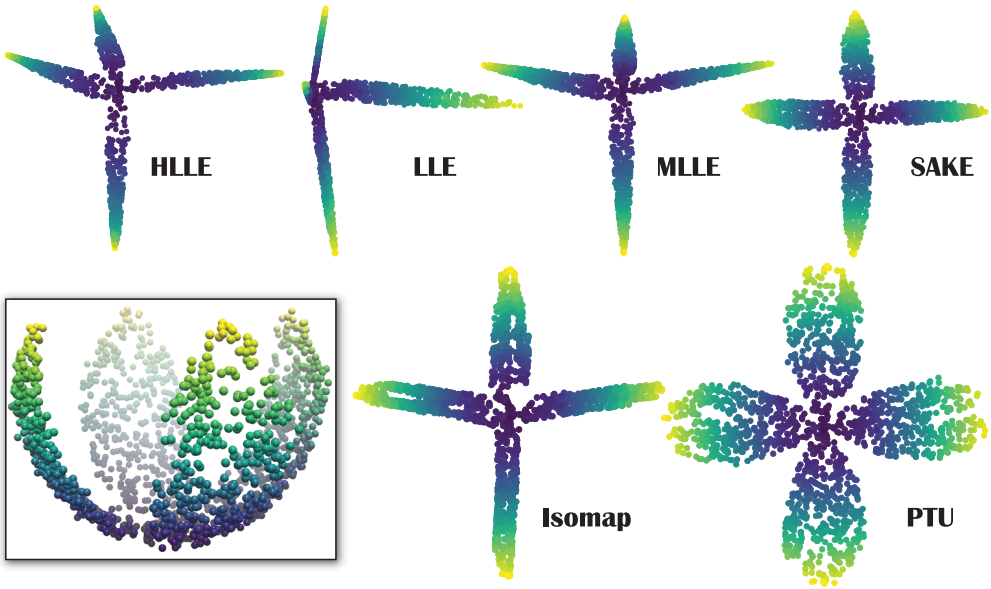


Figure B.3: **Noisy Petals.** Given a 3D sampling of a 4-petal shaped portion of a sphere (see Fig.B.1) with added Gaussian noise in the normal direction (σ: 3% of sphere radius), PTU recovers an almost perfect quasi-isometric 2D parametrization, while Isomap still fails (bottom). Local methods, not exploiting large geodesic distances, fail even worse, with the notable exception of SAKE that performs better than Isomap.

dataset, as well as on the highly-irregular S-shaped dataset from Fig. 11. Here again, less than 1% of landmarks is enough to capture the shape almost perfectly.

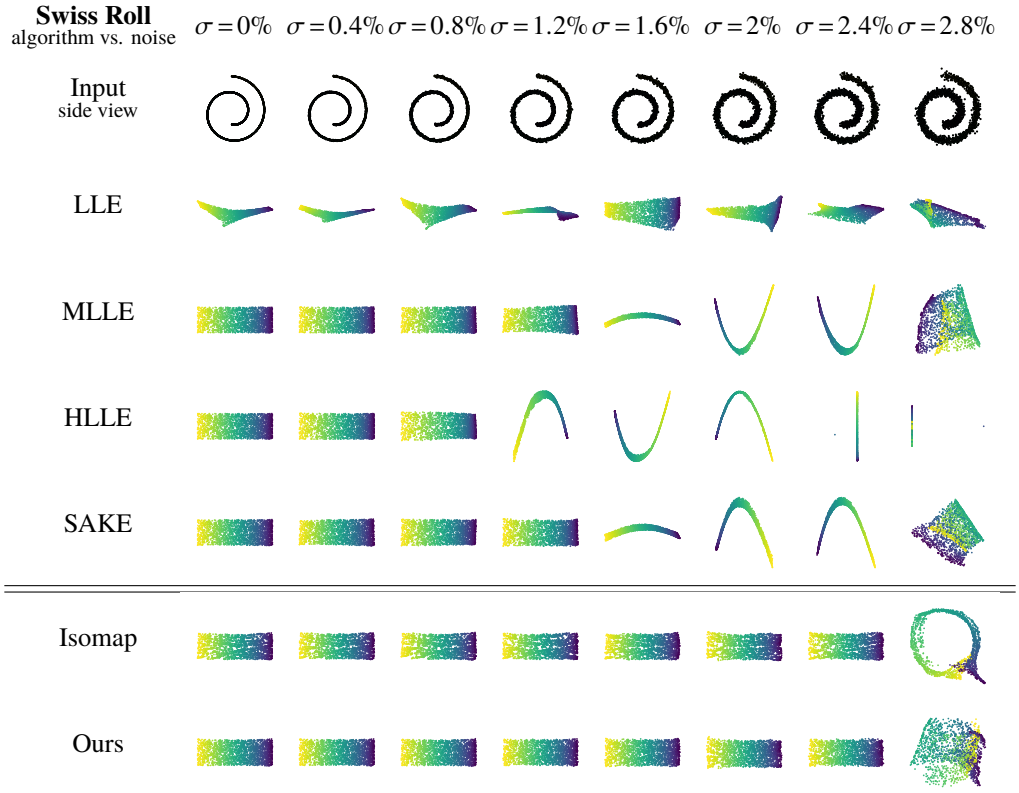


Figure B.4: **Effects of Noise on Local and Global Methods.** Using the Swiss Roll dataset, Gaussian noise with standard deviation given as a percentage of the bounding box of the original noiseless swiss roll is added along the normal. We use the same number of neighbors (10) for local methods to provide a fair comparison (it prevents shortcutting as much as possible; using larger values would make the local methods fail earlier). Local methods all failed around $\sigma = 1.3\%$, while global methods (Isomap and our approach) fare well until 2.7%. At 2.8%, the neighbors of a datapoint may belong to several different branches of the roll, which makes it impossible even for global methods to handle.

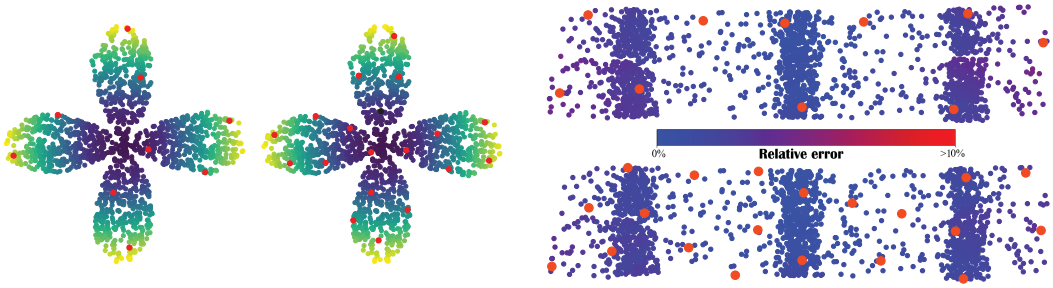


Figure B.5: **Landmark-PTU.** Landmarks are colored red for clarity. (Left) the use of 9 landmarks (left) or 19 landmarks (right) is enough to reconstruct the petals in the noiseless example of Fig. 1 of the submission. (Right) using 10 landmarks (top) vs. 20 landmarks (bottom) is visually very similar on this 2000-point datasets, although one can notice a slight distortion as indicated by the color of the mapped points using the color ramp based on distortion error compared to the expected perfect embedding.

Appendix C

VARIATIONAL DEFINITION OF OPERATOR-ADAPTED BASIS

We briefly summarize in this appendix the original motivation of the operator-adapted decomposition proposed in [140, 141]. Since it matches our construction in the case of scalar-valued basis functions, it is interesting to understand the functional approximation roots of their approach to contrast it with our finite element perspective.

Optimal Choice of Basis Functions. Consider the solution u to Eq. (5.1), and a finite set of measurements $m_i = \langle u, \varphi_i^k \rangle_{L^2}$ on a fixed level k obtained by integrating u against a set of test functions $\{\varphi_i^k\}_{i=1}^{n_k}$. In a context of functional approximation, a natural thing to ask is: what is the optimal selection of "adapted" basis functions φ_i^k, such that the approximation error between u and $\tilde{u}^k = \sum_{i=1}^{n_k} m_i \varphi_i^k$ is minimized, for any choice of u?

Game Theoretical Insight. This functional approximation problem can be formalized as a *zero-sum game* between two players as follows: player I chooses an arbitrary function $u^{\mathrm{I}} \in H$ from the solution space of Eq. (5.1); player II constructs an approximation u^{II} of u^{I} from partial information, only having access to the measurements $\{m_i\}_{i=1}^{n_k}$ of player I's function; the utility that player I aims to maximize and player II to minimize is given by

$$V(u^{\mathrm{I}}, u^{\mathrm{II}}) = \frac{||u^{\mathrm{I}} - u^{\mathrm{II}}||_{\mathcal{L}}}{||u^{\mathrm{I}}||_{\mathcal{L}}}. \tag{C.1}$$

This game has a solution in mixed strategies, with mixed optimal strategy for player I, and *pure* optimal strategy for player II [140]. In particular, optimal strategy for player I consists in drawing u^{I} at random from a weak Gaussian distribution ζ of covariance $\mathcal{L}^{-1}$, while a minimax strategy for player II reduces to computing deterministic function u^{II} as the conditional expectation of ζ given the measurements $\{m_i\}_{i=1}^{n_k}$.

Variational Formulation. The optimal basis functions $\{\varphi_i^k\}_{i=1}^{n_k}$ corresponding to the minimax strategy are, in fact, solutions of the following variational problem:

$$\varphi_i^k = \underset{\phi \in H}{\arg\min}\ ||\phi||_{\mathcal{L}}^2 \ \text{ s. t. } \langle \phi,\ \varphi_j^k \rangle_{L^2} = \delta_{ij} \text{ for } j = 1 \ldots n_k. \tag{C.2}$$

Using this formulation, [140] shows that since the test functions were chosen to be refinable, the optimal basis functions are refinable as well:

$$\varphi_i^k = \sum_{j=1}^{n_{k+1}} \mathbb{C}_{ij}^k \varphi_j^{k+1},$$

where entries of the refinement matrix $\mathbb{C}_{ij}^k$ correspond to conditional expectations of finer test measurements $\langle \varphi_j^{k+1}, \zeta \rangle_{L^2}$ given a single non-zero coarser level measurement $\langle \varphi_l^k, \zeta \rangle_{L^2} = \delta_{il}$ for $l = 1 \dots n_k$. To find the explicit expression of the refinement matrix, we can rewrite the problem (C.2) in matrix form, and using earlier notation $\mathbb{A}_{ij}^k := \langle \varphi_i^k, \varphi_k^k \rangle_{\mathcal{L}}$, we obtain

$$\mathbb{C}^k = \underset{\mathbb{X} \in \mathbb{R}^{n_k \times n_{k+1}}}{\arg\min} \quad \mathrm{Tr}\left[\mathbb{X} \mathbb{A}^{k+1} \mathbb{X}^T\right] \quad \text{s. t.} \quad \mathbb{X}\, \mathbf{C}^{k,T} = \mathbb{I}_{n_k}.$$

The solution to this constrained minimization is precisely Eq. (5.15), which demonstrates the equivalence of variational formulation of Eq. (C.1) to our axiomatic construction (Sec. 5.3) and implies that resulting operator-orthogonal basis functions are optimal in the game-theoretical functional-approximation sense of Eq. (C.1).

They are also optimal in the Galerkin sense on any resolution level k [140]: for any $u \in H$ and its approximation $u^k = \sum_{i=1}^{n_k} m_i \varphi_i^k$, we have

$$\|u - u^k\|_{\mathcal{L}} = \inf_{v \in \mathcal{V}^k} \|u - v\|_{\mathcal{L}}.$$

In other words, u^k is the $\mathcal{L}$-orthogonal projection of u onto $\mathcal{V}^k$.

Appendix D

PROPERTIES OF THE BOTTOM-UP WAVELET CONSTRUCTION

In this first appendix, we briefly review the properties of the (scalar-valued) gamblets construction derived in [139], as they apply to our extension as well. Readers are referred to [139] for rigourous proofs: we only provide a summary of the properties, along with the key conditions required.

D.1 Bounded Condition Numbers

Let $||\cdot||_{\mathcal{L}^{-1}}$ be the norm of H^* defined as the dual to the energy norm $||\cdot||_{\mathcal{L}}$. The condition numbers of stiffness matrices $\mathbb{B}^k$ and $\mathbb{A}^1$ are uniformly bounded, provided the spans of the test functions $\{\mathcal{V}^k\}_{k=1}^q$ are regular and weakly aligned with the eigensubpaces of $\mathcal{L}^{-1}$ in the following sense: there exists a constant c_1, such that

- the coarsest test functions can capture the eigenspaces of $\mathcal{L}^{-1}$, i.e.,

$$\sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^{n_1};|\mathbf{x}|=|\mathbf{y}|=1} \frac{\| \sum_{i=1}^{n_1} \mathbf{x}_i \varphi_i^1 \|_{\mathcal{L}^{-1}}}{\| \sum_{i=1}^{n_1} \mathbf{y}_i \varphi_i^1 \|_{\mathcal{L}^{-1}}} \leq c_1;$$

- the condition numbers of the wavelet refinement matrices are bounded for $k = 1, \ldots, q-1$:

$$\mathrm{cond}\left(\mathbf{W}^k \mathbf{W}^{k,T}\right) \leq c_1;$$

- the refinement matrices of test functions are bounded for $k = 1, \ldots, q-1$:

$$\|\mathbf{C}^k\|_2 \leq c_1;$$

- eigensubspaces of $\mathcal{L}^{-1}$ and test functions must be weakly aligned for $k = 2, ..., q$, i.e.,

$$\sup_{x\in\mathrm{Ker}\,\mathbf{C}^{k-1},|x|=1} \inf_{y\in\mathbb{R}^{n_{k-1}}} \frac{\left\| \sum_{i=1}^{n_k} x_i \varphi_i^k - \sum_{j=1}^{n_{k-1}} y_j \varphi_j^{k-1} \right\|_{\mathcal{L}^{-1}}^2}{\inf_{z\in\mathbb{R}^{n_k},|z|=1} \left\| \sum_{i=1}^{n_k} z_i \varphi_i^k \right\|_{\mathcal{L}^{-1}}^2} \leq c_1.$$

This inequality can also be interpreted as a bound on the relative gap between information that is lost vs. the one that is propagated during transition from fine

to coarse levels. Note that slightly stronger versions of all these conditions can provide bounds for the *minimum* and *maximum* eigenvalues of matrices $\mathbb{A}^k$ and $\mathbb{B}^k$, revealing that the eigenranges of $\mathbb{B}^k$ correspond to k frequency subbands of the original stiffness matrix $\mathbb{A}^q$ corresponding to the finest level [139, 140].

D.2 Exponential Decay

The fast decay of our operator-adapted basis functions on a given resolution level k follows from the localization properties of both the underlying operator and of the test functions. For each $i = 1..n^k$, define $\tau_i^k \in \Omega$ to be a small convex region including the support of the test function φ_i^k, containing a ball of radius h_k, and being contained in a ball of radius δh_k for some constants $\delta, h_k \in [0, 1]$. Let Ω_i^k be a small neighborhood of τ_i^k, such that $\Omega = \cup_{i=1}^{n_k} \Omega_i^k$, and the distance between τ_i^k and complement of Ω_i^k is between δh and h. Denoting the standard Sobolev norm (with appropriate boundary conditions) on $H(\Omega)$ by $||\cdot||_{H(\Omega)}$ and its dual norm on $H^*(\Omega)$ by $||\cdot||_{H^*(\Omega)}$, the key (sufficient) condition required for exponential decay of operator-adapted basis is the existence of two constants $0 < c_{\min} \leq c_{\max} < \infty$ that satisfy the following frame inequality for all $v \in H^*(\Omega)$:

$$c_{min} \inf_{\varphi \in \mathcal{V}^k} ||v - \varphi||^2_{H^*(\Omega)} \leq \sum_{i=1}^{n_k} \inf_{\varphi \in \mathcal{V}^k} ||v - \varphi||^2_{H^*(\Omega_i^k)} \leq c_{max} \inf_{\varphi \in \mathcal{V}^k} ||v - \varphi||^2_{H^*(\Omega)}.$$

The exponent of the decay rate depends on $c_{\min}$, $c_{\max}$ and the locality of the operator $\mathcal{L}$. This condition, describing the localization of test functions in the dual norm, is in fact implied by three simple and natural inequalities (for some finite constant $c_2 > 0$):

- *Poincaré inequalities*: any $\varphi^\perp$ from the L^2-complement of test functions $\mathcal{V}^{k,\perp} = \{v \in H : \langle v, \varphi_i^k \rangle_{L^2} = 0 \, \forall \varphi_i^k \in \mathcal{V}^k\}$ has bounded derivatives:

$$\left\| D^t \varphi^\perp \right\|_{L^2(\Omega)} \leq c_2 \, h_k^{s-t} \|\varphi^\perp\|_{H_0^s(\Omega)} \qquad \forall t \in \{0, \ldots, s\};$$

- *Frame inequalities (boundedness of test functions)*:

$$\sum_{i=1}^{n_k} \langle \varphi_i^k, f \rangle^2_{L^2} \leq c_2 \sum_{t=0}^{s} h_k^{2t} \|D^t f\|_{L^2(\Omega)} \qquad \forall f \in H_0^s(\Omega);$$

- *Inverse Poincaré inequalities*:

$$h_k^{2s} \leq c_2 \|\varphi_i^k\|^2_{H^*(\tau_i)} \qquad \forall i \in \{1, ..., n_k\}.$$

Note that these conditions, obtained in [138, 139], provide a generalization of [113, 124, 140].

D.3 Homogenization Property

The operator-adapted basis functions on coarser levels carry over information from finer scales, nicely approximating the operator eigensubspaces of corresponding frequencies. Indeed, their resulting shape better captures the operator on a given resolution level than generic polynomial finite-element basis functions, with a bound proven analytically in [139] for several cases. More concretely, the solution u to Eq. (5.1) and its operator-adapted level-k FEM approximation u^k (Eq. (5.19)), assembled using k coarsest resolution levels, satisfy the following inequality for some constant $c > 0$:

$$\frac{||u - u^k||_{\mathcal{L}}}{||g||_{L^2}} \leq c h_k^s, \tag{D.1}$$

where the right-hand side is independent of $\mathcal{L}$ and h_k characterizes the radius of support of a test functions of level k. This last inequality implies that most of the "energy" concentrates on coarser resolutions and serves as one of the key reasons behind the homogenization properties of operator-adapted multiresolution analysis and its ability to use only a few discretization levels to get good approximate solutions: it guarantees that omitting (some or all) information from higher resolution levels does not degrade accuracy too much.