

# Multiscale Representations for Manifold-Valued Data

Inam Ur Rahman\*, Iddo Drori\*, Victoria C. Stodden\*  
David L. Donoho\*, Peter Schroeder†

January 20, 2005

## Abstract

We describe multiscale representations for data observed on equispaced grids and taking values in manifolds such as: the sphere  $S^2$ , the special orthogonal group  $SO(3)$ , the positive definite matrices  $SPD(n)$ , and the Grassmann manifolds  $G(n, k)$ . The representations are based on the deployment of Deslauriers-Dubuc and Average-Interpolating pyramids ‘in the tangent plane’ of such manifolds, using the *Exp* and *Log* maps of those manifolds. The representations provide ‘wavelet coefficients’ which can be thresholded, quantized, and scaled much as traditional wavelet coefficients. Tasks such as compression, noise removal, contrast enhancement, and stochastic simulation are facilitated by this representation. The approach applies to general manifolds, but is particularly suited to the manifolds we consider, i.e. Riemannian symmetric spaces, such as  $S^{n-1}$ ,  $SO(n)$ ,  $G(n, k)$ , where the *Exp* and *Log* maps are effectively computable. Applications to manifold-valued data sources of a geometric nature (motion, orientation, diffusion) seem particularly immediate. A software toolbox, SymmLab, can reproduce the results discussed in this paper.

**Keywords:** Symmetric Space, Lie Group, Two-Scale Refinement Scheme, Wavelets, Compression, Denoising, Nonlinear Refinement Scheme.

**Acknowledgements.** Partial support from NSF DMS 00-77261, and 01-40698 (FRG) and ONR-MURI. We thank Roberto Altschul and Sabyasachi Basu from Boeing for data on aircraft orientations. We thank Robert Dougherty and Brian Wandell from Stanford for the magnetic resonance data (R01 EY015000). DLD would like to thank Leo Guibas and Claire Tomlin of Stanford University for helpful discussions, as well as Nira Dyn of Tel-Aviv University, John F. Hughes of Brown University and Jonathan Kaplan of Harvard University.

\*: Statistics, Stanford University

†: Computer Science, CalTech

## 1 Introduction

### 1.1 The Challenge

The current ‘data deluge’ inundating science and technology is remarkable not merely for the often-mentioned *volumes* of data, but also for the rapid proliferation in new data *types*. In addition to the old standby of simple numerical arrays, we are starting to see arrays where the entries have highly structured values obeying nonlinear constraints.

Many such examples can be given. We have in mind data arrays of the form  $p(t)$ ,  $p(x, y)$ , or  $p(x, y, z)$  where  $t$ ,  $x$ ,  $y$ ,  $z$  run through equispaced values in a cartesian grid, and  $p$  takes values in a manifold  $M$ . Consider these examples:

- *Headings.* Here  $p$  specifies directions in  $\mathbf{R}^2$  or  $\mathbf{R}^3$ , and so  $M$  is either the unit circle  $S^1 \subset \mathbf{R}^2$  or the unit sphere  $S^2 \subset \mathbf{R}^3$ . Such data can arise as a time series of observations of vehicle headings.
- *Orientations.* Here  $p$  gives ‘tripods’, i.e. orientations belonging to  $M = SO(3)$ . Such data can arise as a time series of aircraft orientations (pitch, roll, yaw).
- *Rigid Motions.* Here  $p$  specifies rigid motions in the special Euclidean group  $M = SE(3)$ . Such data can arise as a time series of placements of an object in space (position, orientation), or as a spatially-organized array giving the displacements and orientations of marker particles having undergone a deformation.
- *Deformation Tensors.* Here  $p$  is a symmetric positive definite matrix in  $M = SPD(n)$ . Spatially-organized data of this kind can arise from measurements of strain/stress and deformation in materials science and earth science. Arrays of this kind also arise in cosmological measurements of gravitational lensing.
- *Distance Matrices.* Here each  $p$  is an  $n$  by  $n$  matrix giving the pairwise distances between all pairs in a cloud of  $n$  points. Time series of this kind can arise as representing the state of a swarm of maneuvering vehicles, each of which can sense its distance to all other members of the swarm.
- *Projections, Subspaces.* Here  $p$  is a projector with  $k$ -dimensional range, or what is the same thing, a  $k$ -subspace of  $\mathbf{R}^n$ . Such values belong to the Grassman manifold  $G(k, n)$ . Time series of this kind can arise in array signal processing, where the subspace is associated with the signal-generating sources.

The proliferation of novel data types presents a challenge: find data representations which are sufficiently general to apply to many data types and yet respect the manifold structure.

## 1.2 Our Approach

In this article, our goal is to generalize wavelet analysis from the traditional setting of functions indexed by time or space and taking real values to the case where functions, still indexed by time or space, take values in a more general manifold  $M$ . We discuss two basic approaches: (a) interpolatory schemes and (b) midpoint interpolatory schemes. Each gives a meaningful generalization of wavelet analysis to the manifold case; the one second is most appropriate for ‘noisy’ data. However, strictly speaking, this is not a wavelet analysis in the traditional sense. Our “wavelet” transform generates arrays which are organized like wavelet coefficients, stratified by location and scale, but the values in the array are more complex. It might also be called a pyramid transform; however, that terminology is more typically used for transforms which are overcomplete (i.e.  $n$  data points result in more than  $n$  coefficients), whereas the transforms we describe take  $n$  values and produce  $n$  “coefficients”.

The approach we discuss requires the computation of differential-geometric ‘Exp’ and ‘Log’ maps associated with the manifold  $M$ ; for some spaces this task will be easier than for others. We focus here on the *Riemannian symmetric spaces*, where the notion of midpoint is well-defined, and we have implemented our proposals extensively in a wide range of example spaces. The spaces we have studied all involve in some way the general linear group  $GL(n)$ .

- *Subgroups of  $GL(n)$ .* We treat the special orthogonal group and special Euclidean group which allow us to handle orientations and rigid motions.

- *Quotients of  $GL(n)$ .* We treat various quotient spaces of  $GL(n)$ , including Grassmann manifolds, Stiefel manifolds, and the special case of Spheres. These allow us to handle subspaces of  $\mathbf{R}^n$  and to handle headings (direction fields) in  $\mathbf{R}^n$ .
- *Jordan Algebras.* We treat the manifold of Symmetric Positive Definite Matrices  $SPD(n)$ , allowing us to handle deformation tensors and diffusion tensors.

All our examples are in some way representable using the general linear group  $GL(n)$  and the differential-geometric *Exp* and *Log* maps involve, as we show, astute use of the matrix exponential and matrix logarithm. Taking matrix logarithm and matrix exponential as given efficiently computable primitives, our algorithms are computationally practical, giving order  $O(n)$  algorithms for wavelet analysis and reconstruction.

### 1.3 Our Contributions

In addition to developing a general approach to multiscale analysis which works for many specific manifolds, we consider three key application areas:

- *Compression:* Approximately representing an  $M$ -valued dataset using few bits.
- *Noise Analysis/Removal:* Understanding ‘noise’ in  $M$ -valued data, representing its properties and separating noise from signal.
- *Feature Extraction:* Representing specific properties of an  $M$ -valued dataset in terms more amenable to pattern recognition.

Our approach provides a unified viewpoint for these practical tasks, completely paralleling the wavelet approaches which have proven so successful with real-valued data, and are easily implemented in software. In effect, in our examples, real-valued wavelet coefficients are replaced by matrix-valued coefficients.

At a higher level we contribute following:

1. *Practical Tools.* We have developed a toolbox of Matlab m-files able to handle time series and spatial arrays of  $M$ -valued data. Called **SymmLab**, it is patterned after the earlier Matlab toolboxes **WaveLab** [3] for wavelet analysis and **BeamLab** [5] for beamlet analysis. See the companion article [1] and the website at <http://www-stat.stanford.edu/~symmlab/>.
2. *Awareness of  $M$ -valued data.* We make an effort to call attention to the wide range of application areas in which manifold-valued data are now being gathered.
3. *Understanding.* In generalizing “wavelet” analysis from  $\mathbf{R}$ -valued to  $M$ -valued data, certain new concepts arise which were not evident in the  $\mathbf{R}$  case; an example is the fact that wavelet coefficients must live in the tangent space while coarse-scale coefficients of ‘father wavelets’ live in the manifold.
4. *Inertia & Compression.* The constructions we describe make geodesic motions highly compressible. Such motions correspond to  $p(t)$  evolving in time on the manifold without external forces; in other words, inertial motions in  $M$ . Hence, in our representation, inertial motions are highly compressible, which has advantages for systems which often operate inertially.

5. *Open Problems.* In effect we define a class of nonlinear refinement schemes: interpolating and average interpolating refinement schemes in the tangent space. It is empirically quite clear that these nonlinear schemes have the same smoothness properties as their linear counterparts. Proofs for the smoothness of the limits of those schemes remain to be found. Perhaps this article will mobilize some interest and efforts in the direction of a solution for these nonlinear refinement schemes.

## 1.4 Relation to Other Work

The work described here has been underway for five years, and was presented at a meeting on Constructive Approximation in Charleston South Carolina, May 2001 [14], and at the meeting on Curves and Surfaces in Saint Malo, June 2002 [15]. We are aware of several groups working on refinement schemes for manifold data working independently of us, and will attempt to mention them in Section 8.2. The delay in publication of our work creates the possibility that we have not cited all relevant work; we will correct such oversights in future revisions of the manuscript.

## 2 Classical Multiscale Transforms

We now quickly review some basic notations and constructions associated with multiscale representations for real-valued data, and with the properties of such constructions. We focus on two kinds of refinement schemes and the transforms they generate.

### 2.1 Interpolating Approach

Each approach to multiscale representation that we describe has three ingredients: a multiscale pyramid summarizing a function  $f : \mathbf{R} \mapsto \mathbf{R}$  across scales and locations, a refinement scheme, showing how to impute fine-scale behavior from coarse-scale behavior, and a wavelet analysis scheme, combining the first two elements.

#### 2.1.1 Pyramid of Point Evaluations

Let  $t_{j,k} = k/2^j$  denote the  $k$ -th dyadic point at scale  $j$ . This collection of dyadic rationals is dense in  $\mathbf{R}$ , and each collection  $(t_{j,k})_k$  at one fixed scale makes a grid  $\mathbf{Z}/2^j$ . Because of the nesting  $\mathbf{Z}/2^j \subset \mathbf{Z}/2^{j+1}$  this set of points is redundant; indeed

$$t_{j,k} = t_{j+1,2k}, \quad j, k \in \mathbf{Z};$$

the ‘novel’ points in  $(t_{j+1,k})_k$  not already present in  $(t_{j,k})_k$  come at the  $t_{j+1,2k+1}$  which fall halfway between the points in the grid  $\mathbf{Z}/2^j$ .

Suppose now we are given a uniformly continuous function  $f : \mathbf{R} \mapsto \mathbf{R}$ . It is determined by its values at the dyadic rationals, which can be organized into the array of point values

$$\beta_{j,k} = f(t_{j,k}), \quad j \geq 0, k \in \mathbf{Z}.$$

This collection of values provides a multiscale pyramid – one which we will later improve. It obeys the two-scale relation

$$\beta_{j,k} = \beta_{j+1,2k} \quad j, k \in \mathbf{Z};$$

at scale  $j + 1$ , the novel information in  $(\beta_{j+1,k})_k$  not already present in scale  $j$ ’s array  $(\beta_{j,k})_k$  is contained in the midpoint values  $\beta_{j+1,2k+1}$ . However, if  $f$  has any smoothness,  $\beta_{j+1,2k+1}$  will

typically be close to what one would expect from the coarse-scale values. Later we define wavelet coefficients measuring the deviation between  $\beta_{j+1,2k+1}$  and the ‘expected’ value. First we define a notion of what to ‘expect’ at midpoints based on refinement schemes.

### 2.1.2 Interpolating Refinement Schemes

The Deslauriers-Dubuc refinement scheme [10, 11, 17] works as follows. Starting from real-valued data  $f(k)$  available at the integers  $k$ , it interpolates values at all the dyadic rationals  $k/2^j$  by successive refinement through a series of stages. At the first stage, the original data,  $f(k)$ , are used to impute values at the half-integers  $\tilde{f}(k/2)$  by a *two-scale refinement* scheme. Afterwards values are available at all integers and half-integers. At the second stage, the same two-scale refinement scheme is applied to those values, yielding values at the quarter integers. And so on. In this way, one can fill in values at all binary rationals.

At the center of the process is the Deslauriers-Dubuc two-scale refinement scheme. Let  $D$  be an odd integer, for example 3.

In discussing two-scale refinement, we speak of the coarser scale – where values are already known, and a finer scale – where they will be imputed. The coarser scale at the  $j$ -th stage consists of integer multiples  $t_{j,k} = k/2^j$  of a dyadic fraction  $2^{-j}$  and the finer scale consists of integer multiples  $t_{j+1,k}$  of the next smaller fraction  $2^{-j+1}$ . To obtain the imputed values at the fine scale, we recall that the points  $k/2^j$  belonging to the coarse-scale grid also belong to the fine scale grid  $(2k)/2^{j+1}$ , so the imputations are immediate:

$$\tilde{f}((2k)/2^{j+1}) = f(k/2^j).$$

To get values at odd multiples of  $2^{-j-1}$ , say  $2k+1$ , we apply a simple local rule. We collect the  $D+1$  values located at the  $D+1$  closest coarse-scale sites to the fine grid location of interest. We fit a polynomial  $\pi_{j,k}(t)$  interpolating those values:

$$\pi_{j,k}(k'/2^j) = \tilde{f}(k'/2^j), \quad |k' - k| < (D+1)/2;$$

(the polynomial is unique). We then evaluate the fitted polynomial in the midpoint of interest, getting the imputed value

$$\tilde{f}((k+1/2)/2^j) \equiv \pi_{j,k}((k+1/2)/2^j).$$

The process of fitting a polynomial and imputing a value is illustrated in Figure 2.1.

Results from applying this rule through several stages appear in Figure 2.2, in which a Kronecker sequence is refined, turning a coarsely-sampled ‘spike’ into a (visually) smooth ‘bump’.

This full multiscale process results in values defined at all the binary rationals; in fact these values are uniformly continuous and have a unique continuous extension to the reals. This extension is not merely continuous, but Hölder regular of order  $R$  where  $R = R(D)$ . See Section 2.3.1 below.

### 2.1.3 Interpolatory Wavelet Transform

Given an interpolatory refinement scheme, we can build a wavelet transform [11]. The two-scale refinement scheme just discussed furnishes us with an operator which, starting from values

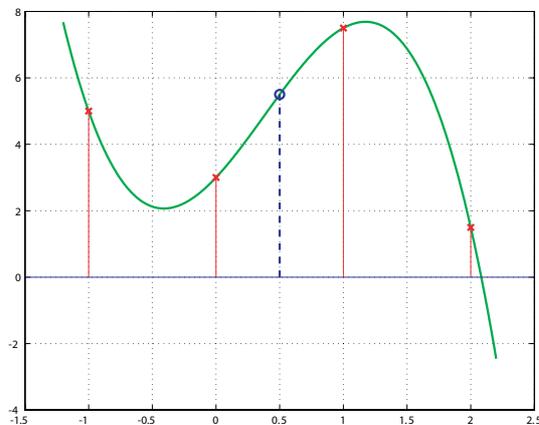


Figure 2.1: Deslauriers-Dubuc Interpolation at a single scale and location. Values at four sites  $k = -1, 0, 1, 2$ ; a cubic polynomial  $\pi_{0,0}$  interpolating the values, and the imputed value at  $1/2$ .

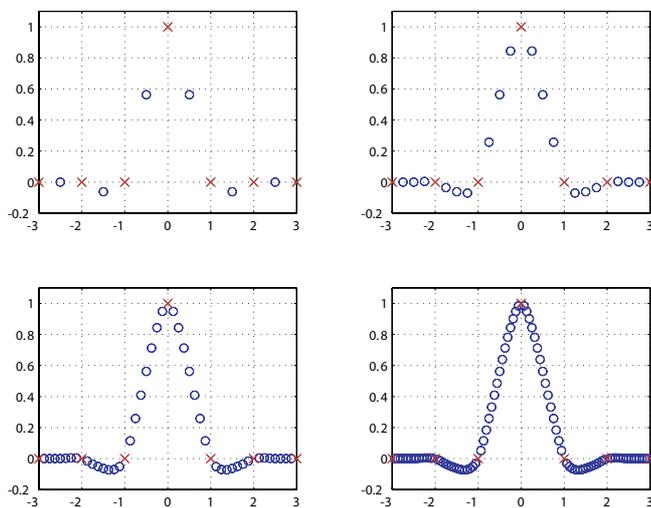


Figure 2.2: Successive Applications of Two-Scale Deslauriers-Dubuc Refinement, starting from a Kronecker sequence at the integers (red x's). Convergence is visually evident.

$\beta_{j,k} = f(k/2^j)$ ,  $k \in \mathbf{Z}$ , imputes values at midpoints  $\tilde{\beta}_{j+1,2k+1} = \tilde{f}((k+1/2)/2^j)$ ; defining wavelet coefficients by midpoint deflection

$$\begin{aligned}\alpha_{j,k} &= 2^{j/2}(f(k+1/2)/2^j) - \tilde{f}((k+1/2)/2^j), \quad k \in \mathbf{Z}, j \geq 0. \\ &= 2^{j/2}(\beta_{j+1,2k+1} - \tilde{\beta}_{j+1,2k+1}), \quad k \in \mathbf{Z}, j \geq 0\end{aligned}$$

We are explicitly measuring the difference between the observed and imputed values at midpoints  $(k+1/2)/2^j$  halfway between coarse-scale gridpoints  $k/2^j$  and  $(k+1)/2^j$ . In addition to the fine-scale information at  $j \geq 0$ , we need the coarse-scale information

$$\beta_{0,k} = f(k), \quad k \in \mathbf{Z}.$$

Taken together, the information in the coarse-scale samples  $(\beta_{0,k})_k$ , and in the wavelet coefficients  $((\alpha_{j,k})_{k \in \mathbf{Z}})_{j \geq 0}$  allows us to reconstruct any continuous  $f$ . Indeed the information in  $(\beta_{0,k})_k$  already provides the coarse-scale samples  $f(k)$ . We then use the samples  $(f(k))_k$  and apply two-scale refinement, getting imputations  $\tilde{f}(k+1/2)$ ; then rewriting

$$f(k+1/2) = \tilde{f}(k+1/2) + (f(k+1/2) - \tilde{f}(k+1/2)),$$

we see that

$$\beta_{1,2k+1} = \tilde{\beta}_{1,2k+1} + \alpha_{j,k}/2^{j/2};$$

and, of course,

$$\beta_{1,2k} = \beta_{0,k}, \quad k \in \mathbf{Z},$$

so that the  $(\alpha_{0,k})_k$  and  $(\beta_{0,k})_k$  together allow us to reconstruct  $(\beta_{1,k})_k$ . Continuing in this way, we reconstruct  $((\beta_{j,k})_k)_{j \geq 0}$ , i.e. we get  $f$  at all dyadic rationals.

## 2.2 Average-Interpolating Approach

We now turn to schemes based on averages rather than point values.

### 2.2.1 Pyramid of Block-Averages

Define the dyadic intervals  $I_{j,k} = [k/2^j, (k+1)/2^j)$ ,  $j, k \in \mathbf{Z}$ , again  $j$  is the scale parameter and  $k$  is a location index. The intervals at a single scale partition the line, and there is the two-scale refinement relation

$$I_{j,k} = I_{j+1,2k} \cup I_{j+1,2k+1}.$$

Consider now, for a given integrable function  $f$  the pyramid of values

$$\beta_{j,k} = \text{Ave}\{f|I_{j,k}\}, \quad j, k \in \mathbf{Z}.$$

This gives averages over intervals spanning a range of dyadic scales and locations, and such information characterizes the function  $f$ . Noting that

$$\text{Ave}\{f|[0, 1]\} = (\text{Ave}\{f|[0, 1/2]\} + \text{Ave}\{f|[1/2, 1]\})/2,$$

we see that the pyramid is redundant, obeying the two-scale relation

$$\beta_{j,k} = (\beta_{j+1,2k} + \beta_{j+1,2k+1})/2.$$

Moreover, if  $f$  has any smoothness then  $\beta_{j+1,2k+1}$  is expected to be close to  $\beta_{j,k}$ ; we now develop refinement schemes to predict fine-scale behavior from coarse-scale coefficients and a wavelet transform to remove the redundancy.

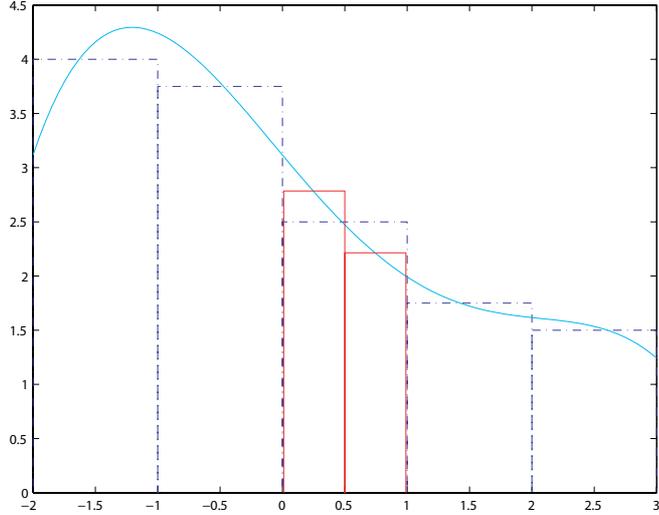


Figure 2.3: Average Interpolation at a single scale and location. Averages at five adjacent intervals  $I_{0,k}$ ,  $k = -1, 0, 1, 2$ , a quartic polynomial  $\pi_{0,0}$  interpolating the values, and the imputed averages at  $I_{1,0}$  and  $I_{1,1}$ .

## 2.2.2 Average-Interpolating Refinement Schemes

Average-Interpolating (AI) Refinement works in a fashion paralleling the Deslauriers-Dubuc scheme, but is based on local averaging over dyadic intervals  $I_{j,k}$  as opposed to point sampling at dyadic points  $t_{j,k}$  [12, 20]. It starts from averages  $\beta_{0,k}$  over intervals of unit length. It then generates data  $\tilde{\beta}_{j,k}$  at finer scales by successive refinement through a series of stages. At the first stage, averages  $\tilde{\beta}_{1,k}$  are imputed for dyadic intervals of length  $1/2$  by the following device. We fix  $D$  as an even integer, for example 4. We then, for each interval  $I_{0,k}$ , collect the  $D+1$  coarse-scale values at integral sites  $k'$  closest to  $k$ , and fit a polynomial  $\pi_{0,k}(t)$  *average*-interpolating those values:

$$Ave\{\pi_{0,k}(t) : t \in I_{0,k}\} = \beta_{0,k}, \quad |k' - k| < (D+1)/2.$$

We then impute by evaluating the averages of this polynomial over finer scale intervals.

$$\tilde{b}_{1,2k+\ell} \equiv Ave\{\pi_{0,k}(t) : t \in I_{1,2k+\ell}\}, \quad \ell = 0, 1.$$

See Figure 2.3.

Given imputed averages over dyadic intervals of length  $1/2$ , we now treat all those averages as given data at a newly-defined coarse scale and repeat the above two-scale refinement to impute averages at the dyadic intervals of length  $1/4$ ; we next impute averages over intervals of length  $1/8$ , and so on. The process is illustrated in Figure 2.4.

The full multiscale process results in averages defined at all the dyadic intervals; in fact there is a unique continuous function  $\tilde{f}$  consistent with those averages. In fact  $\tilde{f}$  is not merely continuous, but Hölder regular of order  $R$  where  $R = R(D)$ ; [12]. See Section 2.3.1.

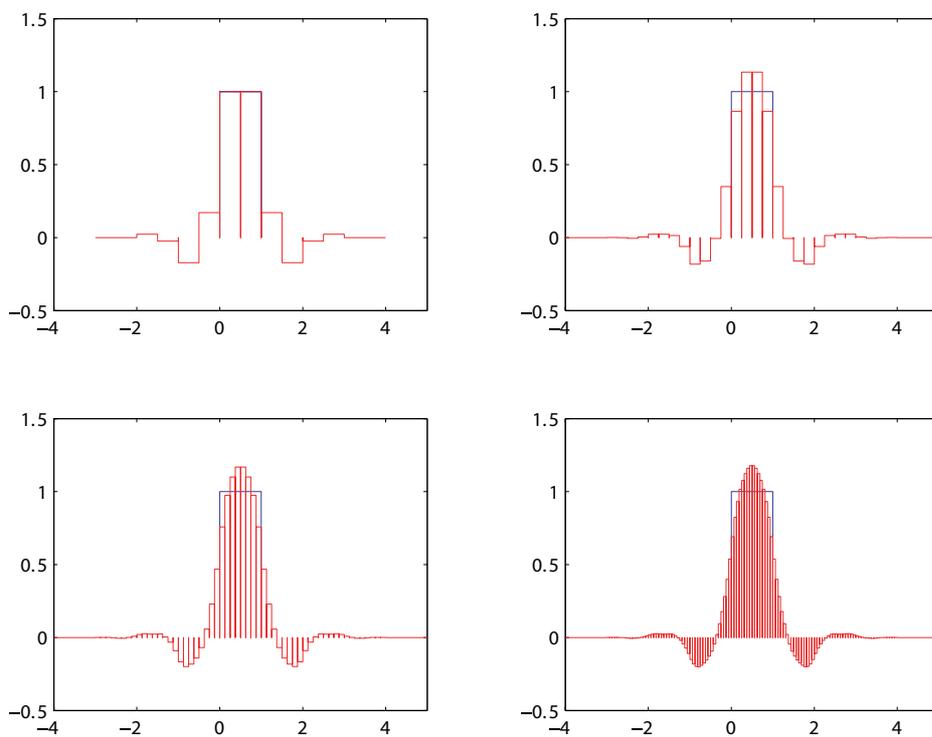


Figure 2.4: Successive Applications of Two-Scale Average-Interpolating Refinement, starting from a Kronecker sequence at the integers. Convergence is visually evident.

### 2.2.3 Average-Interpolating Wavelet Transform

The average-interpolating refinement scheme also leads to a wavelet transform; [12]. The two-scale refinement operator gives a way to pass from coarse-scale averages  $(\beta_{j,k})_k$  to imputed fine-scale averages  $(\tilde{\beta}_{j+1,k})_k$ . We then define wavelet coefficients

$$\alpha_{j,k} = 2^{j/2}(\beta_{j+1,2k+1} - \tilde{\beta}_{j+1,2k+1}), \quad k \in \mathbf{Z}, \quad j \geq 0.$$

These measure the deviation between the behavior of fine-scale averages and the anticipated behavior imputed from coarse-scales.

Equipped with  $(\beta_{0,k})_k$  and  $((\alpha_{j,k})_{k \in \mathbf{Z}})_{j \geq 0}$ , we can reconstruct  $f$  by a pyramid process. We start with the coarse-scale averages  $(\beta_{0,k})$  and coarse-scale wavelet coefficients  $(\alpha_{0,k})$  and combine them to produce averages

$$\beta_{1,k} = Ave\{f|I_{1,k}\}, \quad k \in \mathbf{Z}.$$

Indeed, we simply apply the two-scale refinement operator to the  $(\beta_{0,k})$ , obtaining imputed averages  $(\tilde{\beta}_{1,k})_k$  and then set

$$\beta_{1,2k+1} = \tilde{\beta}_{1,2k+1} + 2^{-j/2}\alpha_{0,k};$$

we also set

$$\beta_{1,2k} = (2\beta_{0,k} - \beta_{1,2k+1}).$$

This enforces the coarse/fine consistency constraint

$$\beta_{0,k} = (\beta_{1,2k} + \beta_{1,2k+1})/2.$$

Then, we have reconstructed all averages at scale  $j = 1$ . Repeating this process, we obtain averages at scale  $j = 2$ , then at scale  $j = 3$ , etc. The function  $f$  is given as the limit of these averages.

## 2.3 Properties of Wavelet Constructions

These multiscale constructions have 3 key properties.

### 2.3.1 Smoothness

The fundamental and surprising fact about both refinement schemes – AI and DD – is the smoothness of refinement limits. Iterative two-scale refinement, applied to data at a fixed coarse scale, yields a sequence of imputed values consistent with a smooth function, having  $R$  continuous derivatives, where  $R$  depends on the degree  $D$  and on the type of scheme (AI/DD). Deslauriers and Dubuc showed that, for the DD scheme, the four-point neighborhood gave  $C^R$  solutions with  $R = 1.99+$ . For the AI scheme  $R(5)$  is almost 2 as well [12]. Moreover, with increasing values of  $D$ , the regularity increases, growing roughly proportionally to  $D$ . See [7, 38].

It follows from this that, for either the DD or AI wavelet transform, if the wavelet coefficients vanish beneath some fixed scale, then the object reconstructed from those coefficients will have  $C^R$  smoothness, for the same  $R$  as the refinement scheme. So setting fine-scale wavelet coefficients to zero is a kind of ‘smoothing’ operation.

### 2.3.2 Coefficient Decay

We mention two decay properties of wavelet coefficients:

- If  $f(t)$  follows a polynomial of degree  $D$  or less, the wavelet coefficients vanish. This follows immediately from the fact that the polynomial interpolation will yield imputed values which are perfectly accurate.
- Suppose that  $f(t)$  is an  $R$ -times differentiable function. Suppose also that the order  $D$  of the DD or AI scheme is greater than  $R$ . Then the wavelet coefficients obey

$$|\alpha_{j,k}| \leq C \cdot 2^{-j(R+1/2)}, \quad j \geq 0.$$

This gives them a rather rapid decay as one goes to finer scales.

### 2.3.3 Coefficients of Noise

Let  $(z_k)$  be a sequence of random ‘noise’ values, i.i.d.  $N(0,1)$ , say. Fix a scale  $J > 0$ , and consider a function  $f$  which is simply piecewise constant on dyadic intervals  $I_{J,k}$ :

$$f = z_k \text{ on } I_{J,k}.$$

This is a kind of ‘pure noise’ function, at least at scales  $0 \leq j < J$ . The AI wavelet coefficients of such a ‘noise’ function are themselves basically noise, weakly dependent and with variance independent of location and of scale  $0 \leq j < J$ . In short, the AI wavelet coefficients of ‘pure noise’ are random but roughly the same size at all scales and locations, and roughly independent. This fact is fundamental to wavelet-based denoising methods.

Note that the DD wavelet coefficients do not have such constant variances, in fact getting noisier at fine scales, this makes them unsuited for noise-removal applications.

## 3 Multiscale Representations for Manifold-valued Data

We now develop tools to represent a function  $p : \mathbf{R} \mapsto M$ , where  $M$  is a smooth manifold. Informally this is the case  $p(t)$  where  $t$  runs through the ‘time domain’. In a later section we will discuss the  $p(x, y)$  ‘space domain’ case. We will see that the Interpolating wavelet transform and Average-Interpolating wavelet transform have natural analogs in this  $M$ -valued setting.

### 3.1 Manifold notation/concepts

We use standard notation associated with manifolds; for more details, see [26, 32]. The manifold has at each  $p_0 \in M$  a tangent space  $T_{p_0}(M)$  consisting of vectors  $\theta$  corresponding to derivatives of smooth paths  $p(t) \in M$ ,  $t \in [-1, 1]$ , with  $p(0) = p_0$ . We let  $d$  denote the dimension of the manifold; the tangent space is  $d$ -dimensional as well.

The manifolds we are interested in all are conventionally viewed as Riemannian manifolds, with a metric on the tangent space. If for tangent vectors  $\theta$  we adopt a specific coordinate representation  $\theta_i$  this quadratic form can be written  $\sum_{ij} g_{ij}(p)\theta_i\theta_j$  (in the cases of interest the metric  $g_{ij}$  is typically the trivial  $\delta_{ij}$ , so that the metric is Euclidean). Now, between any two points  $p_0$  and  $p_1$  in the manifold, there is (at least one) shortest path, having arc length  $\ell(p_0, p_1)$ . Such a geodesic has an initial position –  $p_0$  – an initial direction  $\theta/\|\theta\|_2$ , and initial speed  $\|\theta\|_2$ . Geodesics are important because they follow *inertial* paths on the manifold – the result of smooth motion without external forces.

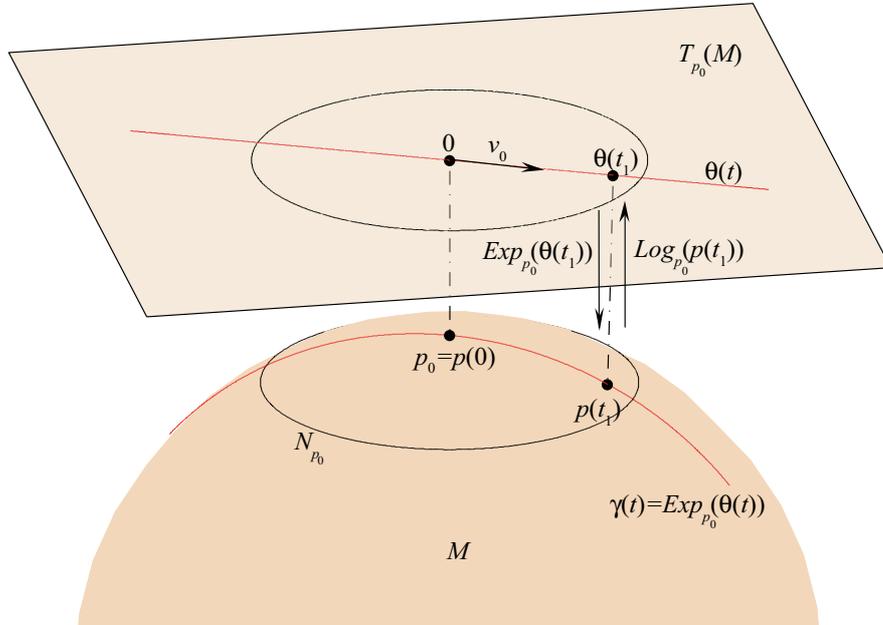


Figure 3.5: A manifold, its tangent plane, and the correspondence between a line in the tangent plane and a geodesic in the manifold.

The procedure of fixing a vector in  $\theta \in T_p(M)$  as an initial velocity for a (constant-speed) geodesic establishes an association between  $T_{p_0}(M)$  and a neighborhood of  $p$  in  $M$ . This association is one-one over a ball of sufficiently small size in  $T_{p_0}(M)$  – up to the so-called injectivity radius  $\rho$ . The association is formally captured by the *exponential map*  $p_1 = Exp_{p_0}(\theta)$ . Within an appropriate neighborhood  $N_{p_0}$  of  $p_0$ , the inverse map – the so-called ‘Logarithm map’ – is well-defined, taking  $N_{p_0} \subset M$  into  $T_{p_0}(M)$ . Formally, this correspondence is written as  $\theta = Log_{p_0}(p_1)$ . This correspondence is illustrated in Figure 3.5.

We are only interested in manifolds for which Log/Exp maps can be explicitly given; examples will be provided below.

### 3.2 $M$ -valued Interpolatory Approach

Clearly, the interpolatory pyramid  $\beta_{j,k} = p(t_{j,k})$  makes just as much sense as in the  $\mathbf{R}$ -valued case, has the same ‘hard’ redundancies  $\beta_{j+1,2k} = \beta_{j,k}$ , and the same ‘expected’ redundancies  $\beta_{j+1,2k+1} \approx \beta_{j,k}$  for smooth functions. We first discuss how to ‘predict’ coarse-to-fine on manifolds, giving  $M$ -valued two-scale refinement schemes, and then describe a wavelet pyramid  $(\alpha_{j,k})_{j,k}$  removing the redundancy from  $(\beta_{j,k})_{j,k}$ .

#### 3.2.1 Interpolatory Refinement on Manifolds

Given a sequence  $p(k)$ ,  $k \in \mathbf{Z}$  taking values  $p(k) \in M$ , we can (often) impute data at the half-integers by a scheme which might be called “Deslauriers-Dubuc in the tangent space”.

Fix an odd integer  $D$ , for example 3. To get an imputation  $\tilde{p}(1/2)$ , we use the data  $p(\ell)$  at the  $D + 1$  integer sites  $\ell$  nearest to  $1/2$ . Letting  $p_0 = p(0)$ , we then map these points to the tangent plane  $T_{p_0}(M)$  via

$$\theta(\ell) = Log_{p_0}(p(\ell)), \quad \ell = -(D - 1)/2, \dots, (D + 1)/2.$$

The resulting  $\theta(\ell)$  belong to a vector space and it makes sense to add, scale, subtract, and so on. We take a basis  $(e_j)$  for this vector space, getting a  $d$ -dimensional coordinate representation

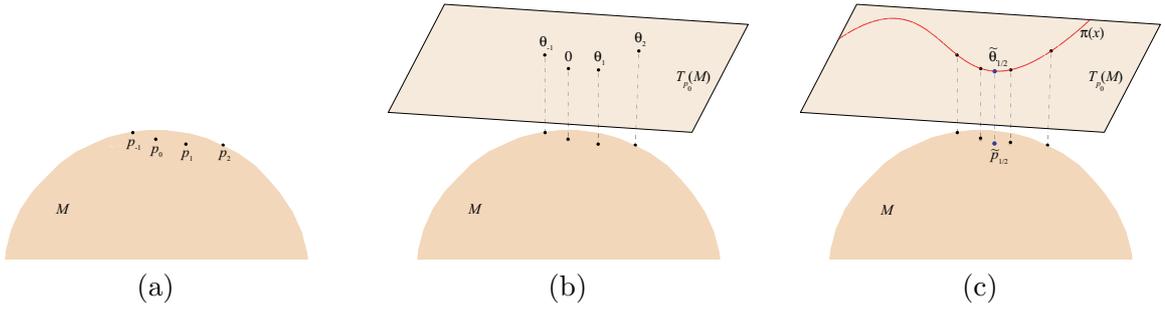


Figure 3.6: Interpolatory refinement scheme for  $M$ -valued data, at a single scale and location. (a) Points at four consecutive times  $k = -1, 0, 1, 2$ ; (b) Points mapped to the tangent plane by the Logarithm; (c) Fitted polynomial curve, and imputed midpoint.

with coordinates  $(\tau_1, \dots, \tau_d)$ :

$$\theta(\ell) = \sum_{i=1}^d \tau_i(\ell) e_i.$$

We now apply the Deslauriers-Dubuc idea to each of the *real-valued* sequences  $(\tau_i(\ell))_{\ell=-(D-1)/2}^{(D+1)/2}$ , fitting an interpolating polynomial  $\pi$  to each, thus obtaining a midpoint value  $\tilde{\tau}_i(1/2)$  for each coordinate. These imputed coordinates specify an imputed vector

$$\tilde{\theta}(1/2) = \sum_{i=1}^d \tilde{\tau}_i(1/2) e_i.$$

From this, we obtain an imputed point on the manifold by exponentiating:

$$\tilde{p}(1/2) = \text{Exp}_{p_0}(\tilde{\theta}(1/2)).$$

The process is illustrated in Figure 3.6, which considers the case where  $M$  is the sphere  $S^2$  in  $\mathbf{R}^3$ .

The points  $p(-1), p(0), p(1), p(2)$  on the manifold  $M$  are shown in (a). The tangent plane  $T_{p_0}(M)$  and the points lifted to it  $\theta(-1), \theta(0), \theta(1), \theta(2)$  are illustrated in (b). Finally, the polynomial curve  $\pi$  in the tangent plane and the imputed points at  $\tilde{\theta}(1/2)$  and  $\tilde{p}(1/2)$  are shown in (c).

This process can be repeated at other sites  $k$ , obtaining  $\tilde{p}(k + 1/2)$  from  $p(k - (D - 1)/2), \dots, p(k + (D + 1)/2)$  for all  $k$ , thus filling in imputed data at all the half-integers. It can be applied to the resulting samples/imputations at integers and half-integers to obtain imputations at the quarter-integers, and so on. An example is given in Figure 3.7, again where  $M$  is the sphere  $S^2$  in  $\mathbf{R}^3$ . We have implemented this scheme on numerous manifolds (see below), always with satisfactory results.

There is one conceivable obstacle to this approach: the data  $p(k)$  associated with a local neighborhood may not all be capable of being mapped onto one single tangent plane. This can happen if some of the  $D + 1$  points are farther from  $p(0)$  than the injectivity radius of the  $\text{Exp}$  map. Associated with a given base point  $p_0$ , there will be a specific neighborhood  $N_{p_0}$  in  $M$  on which  $\text{Log}$  and  $\text{Exp}$  are one-to-one. If the data  $p(\ell)$ ,  $\ell = -(D - 1)/2, \dots, (D + 1)/2$  do not all lie inside that neighborhood, one can imagine that certain problems will occur. At the same

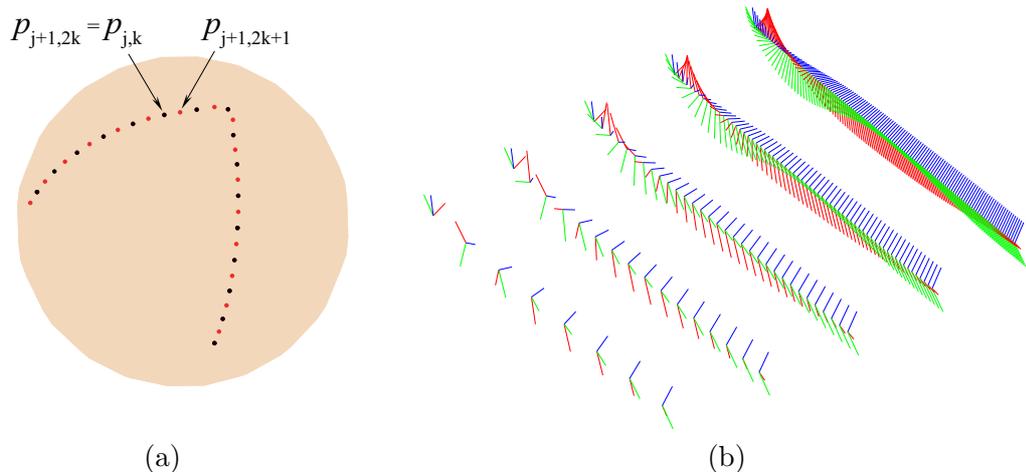


Figure 3.7: Interpolatory refinement scheme. (a) One step of DD Refinement scheme on sphere. (b) Successive iterations of DD Refinement for SO3 valued data.

time, it is conceivable that the data all lie inside such a neighborhood, but the imputed point lies outside that neighborhood, in which case, additional problems might be anticipated.

There is also an untidy aspect, in that we must apparently make a choice of coordinates in  $T_{p_0}(M)$ , and it is conceivable that this affects the results of our procedure in some way. This turns out not to be a problem; the approach is invariant to linear changes of coordinates on the tangent space.

### 3.2.2 An Interpolatory Pyramid Transform for M-valued data

Given an interpolatory refinement scheme for  $M$ -valued data, we can construct a pyramid transform for function  $p : \mathbf{R} \mapsto M$  very analogous to the classical interpolatory wavelet transform.

We start from data sampled at the coarsest scale  $\beta_{0,k} = p(k)$ ; then we apply one scale of refinement, obtaining imputed midpoints  $\tilde{p}(k + 1/2)$ . We then compare the imputed midpoints to the actual ones:

$$\alpha_{0,k} = \text{Log}_{\tilde{p}(k+1/2)}(p(k + 1/2)), \quad k \in \mathbf{Z}.$$

We can repeat this process at finer scales, starting from point samples at scale  $j$ ,

$$\beta_{j,k} \equiv p(k/2^j), \quad k \in \mathbf{Z},$$

using these to impute samples halfway in between:

$$\tilde{p}((k + 1/2)/2^j), \quad k \in \mathbf{Z},$$

and defining the wavelet coefficients

$$\alpha_{j,k} = \text{Log}_{\tilde{p}(k+1/2)}(p(k + 1/2)), \quad k \in \mathbf{Z}.$$

From the coarse-scale samples  $(\beta_{0,k})$  and the wavelet coefficients  $((\alpha_{j,k})_{k \in \mathbf{Z}})_{j \geq 0}$  one can recover  $p$  at all dyadic rationals. Indeed, one takes the coarse-scale samples  $(\beta_{0,k})$ , imputes data at the half-integers, getting  $(\tilde{\beta}_{1,2k+1})_k$ , and then sets

$$\beta_{1,2k+1} = \text{Exp}_{\tilde{\beta}_{1,2k+1}}(\alpha_{0,k}), \quad k \in \mathbf{Z}.$$

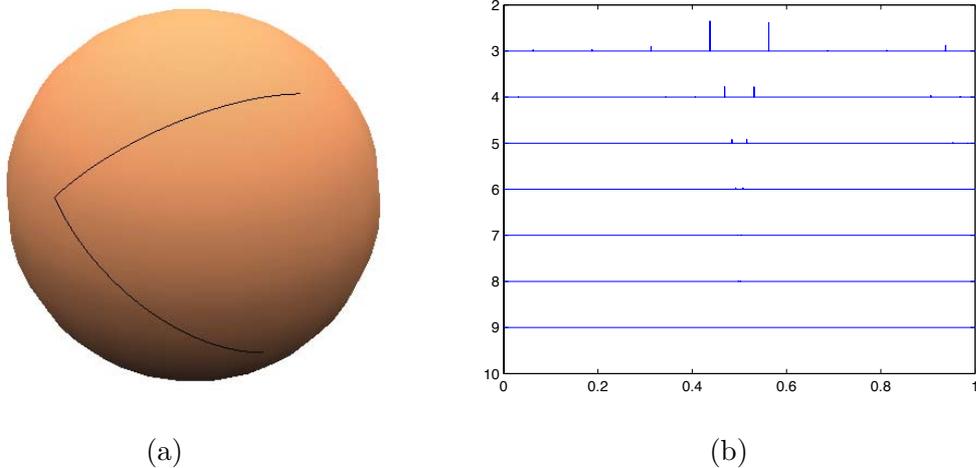


Figure 3.8: (a) Motion on sphere consisting of two segments, each one part of a great circle. (b) Frobenius norm of  $S^2$  wavelet coefficients as a function of scale and location.

Equipped then with values of  $p$  at the half-integers, one again applies two-scale refinement to get imputed values  $(\tilde{\beta}_{2,k})_k$ . The values at the fourth-integers are available via  $\beta_{2,2k} = \beta_{1,k}$  at even sites, and

$$\beta_{2,2k+1} = \text{Exp}_{\tilde{\beta}_{2,2k+1}}(\alpha_{1,k}).$$

And so on.

In Figure 3.8 we give a simple example, with  $M$  the sphere  $S^2$ . A motion on the sphere consists of two segments, each one part of a great circle. The wavelet coefficients are vectors. The figure displays the Euclidean norm of the wavelet coefficients, as a function of scale and location. At each scale, there are only a few nonzero coefficients, and these all occur in the vicinity of the ‘kink’ in the motion.

### 3.3 Midpoint-Interpolating Approach

We now develop the analog of average-interpolating wavelet transform for manifolds. Since ‘averages’ are not immediately defined for manifolds, we define a convenient notion, the midpoint, and the pyramid of midpoints. We then give the analogs of refinement and wavelet transform.

#### 3.3.1 Midpoint Pyramid

We now define a pyramid of values  $m_{j,k} \in M$  measuring (in some vague sense) ‘midpoints’ of an  $M$ -valued function over intervals  $I_{j,k}$ . This is based a coarsening operator which generates midpoints of point-pairs.

Given a pair of points  $m_0, m_1 \in M$ , suppose that there is a unique geodesic connecting them. Then there is a unique midpoint on that geodesic,  $m_{1/2}$ , say. If  $M$  is Euclidean space, of course, then  $m_{1/2}$  is just the arithmetic mean. More generally, we can regard this as a replacement for the arithmetic mean of two points in a manifold, and label it  $\text{Mid}\{m_0, m_1\}$ ; but it is only well-defined when  $m_0$  and  $m_1$  are closer together than the injectivity radius of the manifold.

A *midpoint pyramid* in  $M$ ,  $((m_{j,k})_k)_{j \geq j_0}$  is a set of points in  $M$  obeying the coarsening relation

$$m_{j,k} = \text{Mid}\{m_{j+1,2k}, m_{j+1,2k+1}\}, \quad k \in \mathbf{Z}, \quad j \geq j_0.$$

It is assumed that *Mid* is always well-defined here, i.e. that every pair has a unique midpoint. In practice this means that the coarsest-scale  $j_0$  is constrained to not be too large, so that the midpoints stay close to the data in the associated dyadic intervals.

### 3.3.2 Midpoint-Interpolating Refinement

The Midpoint-Interpolating (MI) refinement scheme for  $M$ -valued data starts from values  $m_{0,k}$  at the integers  $k$ , which will be called ‘midpoints’ for reasons given below. It then generates imputed midpoints  $\tilde{m}_{j,k}$  at all finer scales through a series of stages, repeatedly applying a two-stage refinement scheme which might be called ‘Average Interpolation in Tangent Space’.

At the first stage, a midpoint  $\tilde{m}_{1,k}$  is imputed for each dyadic interval  $I_{j,k}$  of length  $1/2$  by the following device. We fix  $D$  as an even integer, for example 4. We then, for each  $k$  at the coarse scale, collect the  $D + 1$  coarse-scale values  $m_{0,k'}$  corresponding to dyadic intervals closest to  $I_{j,k}$ . We convert those values to tangent vectors in  $T_{p_0}(M)$ , where  $p_0 = m_{j,k}$ , via

$$\theta(\ell) = \text{Log}_{p_0}(m_{0,k+\ell}), \quad (D-1)/2 \leq \ell \leq (D+1)/2.$$

We again adopt coordinates  $(\tau^i)$  on  $T_{p_0}(M)$  and fit a polynomial  $\pi_{0,k}^i(t)$  to each coordinate separately, by *average*-interpolation:

$$\text{Ave}\{\pi_{0,k}^i(t) : t \in I_{0,k'}\} = \tau^i(\ell), \quad |k' - k| < (D+1)/2.$$

We then impute averages to the coordinates at the finer scale:

$$\tilde{\tau}_{1,2k+\ell}^i = \text{Ave}\{\pi_{0,k}^i(t) : t \in I_{1,2k+\ell}\}, \quad \ell = 0, 1.$$

Using the coordinates, we impute vectors by

$$\tilde{\theta}_{1,2k+\ell} = \sum_{i=1}^d \tilde{\tau}_{1,2k+\ell}^i e_i, \quad \ell = 0, 1.$$

Finally we jump back to the manifold

$$\tilde{m}_{1,2k+\ell} = \text{Exp}_{p_0}(\tilde{\theta}_{1,2k+\ell}), \quad \ell = 0, 1.$$

This process is illustrated in Figure 3.9.

We now have available midpoints over dyadic intervals of length 1 and  $1/2$ ; by treating those as givens and repeating the above steps we can impute midpoints at the intervals of length  $1/4$ . Continuing in this way we get values at intervals of length  $1/8$ , and so on. The process is illustrated in Figure 3.10.

This process results in midpoints defined at all the dyadic intervals; it seems there is a unique continuous function  $\tilde{p}(t)$  consistent with those averages.

In fact, more seems to be true: the refinement of a sequence of coarse-scale points on the manifold  $\beta_{0,k}$  produces an imputed result  $\tilde{\beta}_{j,k}$  which has  $C^R$  regularity, where  $R = R(D)$  is the regularity of the underlying Deslauriers-Dubuc refinement scheme.

### 3.3.3 Midpoint-Interpolating Wavelet Transform

The midpoint-interpolating refinement scheme also leads to a wavelet transform, based on a midpoint pyramid rather than an average pyramid. The two-scale refinement operator gives a

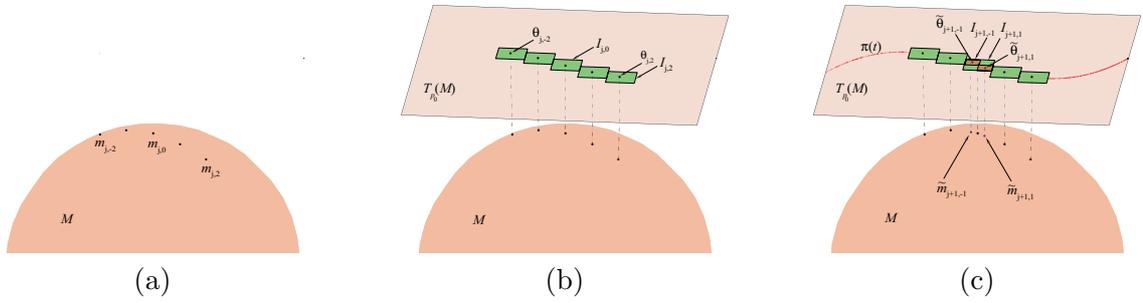


Figure 3.9: Midpoint Interpolatory refinement scheme for  $M$ -valued data, at a single scale and location. (a) Averages at five consecutive times  $k = -2, -1, 0, 1, 2$ ; (b) Averages mapped to the tangent plane by the Logarithm; (c) Fitted polynomial curve, and imputed averaged midpoints.

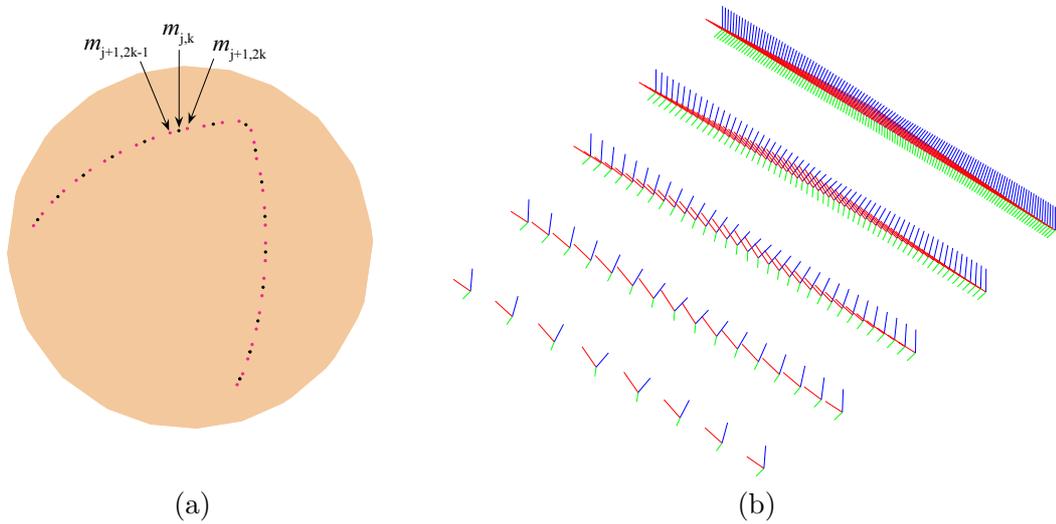


Figure 3.10: MI Refinement. (a) One step of refinement on Sphere. (b) Several iterations on  $SO_3$  data.

way to pass from coarse-scale midpoints  $(m_{j,k})_k$  to imputed fine-scale midpoints  $(\tilde{m}_{j+1,k})_k$ . We then define wavelet coefficients

$$\alpha_{j,k} = 2^{j/2} \text{Log}_{\tilde{m}_{j+1,2k+1}}(m_{j+1,2k+1}), \quad k \in \mathbf{Z}, \quad j \geq 0.$$

These measure the deviation between the behavior of fine-scale midpoints and the anticipated behavior imputed from coarse-scales. We also define coarse-scale coefficients

$$\beta_{0,k} = m_{0,k}.$$

Equipped with  $(\beta_{0,k})_k$  and  $((\alpha_{j,k})_k)_{j \geq 0}$ , we can reconstruct  $p$  by the now-familiar pyramid algorithm. We take the coarse-scale midpoints  $(\beta_{0,k})_k$  and coarse-scale wavelet coefficients  $(\alpha_{0,k})_k$  and combine them to produce midpoints  $(\beta_{1,k})_k$ .

To carry this out, we simply apply the two-scale refinement operator to the  $(\beta_{0,k})_k$ , obtaining imputed midpoints  $(\tilde{m}_{1,k})_k$  and then setting

$$\beta_{1,2k+1} = \text{Exp}_{\tilde{m}_{1,2k+1}}(2^{-j/2} \alpha_{0,k});$$

and defining each  $m_{1,2k+1}$  by the pyramid consistency relation

$$\beta_{0,k} = \text{Mid}(\beta_{1,2k}, \beta_{1,2k+1}).$$

This reconstructs the midpoints at scale  $j = 1$ . Repeating this process, we obtain midpoints at scale  $j = 2$ , then at scale  $j = 3$ , etc.

## 3.4 Properties of Wavelet Coefficients

### 3.4.1 Structural Properties

This approach makes vivid an important structural distinction between coarse-scale information  $\beta_{0,k}$  and the fine scale information  $\alpha_{j,k}$ . The  $\beta_{j,k}$  always belong to the manifold  $M$ , while the  $\alpha_{j,k}$  always belong to a tangent space  $T_{j,k} \equiv T_{\tilde{\beta}_{j+1,2k+1}}(M)$ . This generalizes the real-valued case, where the manifold and tangent space are both copies of  $\mathbf{R}$ , and so this distinction is not evident.

Since the wavelet coefficients belong to a vector space, it makes perfect sense to scale them, to operate on them with linear algebra, to quantize them, and even to set them to zero. After such operations, applying the reconstruction algorithms discussed above will yield an object which is slightly different; this can be put to use.

### 3.4.2 Inertial Motion

If  $p(t)$  describes a constant speed path along a geodesic, then all the interpolatory/midpoint interpolatory wavelet coefficients vanish. Indeed, each consecutive sequence of  $D + 1$  values in the pyramid will correspond to equispaced points on the geodesic. Each sequence of equispaced points on a geodesic will transform, under the logarithm map, into a straight line in the tangent space. The DD/AI schemes both preserve straight lines (linear functions of  $t$ ) [11, 12]. Accordingly, the imputed point in tangent space will lie along that same line, midway between its neighbors; applying the exponential map, the imputed point on the manifold will lie on the geodesic, midway between its neighbors at the coarser scale. The wavelet coefficients obey the formula

$$\alpha_{j,k} = 2^{j/2} \text{Log}_{\tilde{\beta}_{j+1,2k+1}}(\beta_{j+1,2k+1}),$$

which vanishes if  $\beta_{j+1,2k+1}$  is the geodesic midpoint of  $\beta_{j,k}$  and  $\beta_{j,k+1}$  – as just shown.

Alternately, we can say that *inertial motions have vanishing wavelet coefficients*. Indeed, inertial motions evolve within a manifold without external applied forces, and thus pursue geodesic trajectories. Hence their wavelet coefficients vanish. Ultimately, this means that very simple motions, with no active forces, are highly compressible, and require only coarse-scale data to represent them.

### 3.4.3 Coefficient Decay

We measure the size of vectors  $\alpha_{j,k}$  in the tangent space  $T_{j,k}$  according to the Euclidean norm  $\|\alpha_{j,k}\|_2$ . Importantly, this measure of size is invariant to the choice of (orthogonal) basis for  $T_{j,k}$ ; it is also an intrinsic measure, agreeing with the geodesic distance between the coarse-scale-derived imputation of fine scale behavior,  $\tilde{\beta}_{j+1,2k+1}$  and the actual value  $\beta_{j+1,k}$ .

Suppose that  $M$  is a  $C^R$  smooth manifold, with smoothness index  $R > 2$ , and that  $p(t)$  describes an  $R$ -times differentiable path through  $M$ . Suppose also that the order  $D$  of the DD or AI scheme is greater than  $R$ . Then the wavelet coefficients obey

$$\|\alpha_{j,k}\|_2 \leq C \cdot 2^{-j(R+1/2)}, \quad j \geq 0, \quad k \in \mathbf{Z}.$$

These are in some sense exact analogs of comparable properties in the classical wavelet case, and show that, for objects which are smooth, the wavelet coefficients decay geometrically with scale.

### 3.4.4 Coefficients of Noise

The reader may remark that the coefficient normalization we have chosen, with  $2^{j/2}$  factors, in (3.1) is designed to make the wavelet coefficients similar to classical wavelet coefficients in another way: so that, in the presence of ‘white noise’, they will be roughly stable as a function of scale and position. To illustrate the scaling behavior of coefficients, we give an example of what happens in the case of the sphere  $M = S^2$ . Figure 3.11 illustrates behavior in three instances cutting across smooth and noisy cases. First, it considers the MI wavelet coefficients of a ‘noise sequence’ consisting of data defined by piecewise constant behavior on dyadic intervals  $I_{j,k}$  with random values on those intervals. As can be seen, the size of typical wavelet coefficients in that case is independent of scale. Second, it considers the MI wavelet coefficients for a smooth function of time. As can be seen, the size of typical wavelet coefficients decays linearly on a log-log scale, consistent with the previous subsection. Third, it considers a smooth function with a certain degree of noise. As can be seen, the size of typical wavelet coefficients behave at coarse scales like those of a smooth function and at fine scales like those of a pure noise.

## 4 Manifolds with Tractable Exp/Log Maps

An important special case of the above ‘general manifold’ viewpoint comes when  $M$  is a Riemannian symmetric space. This special case gives a class of manifolds rich enough to model all the data types we mentioned in the introduction, and small enough to be very tractable. A Riemannian symmetric space [21, 24] is a Riemannian manifold with a globally-defined notion of reflection symmetry. In such manifolds, each pair of points  $(p_0, p_1)$  with a well-defined midpoint  $p_{1/2}$  defines a natural isometry about the midpoint, exchanging the roles of  $p_0$  and  $p_1$  and leaving  $p_{1/2}$  fixed. We specialize to this case here, which has certain advantages when considering midpoint-interpolation. We also specialize to the case where  $M$  is subspace or quotient space of

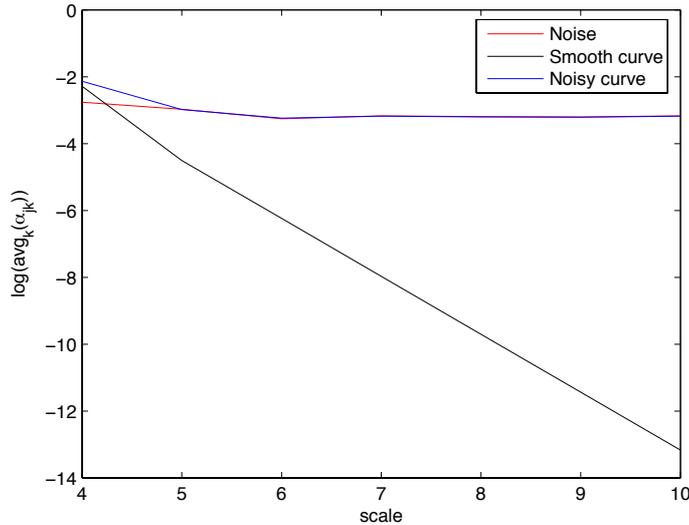


Figure 3.11:  $S^2$ -wavelet coefficients of noise, of a smooth curve and of a noisy curve.

the manifold  $GL(n)$  of  $n \times n$  matrices. Our choices have the advantage that it is straightforward to compute  $Log$  and  $Exp$ , typically involving just the matrix logarithm and matrix exponential.

In this section we review our manifolds of interest, first discussing the calculation of  $Log$  and  $Exp$  maps, and then reviewing the connections of our work to Lie Algebras.

#### 4.1 Simple Examples

$M = \mathbf{R}$ ; the reals. Here  $T_p(M)$  is just  $\mathbf{R}$ , and  $Exp_{p_0}(\theta) = p_0 + \theta$ , while  $Log_{p_0}(p_1) = (p_1 - p_0)$ . Hence both  $Exp$  and  $Log$  are linear. Our formulations for  $M$ -valued data will, in this case, of course reduce to the Deslauriers-Dubuc and Average-Interpolating schemes.

$M = \mathbf{R}^+$ ; the positive reals. Here  $T_p(M) = \mathbf{R}$ , and  $Exp_{p_0}(\theta) = \exp(\theta) \cdot p_0$ , while  $Log_{p_0}(p_1) = \log(p_1/p_0)$ . Hence both  $Exp$  and  $Log$  involve the classical exponential and logarithm functions. Our  $M$ -valued framework thus provides a notion of refinement and decomposition of strictly positive data.

$M = S^1$ ; the circle. We have two options. On the one hand, regard  $S^1$  as the unit circle in the complex plane. Then  $T_p(M) = \mathbf{R}$ , and  $Exp_{p_0}(\theta) = \exp(\sqrt{-1}\theta) \cdot p_0$ , while  $Log_{p_0}(p_1) = \arg(p_1/p_0)$ , with all formulas interpreted as involving complex arithmetic and analytic functions. On the other hand, fitting better with our general approach, instead regard  $S^1$  as the collection of real-valued matrices

$$p = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad c^2 + s^2 = 1.$$

Then  $T_p(M)$  is viewed as the collection of skew-hermitian matrices

$$\theta = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix},$$

and we equip the tangent space with the Euclidean metric  $|a|$ , inducing a Riemannian metric. Now  $Exp_{p_0}(\theta) = \exp(\theta)p_0$  and  $Log_{p_0}(p_1) = \log(p_1p_0^{-1})$ , where products mean matrix products,

$p_0^{-1}$  denotes matrix inverse, and  $\exp$  and  $\log$  denote matrix exponential and logarithm. Both approaches give equivalent results.

## 4.2 $GL(n)$ and Subgroups

A simple but general class of cases comes from the General Linear group  $GL(n)$  of  $n \times n$  real matrices and its subgroups .

$M = GL(n)$ . Here  $T_p(M)$  is again  $GL(n)$ , and the Riemannian metric is induced by the Frobenius norm on the tangent space. Then  $Exp_{p_0}(\theta) = \exp(\theta)p_0$  and  $Log_{p_0}(p_1) = \log(p_1p_0^{-1})$ , where here products mean matrix products,  $p^{-1}$  denotes matrix inverse, and  $\exp$  and  $\log$  denote matrix exponential and logarithm.

$M = SO(n)$ ; the special orthogonal group. These are matrices in  $GL(n)$  with  $p^T p = I$  and  $Det(p) = 1$ . The tangent space  $T_p(M)$  is identified with the collection of skew-hermitian matrices; for example, in the  $n = 3$  case, these take the form

$$\theta = \begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix}.$$

The Riemannian metric is again induced from the Frobenius norm on the tangent space. Again  $Exp_{p_0}(\theta) = \exp(\theta)p_0$  and  $Log_{p_0}(p_1) = \log(p_1p_0^{-1})$ , where again standard matrix interpretations are applied.

$M = SE(n)$ ; the special Euclidean group. Elements are matrices in  $GL(n+1)$  with the form

$$p = \begin{bmatrix} U & v \\ 0 & 1 \end{bmatrix}, \quad U \in SO(n), \quad v \in \mathbf{R}^n.$$

These act on  $x \in \mathbf{R}^n$  by  $p[x] = Ux + v$ . The tangent space  $T_p(M)$  is identified with the collection of matrices

$$\theta = \begin{bmatrix} \theta_0 & v \\ 0 & 0 \end{bmatrix}, \quad \theta_0 \text{ Skew Hermitian}, \quad v \in \mathbf{R}^n.$$

Again let the Riemannian metric be induced from the Frobenius norm on the tangent space. Then  $Exp_{p_0}(\theta) = \exp(\theta)p_0$  and  $Log_{p_0}(p_1) = \log(p_1p_0^{-1})$ , where again standard matrix interpretations are applied.

## 4.3 Quotients of $GL(n)$

$M = S^{n-1}$ ; the sphere in  $\mathbf{R}^n$ . Now  $M$  is the collection of vectors  $p \in \mathbf{R}^n$  with unit length  $\|p\| = 1$ . This may be viewed as a quotient of  $SO(n)$ , taking an orthogonal matrix  $U$  and retaining only the first column; hence  $M \approx SO(n)/SO(n-1)$ . The tangent space  $T_p(M)$  is the collection of vectors orthogonal to  $p$ , and so is isomorphic to  $\mathbf{R}^{n-1}$ . The Riemannian metric is again induced from the Euclidean norm on the tangent space. Then

$$Exp_{p_0}(\theta) = \cos(\|\theta\|)p_0 + \sin(\|\theta\|)\theta/\|\theta\|.$$

If  $p_1, p_0$  in  $M$  are not antipodal, then  $v = p_1 - \langle p_1, p_0 \rangle p_0 \neq 0$  and we can define

$$Log_{p_0}(p_1) = \arccos(\langle p_1, p_0 \rangle) \cdot v/\|v\|_2.$$

$M = G(n, k)$ , the Grassmanian manifold of  $k$ -planes in  $R^n$  [9, 21, 34]. For simplicity, let  $2k \leq n$ . The  $k$ -planes are in one-one relation with the orthoprojectors of rank  $k$ , and we choose

the orthoprojector representation. The tangent space can be identified with the collection of matrices formed by differentiating a one-parameter family of such projectors  $p_t$ . Such a derivative has the representation

$$\frac{d}{dt}p_t|_{t=0} = U\Theta V^T + V\Theta U^T,$$

where  $U$  and  $V$  are  $n \times k$ ,  $\Theta$  is positive diagonal, and  $U^T V = 0$ . We regard the triple  $(U, \Theta, V)$  as a polar-coordinate representation of  $\theta$ . The exponential is then

$$\text{Exp}_{p_0}(\theta) = UC^2U^T + UCSV^T + VCSU^T + VS^2V^T,$$

where  $C = \cos(\Theta)$ ,  $S = \sin(\Theta)$ . For the logarithm,  $\theta = \text{Log}_{p_0}(p_1)$ , decompose the operator  $p_0 p_1$  by singular value decomposition, getting

$$p_0 p_1 = USW^T$$

where  $S$  is  $k \times k$  diagonal, and  $U$  and  $W$  are  $n \times k$  partial orthogonal. Suppose without loss of generality that diagonal entries in  $S$  obey  $0 < S_{ii} < 1$ , set  $C = \text{diag}((1 - S_{ii}^2)^{1/2})$ , and  $\Theta = \arcsin(S)$ . The matrix  $V = (W - UC)S^{-1}$  is orthogonal, and  $U^T V = 0$ . Then  $(U, \Theta, V)$  is the polar-coordinate representation of  $\theta$ .

#### 4.4 Symmetric Matrices

$M = \text{SPD}(n)$ . This is the class of symmetric positive-definite matrices  $p$ , with tangent space  $T_p(M)$  the collection of symmetric matrices. The Riemannian metric is induced by the locally weighted Frobenius norm  $\|p_0^{-1/2}\theta\|_F$ . The local weighting gives this a different character from the  $GL(n)$  matrix case; the composition rule for positive definite matrices  $(p_0, p_1) \mapsto p_0^{1/2} p_1 p_0^{1/2}$  also deviates from the  $GL(n)$  pattern. Here  $\text{Exp}_{p_0}(\theta) = p_0^{1/2} \exp(\theta) p_0^{1/2}$  and  $\text{Log}_{p_0}(p_1) = \log(p_0^{-1/2} p_1 p_0^{-1/2})$ , where again standard matrix interpretations are applied.

#### 4.5 Fine Point: Identification of Tangent Spaces.

The ‘literal’ tangent space  $T_{p_0}$  for a submanifold  $M$  embedded in  $R^N$  is, of course, the space of path derivatives

$$\frac{d}{dt}p_t|_{t=0}$$

of smooth paths passing through  $p_0$ . We have used this crude identification above only for the case  $M = S^{n-1}$ , in which case it gives each  $T_{p_0}(S^{n-1})$  as a particular  $(n - 1)$ -dimensional hyperplane in  $\mathbf{R}^n$ . In other cases, we found it useful to pick a coordinate system in the ‘literal’ tangent space.

The Lie group examples  $GL(n)$ ,  $SO(n)$ ,  $SE(n)$  all have the structure that the tangent space literally has the form

$$T_{p_0}(G) = Ap_0.$$

where  $G$  denotes the Lie group,  $A$  the corresponding Lie algebra, and  $Ap_0$  denotes right multiplication of matrices in  $A$  by  $p_0$ . With this structure understood, we have identified the tangent space with the Lie Algebra  $A$ . Thus for  $SO(3)$ , we identify the tangent space at each point with the algebra  $so(3)$  of  $3 \times 3$  skew-hermitian matrices, and we write

$$T_p(SO(3)) \simeq so(3),$$

where it is now understood that we are speaking about  $so(3)$  as a coordinate system for the literal tangent space. From this viewpoint, each literal tangent vector  $v = \frac{d}{dt}p_t|_{t=0}$  is linked to its coordinate vector  $\theta$  by

$$v = \theta \cdot p_0.$$

This identification gives every tangent space a common algebraic structure, and shows us that the manifold  $G$  ‘looks the same’, in a very strong algebraic sense, at every point.

## 5 Implementation with Digital Data

So far, we have written as if our goal were to represent an  $M$ -valued function  $p$  of a univariate continuum argument  $p : \mathbf{R} \mapsto M$ . In reality, data will be discretely sampled over an interval and might have a two-dimensional or higher-dimensional domain. We briefly review the issues raised in adapting the above ideas to those settings.

### 5.1 Data on the Interval

It is well-understood how to adapt the interpolating and average-interpolating transforms to ‘life on the interval’ [10, 12].

In the interpolating case suppose we have equispaced data  $p(t_{J,k})$ ,  $k = 0, \dots, 2^J$ . We work fine-to-coarse computing the wavelet coefficients  $(\alpha_{J-1,k})_k$  then  $(\alpha_{J-2,k})_k$ , etc. stopping at some sufficiently coarse scale  $j_0$  where we have both wavelet coefficients  $(\alpha_{j_0,k})_k$  and point values  $\beta_{j_0,k} = p(t_{j_0,k})$ . There are  $2^j$  wavelet coefficients at scale  $j$  and  $2^{j_0} + 1$   $(\beta_{j_0,k})_k$ ’s.

The wavelet coefficients are computed as in the earlier interpolating case. We take the  $D + 1$  sites  $k'$  nearest to  $k$ , map the  $\beta_{j,k}$  to the tangent space, fit interpolating polynomials to the coordinates, impute at the midpoint  $t_{J,2k+1}$ , and define  $\alpha_{J-1,k}$  as  $2^{(J-1)/2} \text{Log}_{\tilde{p}(t_{J,2k+1})}(p(t_{J,2k+1}))$ . An important dissimilarity in the approach comes at the boundaries,  $t = 0$  and  $t = 1$ , for which there was no analog when the domain was  $\mathbf{R}$ . The phrase “sites  $k'$  nearest  $k$ ” picks different configurations for  $k$  near zero and  $2^j$  than at the interior of the interval. Near the middle of the interval, the collection of sites  $k'$  involved in the interpolation is symmetrically disposed about the midpoint  $k$ . However, if  $k$  is 0 then all the sites  $k'$  participating in the interpolatory fit will lie to the right of  $k$ ; while if  $k$  is  $2^j$ , all the sites  $k'$  will lie to the left of  $k$ . Given these remarks, interpolatory reconstruction works as one would expect, after making the obvious adaptations.

In the midpoint-interpolating case, we have fine-scale midpoint data  $m_{J,k}$ ,  $k = 0, \dots, 2^J - 1$ . We then compute the midpoint pyramid by working fine-to-coarse, setting

$$m_{j-1,k} = \text{Mid}(m_{j,2k}, m_{j,2k+1}), \quad j \geq j_0, 0 \leq k \leq 2^{j-1}.$$

We calculate the wavelet coefficients  $\alpha_{j,k}$  by a straightforward adaptation of earlier ideas. We take the  $D + 1$  intervals  $I_{j,k'}$  nearest to  $k$ , map the  $m_{j,k'}$  to the tangent space at  $m_{j,k}$ , fit average-interpolating polynomials to the coordinates, impute averages to the subinterval  $I_{J,2k+1}$ , and define  $\alpha_{J-1,k}$  as  $2^{(J-1)/2} \text{Log}_{\tilde{m}_{J,2k+1}}(m_{J,2k+1})$ . Again the approach self-modifies at the boundaries,  $t = 0$  and  $t = 1$ , for which there was no analog when the domain was  $\mathbf{R}$ . The phrase “intervals  $I_{j,k'}$  nearest  $I_{j,k}$ ” picks different configurations for  $k$  near zero and  $2^j$  than at the interior of the interval. Near the middle of the interval, the collection of intervals  $I_{j,k'}$  involved in the average interpolation is symmetrically disposed about the interval  $I_{j,k}$ . However, if  $k$  is 0 then all other intervals  $I_{j,k'}$  participating in the interpolatory fit will lie to the right of  $I_{j,k}$ ; while if  $k$  is  $2^j$ , all other intervals  $I_{j,k'}$  will lie to the left of  $I_{j,k}$ . Given these remarks, midpoint-interpolatory reconstruction works as one would expect.

## 5.2 Two-Dimensional Data

Suppose now that we have data  $p(x, y)$  where  $(x, y)$  runs through an equispaced cartesian grid. The preceding ideas adapt to this setting in the following way.

In the interpolatory case, we think of the operation that takes the array  $\beta^j \equiv (\beta_{j,k})_{k=0}^{2^j}$  into the two arrays  $\beta^{j-1} \equiv (\beta_{j-1,k})_{k=0}^{2^{j-1}}$ ,  $\alpha^{j-1} \equiv (\alpha_{j,k})_{k=0}^{2^{j-1}}$  as a *rewriting rule*  $(\beta^{j-1}, \alpha^{j-1}) = R_j(\beta^j)$

First, we define the two-dimensional point evaluation pyramid  $\beta_{j,k_1,k_2} = p(t_{j,k_1}, t_{j,k_2})$ , where  $0 \leq k_i \leq 2^j$ . We view this array, for one fixed scale  $j$ , as a matrix. Second, we apply rewriting twice, once in each direction. We apply rewriting to each column of the matrix, creating a new matrix; we then apply rewriting to each row of the matrix. Our result is a matrix with four rectangular subpanels, corresponding to locations in the array involving ‘alpha’ or ‘beta’ in the vertical direction, and ‘alpha’ or ‘beta’ in the horizontal direction. The output involving  $\beta$  in each direction is used as input to the next coarser scale; while the output involving  $\alpha$  in at least one of the two stages is considered a wavelet coefficient  $\alpha_{j,k_1,k_2}^v$ , where  $v \in \{0, 1\}^2$ , the label  $v$  indicating the directionality of the wavelet coefficient. Reconstruction is accomplished by undoing the rewriting operations in a coarse-to-fine fashion. Average-interpolating analysis works in a very similar fashion.

## 6 Examples of Multiscale Representation

We now give some simple examples of multiscale representations based on the above ideas.

### 6.1 A Trajectory on the Sphere

We continue with the earlier artificial example of the case  $M = S^2$ . A ‘V’-shaped path on the sphere is shown in panel (a) of Figure 3.8; it is actually a concatenation of two segments of great circles. The wavelet coefficients are vectors  $\alpha_{j,k} \in R^2$ , and we depict the Euclidean norm of those vectors in panel (b), as a function of scale and location. It is evident that, at fine scales, the nonzero coefficients only occur near the location of the ‘jerk’, where there is a transition from one great circle to the other.

To underscore the fact that partial reconstructions based on only a few wavelet coefficients can achieve substantial accuracy, we display in Figure 6.12 the results of reconstruction using from 8 up to 1024 coefficients. It is evident that 20 coefficients already provide a good visual representation.

### 6.2 Aircraft Headings, I

We now consider a dataset of aircraft orientations as a function of time. The data come from the flight data recorder (‘black box’) of USAir Flight 427, a Boeing 737 which crashed in September, 1994. The data themselves were converted from Pitch/Roll/Yaw form (Euler angles [25]) to time series of orientations in  $SO(3)$ . The raw data are illustrated in panel (a) of Figure 6.13, and display mostly orderly behavior, with two ‘bumps’ and then a catastrophic ‘swerve’ at the very end. The wavelet coefficients are  $3 \times 3$  matrices in  $so(3)$ , and their Frobenius norms are displayed in panel (b) Figure 6.13, as a function of scale and location. It is evident that the coefficients are small except at the end of the time interval in question, when the crash occurred.

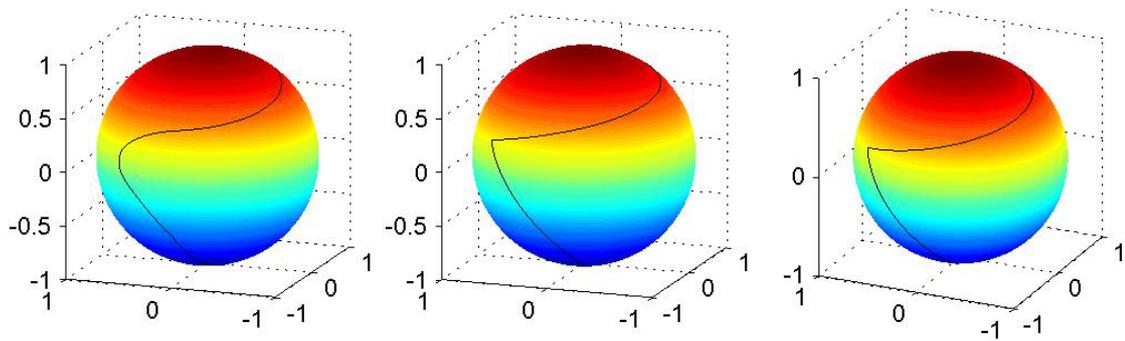


Figure 6.12: Results of partial reconstruction using only 8 coarse-scale coefficients (left); reconstruction using only 28 intermediate wavelet coefficients (center); and the original, perfect reconstruction from all 1024 coefficients (right).

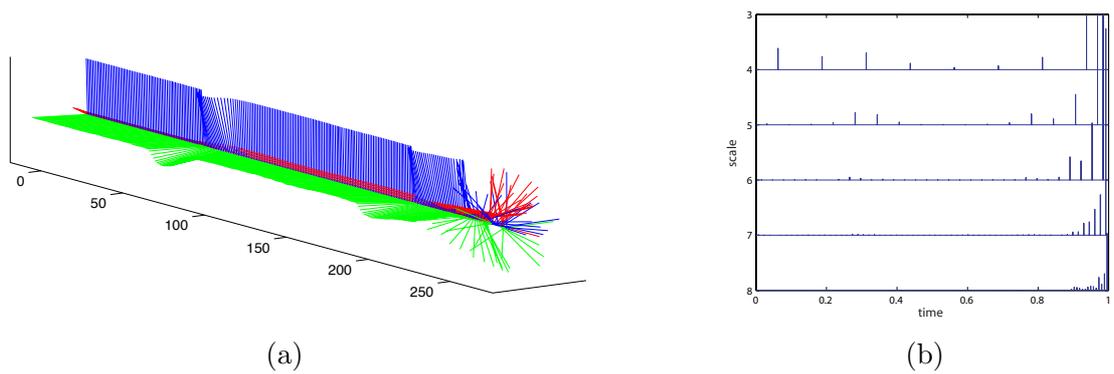
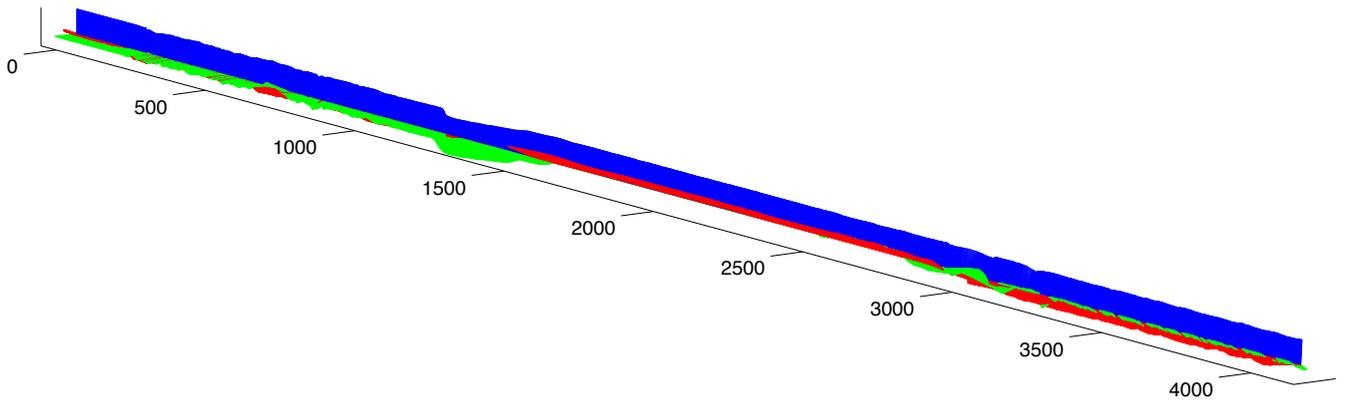
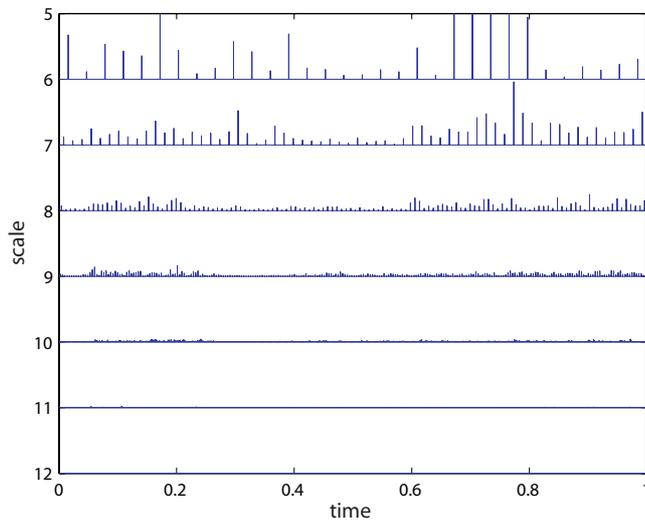


Figure 6.13: (a) Orientations versus time for US Air Flight 427 and (b) Frobenius norm of its  $so(3)$ -wavelet coefficients.



(a)



(b)

Figure 6.14: (a) Orientations versus time for a Boeing 737; (b) norms of its  $so(3)$ -wavelet coefficients. Note the rapid decrease at finer scales.

### 6.3 Aircraft Headings, II

We consider another dataset of aircraft orientations as a function of time, this time with a happier connotation. The data were supplied by Boeing Research Laboratories, and give a time series of orientations from a normal flight of a Boeing 737. Some of the raw data are illustrated in Figure 6.14 (a), 5 samples per second, and display only orderly behavior. A stretch of wavelet coefficients (more properly, their Frobenius norms) is displayed in Figure 6.14 (b), as a function of scale and location. It is clear from the near-constancy of wavelet coefficients at fine scales that the minor changes in orientation happening at those scales are akin to white noise. However, the coefficients are larger in the middle of the series indicating perhaps turbulence.

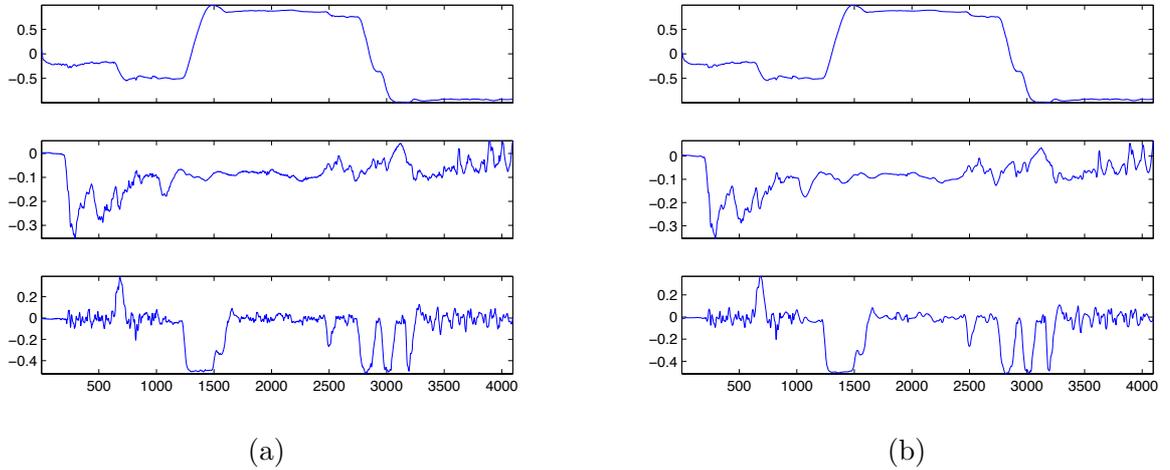


Figure 6.15: Boeing 737 orientations, raw (a) and compressed 20:1 (b). Values of  $SO(3)$  components (1,2)(1,3)(2,3) (top-bottom).

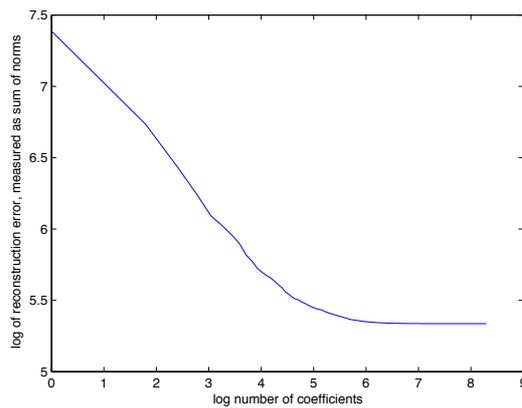


Figure 6.16: Nonlinear approximation curve for Boeing 737 Data. Number of coefficients vs. sum of error norms on a log log scale.

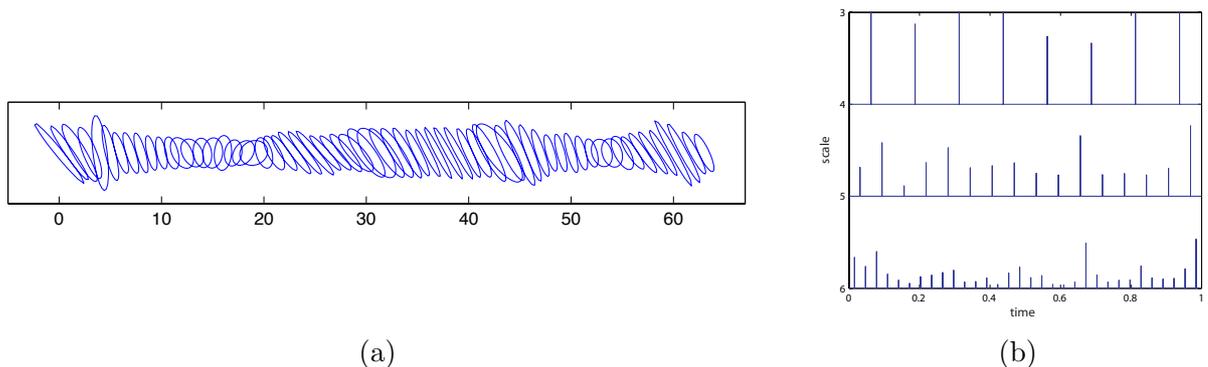


Figure 6.17: (a) Time series of PSD(2)-valued data. (b) Frobenius norm of PSD(2) wavelet coefficients.

## 6.4 Exchange rate data

We now consider a dataset  $p(t)$  of  $2 \times 2$  symmetric nonnegative-definite matrices. The matrices are covariances between exchange rates for the US Dollar vs. Euro and the US Dollar vs. British Pound, within a 10 day sliding window. Figure 6.17 shows the time series in which the symmetric matrices are depicted as ellipses. The Frobenius norms of the wavelet coefficients are depicted in panel (b) of Figure 6.17.

## 6.5 Diffusion Tensor Imaging data

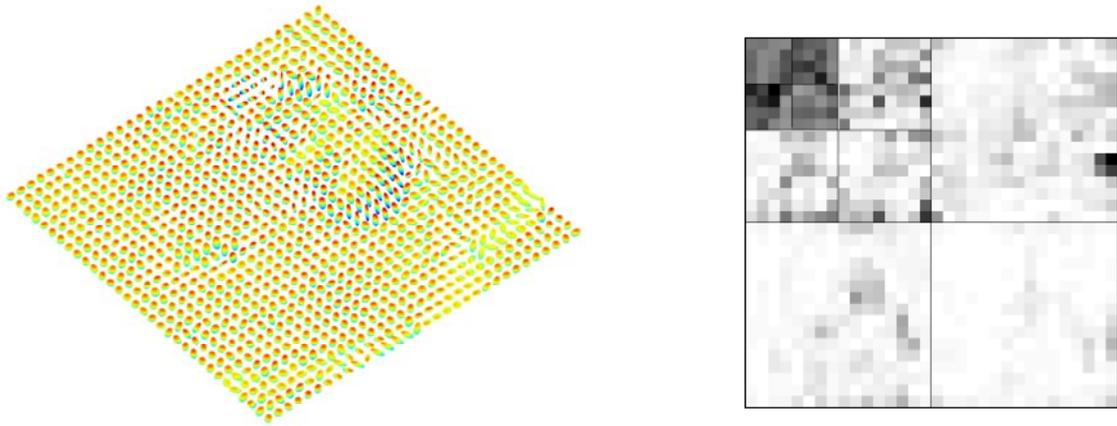
We now consider a dataset  $p(x, y)$  with  $x$  and  $y$  equispaced spatial coordinates and  $p$  being  $3 \times 3$  symmetric nonnegative-definite matrices, obtained by Diffusion Tensor Imaging. The data were obtained from the laboratory of Brian Wandell at Stanford University. Panel (a) Figure 6.18 shows a segment of such data in which the symmetric matrices are depicted as concentration ellipsoids  $E(p) = \{v : v^T p v \leq 1\}$ . The 2-D version of our MI transform was chosen with wavelet coefficients as symmetric matrices. The Frobenius norm of the wavelet coefficients is given in panel (b) of Figure 6.18. The characteristic behavior of wavelet coefficients of ordinary images is clearly seen: fine scales have only a few big coefficients, ‘around the edges’.

## 6.6 Interferometric SAR data

We now consider a dataset  $p(x, y)$  with  $x$  and  $y$  equispaced spatial coordinates and  $p \in S^1$ , obtained by interferometric Synthetic Aperture Radar [4]. Figure 6.19 (a) shows the deformation signature of Hector Mine, CA during the earthquake of Oct. 16, 1999. Each color cycle represent 2.8cm relative surface displacement. Figure 6.19 (b) shows the norms of the  $S^1$ -wavelet coefficients.

# 7 Applications

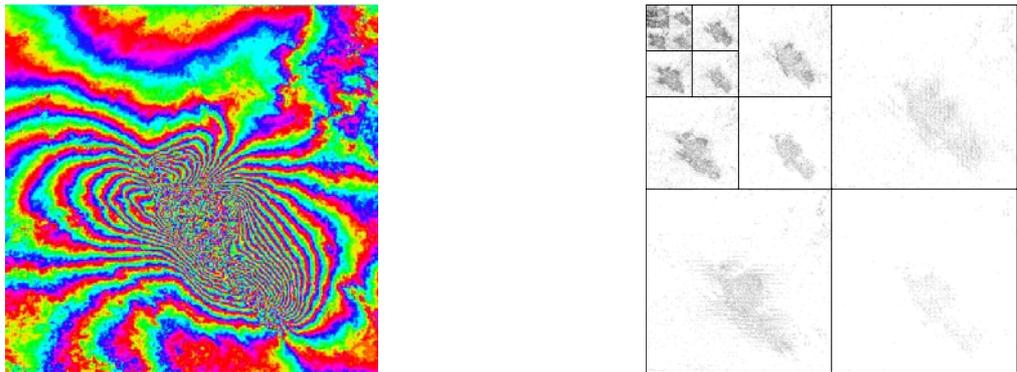
We now sketch some of the basic applications which can be developed using multiscale representations of  $M$ -valued data.



(a)

(b)

Figure 6.18: (a) A fragment of PSD(3)-valued image data. Panel (b) Frobenius norms of PSD(3) wavelet coefficients of full image.



(a)

(b)

Figure 6.19: Panel(a) SAR interferogram ( $S^1$  valued data). (b) Norms of  $S^1$  wavelet coefficients.

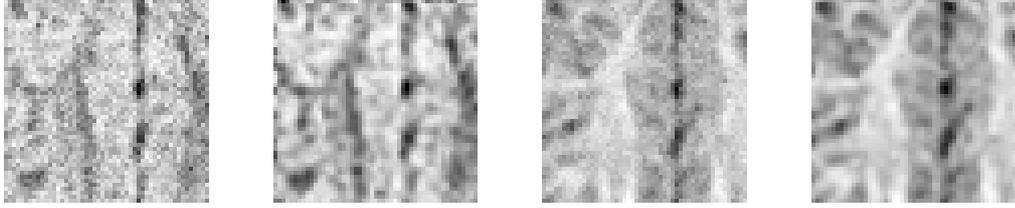


Figure 7.20: Largest and smallest eigenvectors of noisy SPD(3) data derived from Diffusion Tensor Imaging (first, third) and after SPD(3)-wavelet denoising (second, fourth).

### 7.1 Data Compression

A standard application of wavelet analysis for  $\mathbf{R}$ -valued signals is to data compression [16]. One applies a quantization and encoding scheme to the wavelet coefficients, generating a bit stream which is later used to approximately reconstruct the wavelet coefficients and ultimately an approximation to the original signal. Because the wavelet coefficients for  $M$ -valued data are organized in a fashion similar to the ordinary wavelet coefficients, it is possible to use existing ideas, like tree-coding, immediately in this context. To illustrate this, we consider the Boeing 737 data. These consist of 4096 observations of uneventful flight history over 819 seconds and so are highly compressible. Figure 6.15 Panel (a) shows the values of  $SO(3)$  components, and Panel (b) shows the values after 20:1 compression.

### 7.2 Noise Removal

Another standard application of wavelet analysis for  $\mathbf{R}$ -valued signals is to noise removal [13]. One applies a thresholding to wavelet coefficients, setting to zero those coefficients below a certain threshold. The resulting coefficients are used to reconstruct an object from which much of the noise has been removed. Because the wavelet coefficients for  $M$  valued data are organized in a fashion similar to the ordinary wavelet coefficients, it is possible to use existing ideas. One has simply to set a threshold, this time for the Frobenius norm of the matrix-valued wavelet coefficients.

To illustrate this, we consider panels (a) and (b) of Figure 7.20, which show noisy SPD(3) data derived from Diffusion Tensor Imaging, more specifically the largest and smallest eigenvector. Panels (c) and (d) show the result of wavelet denoising. Note the improved visual appearance.

### 7.3 Stochastic Process Generation

Our approach easily allows us to generate stochastic processes on manifolds which are analogous to Brownian motion and other ‘fractal’ models. To make such processes, we generate wavelet coefficients which have (say) Gaussian-distributed coordinates and which are mutually independent. We scale these coefficients according to level  $j$ , by a factor  $2^{-j\alpha}$ , and take the coarsest scale data either as zeros or generated according to some heuristic principle. An example is given in Figure 7.21, which shows a quasi-Brownian motion in  $S^2$ . Specifically, the case  $\alpha = 1/2$  behaves as a Brownian motion at fine scales, but not at the largest scales.

In this construction, the index  $\alpha \geq 0$  controls the fractal dimension. If  $\alpha = 0$  we have a sort of white noise; if  $\alpha = 1/2$  we have a pseudo Brownian motion.

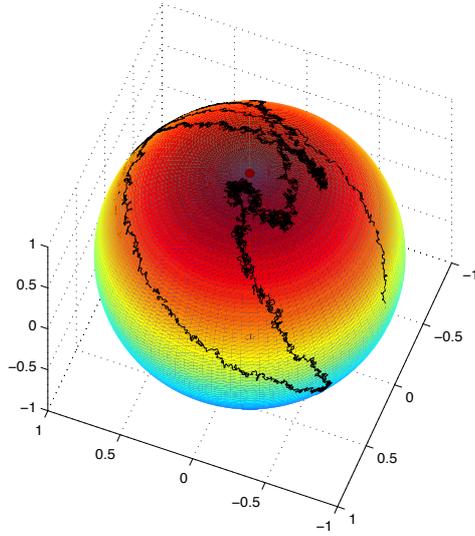


Figure 7.21: A quasi-Brownian motion on  $S^2$  generated from Gaussian random  $S^2$ -wavelet coefficients.

#### 7.4 Contrast Enhancement

Another standard application of wavelet analysis for  $\mathbf{R}$ -valued images is contrast enhancement [33]. One applies a scalar nonlinearity to the wavelet coefficients at finer scales, increasing the sizes of moderately large coefficients between certain thresholds. The resulting coefficients are used to reconstruct an object with stronger edge information. Because the wavelet coefficients for  $M$ -valued data are organized in a fashion similar to the ordinary wavelet coefficients, it is easy to transfer this idea to the case of  $M$ -valued imagery. One has simply to apply a nonlinearity, this time to the Euclidean norm of the vector wavelet coefficient.

To illustrate this, we consider panels (a) and (b) of Figure 7.22, which show SPD(3) data derived from Diffusion Tensor Imaging, more specifically the largest and smallest eigencomponent. Panels (c) and (d) show results from SPD(3)-wavelet-based contrast enhancement. Note the improved visual appearance.

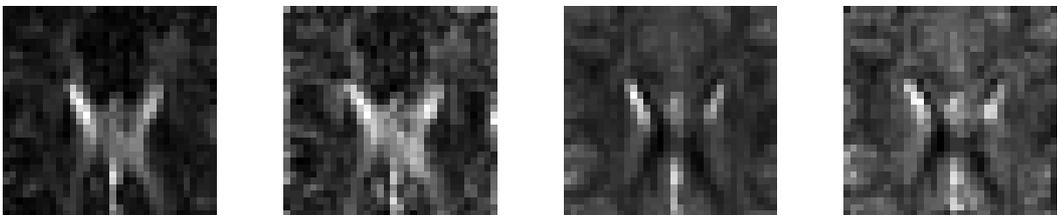


Figure 7.22: Largest and smallest eigencomponents of noisy SPD(3) data derived from Diffusion Tensor Imaging (first, third) and after SPD(3)-wavelet contrast enhancement (second, fourth).

## 8 Discussion

### 8.1 Symmlab Software

SymmLab is a collection of Matlab functions which performs the computations described in this article. In the spirit of *reproducible research* [3, 5], we are making it available to the research community at <http://www-stat.stanford.edu/~symmlab/>. SymmLab has been used by the authors to create the figures and tables used in this article, and the toolbox contains scripts which will reproduce all the calculations of this paper. It includes about 200 Matlab files, datasets, and demonstration scripts. The current version (SymmLab 090) is our initial release, and accommodates data taking values in these manifolds:

- $SO(3)$  – rotation matrices.
- $S^1, S^2$  – spheres in 2-space, 3-space.
- $G(n, k)$  –  $k$ -planes in  $R^n$ .
- $PSD(n)$  –  $n$  by  $n$  positive-definite matrices.

For this release, the data must either be observed at a sequence of equispaced ‘times’ or on an  $n$  by  $n$  grid of ‘pixels’. The current version contains demonstration scripts which illustrate the following concepts: multiscale representation, ‘wavelet coefficients’, noise removal, and data compression.

### 8.2 Other Work on Manifold Representation

There are other approaches to refinement schemes on manifolds; The Wallner-Dyn work [35] differs from our approach in that it requires only the ability to compute geodesics, not the full Log/Exp formalism, and easily generates low-smoothness schemes. On the other hand, the procedure proposed here accommodates very high-order approximation and appears to allow arbitrarily high degrees of smoothness, by selection of  $D$  large; see Section 8.3 below.

We are aware of work in progress on manifold-valued data by Peter Oswald of International University Bremen and by Thomas Yu of Rensselaer Polytechnic Institute, as an adjunct [36, 37]. We also mention that, although we have not pursued it here, our work can easily generate refinement schemes for quaternions. Quaternionic splines are popular in computer graphics, essentially because quaternions may be used to represent  $SO(3)$ -valued data. An entry into this literature may be provided by [2, 22, 23, 30] and subsequent literature.

### 8.3 Smoothness Equivalence Conjecture

It appears to be the case that the refinement schemes for  $M$ -valued data have the same regularity as their  $\mathbf{R}$ -valued counterparts. We say this based on numerous experiments with a range of manifolds. A typical case was provided by  $M = SO(3)$ . We considered a 32-long sequence  $p(k) = I$  for  $-16 \leq k < 16$ ,  $k \neq 0$ , and  $p(0) = \exp(\theta(0))$ , where

$$\theta(0) = \begin{bmatrix} 0 & 0.1 & 0.1 \\ -0.1 & 0 & 0.1 \\ -0.1 & -0.1 & 0 \end{bmatrix}.$$

In words, a slight pitch, roll, and yaw are experienced at time zero, but at other times the orientation is static. We refined this sequence using the  $M$ -interpolatory scheme based on

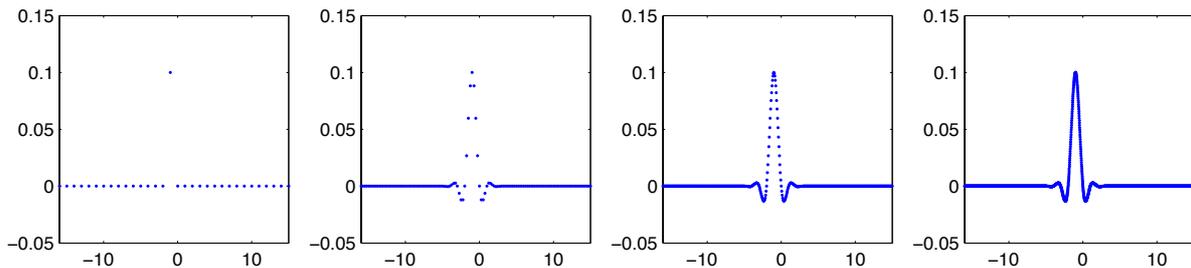


Figure 8.23: Panels – successive generations of interpolatory (DD) refinement on  $SO(3)$ , as seen through the behavior of the  $(1, 2)$  matrix entry. Note similarity to Figure 2.2.

Deslauriers-Dubuc in the tangent space with  $D = 3$ . The results are shown in Figure 8.23, giving the  $1, 2$  coordinate of the  $3 \times 3$  matrix  $\text{Log}_I(p(t_{j,k}))$ . The various iterations are displayed in separate panels. Notice the striking similarity between the apparent refinement limit and the ordinary Deslauriers-Dubuc refinement limit seen earlier in Figure 2.2.

We also refined using the midpoint-interpolating scheme based on Average-Interpolation with  $D = 6$  in the tangent space. Again note the striking similarity between the apparent refinement limit and the ordinary Average-Interpolating refinement limit seen earlier in Figure 2.4. Other entries, i.e.  $(1, 3)$  and  $(2, 3)$ , behave similarly, as do the entries seen for other data types e.g.  $S^1$ ,  $S^2$ .

It seems likely that one could prove that these manifold-based refinement schemes have limits with the same smoothness as their classical counterparts for  $M = \mathbf{R}$ . Roughly speaking, all that is needed is that each coarse-scale neighborhood  $|k - k'| < (D + 1)/2$  be all contained in a ball of sufficiently small geodesic radius. (If pairs of points in one coarse-scale neighborhood are allowed to exceed the injectivity radius, the refinement scheme might be poorly-defined.) In fact, one can easily prove the smoothness equivalence for  $M = \mathbf{R}^+$  and  $M = \mathbf{S}^1$ . It is also empirically verified for all the other cases. A thorough theoretical investigation seems called for, perhaps using the methods of [35].

This work defines nonlinear refinement schemes, therefore it should be noted that there are, outside the  $M$ -valued data context, numerous researchers developing methods for analyzing nonlinear refinement. Some of this can be traced to median-interpolating refinement schemes [18, 19, 28, 29], others to nonlinear schemes for ENO interpolation [6], surface subdivision [8], and other applications [27]. We anticipate that the ideas and tools developed in those papers should be highly relevant to resolving the smoothness equivalence conjecture.

#### 8.4 The Repeated Midpoint Operator

An apparently novel feature of our approach is the use of the midpoint pyramid as a method of summarizing manifold-valued data. At the center of this suggestion is, implicitly, the idea that the *repeated midpoint* is a good summary of noisy manifold-valued data, comparable in quality to the average for real-valued data. Let's make this more explicit. Suppose we have data  $p(1), \dots, p(n)$ , taking values in  $M$ . We desire a measure of central tendency  $\mu_n(p(1), \dots, p(n))$  taking values in  $M$  and being well-calibrated:

$$\mu_n(p_0, p_0, \dots, p_0) = p_0; \tag{8.1}$$

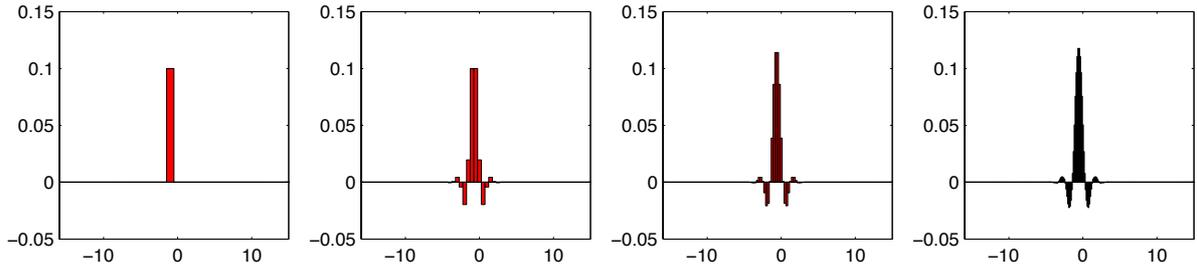


Figure 8.24: Panels – successive generations of midpoint-interpolatory (MI) refinement on  $SO(3)$ , as seen through the behavior of the (1, 2) matrix entry. Note similarity to Figure 2.4.

if the  $p(i)$  are random perturbations of a single  $p_0$ ,  $p(i) = Exp_{p_0}(\theta(i))$ , with  $\theta(i)$  independently and identically distributed with mean 0, then a law of large numbers

$$\mu_n(p(1), \dots, p(n)) \rightarrow_P p_0, \quad n \rightarrow \infty. \quad (8.2)$$

We might even desire more,

$$E(dist^2(\mu_n, p_0)) = O(n^{-1}), \quad n \rightarrow \infty, \quad (8.3)$$

where  $dist$  denotes geodesic distance. If  $M$  is acted on by a group  $G$  of transformations  $g$ , we also desire  $G$ -equivariance:

$$\mu_n(gp(1), \dots, gp(n)) = g\mu_n(p(1), \dots, p(n)). \quad (8.4)$$

Note that for  $M = \mathbf{R}$  and  $\mu_n$  the simple arithmetic average, we get all these properties (with  $G$  the  $ax + b$  group of affine coordinate changes). The simple average has another property: the average of a linear function is just the midpoint of the function. In the  $M$ -valued case we could put this as follows if  $\pi(t)$  is a geodesic in  $M$

$$\mu_n(p(1), \dots, p(n)) = p((n-1)/2). \quad (8.5)$$

In effect, the midpoint pyramid we have proposed defines a functional  $\mu_n$  for  $n$  dyadic,  $n = 2^j$ , for some  $j \in \mathbf{Z}$ , by recursive application of midpoints. That is,

$$\mu_n(p(1), \dots, p(n)) \equiv Mid[\mu_{n/2}(p(1), \dots, p(n/2)), \mu_{n/2}(p(n/2+1), \dots, p(n))].$$

Let's call this the *repeated midpoint* functional. Given data  $p(i/n)$ ,  $i = 0, \dots, n-1$ , the midpoint pyramid values are

$$m_{j,k} = \mu_{n/2^j} \{p(i/n) : (i/n) \in I_{j,k}\}.$$

Now clearly, the repeated midpoint functional has properties (8.1) and (8.5). For the spaces  $SO(n)$ ,  $SE(n)$  and  $GL(n)$  which are also groups and hence act on themselves, we also have (8.4). Extensive experiments reveal behavior entirely consistent with (8.2), in fact with (8.3).

There has recently been considerable interest in defining 'means' on manifolds, unfortunately in most cases the ideas proposed are not very computationally efficient, for example an expensive iterative process. In comparison, the repeated midpoint is highly efficient computationally and has good properties. Further research into its properties seems called for.

The notion of repeated application of nonlinear measures of central tendency has previously been useful in the guise of median-interpolating wavelet transforms, where repeated medians of 3 have proved useful [18]; and in generalizations [19, 28, 29]. In this connection we note that repeated medians have been shown to have interesting statistical properties [31], showing that some nonlinear pyramid summaries can be successfully analyzed.

## References

- [1] Matthew J. Ahlert, David L. Donoho, Iddo Drori, Inam Ur Rahman, and Victoria C. Stodden. Symmlab web site. <http://www-stat.stanford.edu/~symmlab/>.
- [2] Alan H. Barr, Bena Currin, Steven Gabriel, and John F. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions. *Computer Graphics*, 26(2):313–320, 1992.
- [3] Jon Buckheit, Maureen Clerc, David L. Donoho, Mark Reynold Duncan, Xiaoming Huo, Iain Johnstone, Jerome Kalifa, Ofer Levi, Stephane Mallat, and Thomas Yu. Wavelab web site. <http://www-stat.stanford.edu/~wavelab/>.
- [4] Curtis W. Chen and Howard A. Zebker. Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization. *J. Optical Society of America A*, 18(2):338–351, 2001.
- [5] Sou Cheng Choi, David L. Donoho, Ana Georgina Flesia, Xiaoming Huo, Ofer Levi, and Danzhu Shi. Beamlab web site. <http://www-stat.stanford.edu/~beamlab/>.
- [6] Albert Cohen, Nira Dyn, and Basarab Matei. Quasilinear subdivision schemes with applications to ENO interpolation. Technical report, 2001.
- [7] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.
- [8] Ingrid Daubechies, Olof Runborg, and Wim Sweldens. Normal multiresolution approximation of curves. Technical report, Bell Labs, 2002.
- [9] Chandler Davis and William M. Kahan. The rotation of eigenvectors by a perturbation, III. *SIAM J. Numer. Anal.*, 7(1):1–46, 1970.
- [10] Gilles Deslaurier and Serge Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5(1):49–68, 1989.
- [11] David L. Donoho. Interpolating wavelet transforms. Technical report, Department of Statistics, Stanford University, 1992.
- [12] David L. Donoho. Smooth wavelet decompositions with blocky coefficient kernels. In Larry L. Schumaker and Glenn Webb, editors, *Recent Advances in Wavelet Analysis*, pages 259–308. Academic Press, Boston, 1993.
- [13] David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.
- [14] David L. Donoho, Nira Dyn, Peter Schroeder, and Victoria C. Stodden. Presentation: Multiscale representation of equispaced data taking values in a lie group. In *Approximation Theory at 60: Conference in Honor of R. A. DeVore*, Charleston SC, May 2001.

- [15] David L. Donoho, Nira Dyn, Peter Schroeder, and Victoria C. Stodden. Presentation: Multiscale representation of equispaced data taking values in a symmetric space. In *Curves and Surfaces IV*, Saint Malo, France, June 2002.
- [16] David L. Donoho, Martin Vetterli, Ron A. DeVore, and Ingrid Daubechies. Data compression and harmonic analysis. *IEEE Transactions on Information Theory*, 44(6):2435–2476, 1998.
- [17] David L. Donoho and Thomas Pok-Yin Yu. *Deslauriers-Dubuc: Ten Years After*, pages 355–369. CRM Proceedings and Lecture Notes. American Mathematical Society, 1999.
- [18] David L. Donoho and Thomas Pok-Yin Yu. Nonlinear pyramid transforms based on median-interpolation. *SIAM Journal of Math. Anal.*, 31(5):1030–1061, 2000.
- [19] Tim N. T. Goodman and Thomas Pok-Yin Yu. Interpolation of medians. *Advances in Computational Mathematics*, 11(1):1–10, 1999.
- [20] Ami Harten. Multiresolution representation of cell averaged data. Technical report, UCLA Computational and Applied Mathematics Report 94-21, 1994.
- [21] Sigurdur Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. American Mathematical Society, Providence, R.I., 2001.
- [22] Myoung-Jun Kim, Myung-Soo Kim, and Sung Yong Shin. A  $C^2$ -continuous B-spline quaternion curve interpolating a given sequence of solid orientations. In *CA '95: Proceedings of the Computer Animation*, pages 72–81, 1995.
- [23] Myung-Soo Kim and Kee-Won Nam. Interpolating solid orientations with circular blending quaternion curves. *Computer-aided Design*, 27(5):385–398, 1995.
- [24] Serge Lang. *Introduction to Differentiable Manifolds*. Springer-Verlag, New York, second edition, 2002.
- [25] Yi Ma, Stefano Soatto, Jana Kosecká, and Shankar S. Sastry. *An Invitation to 3-D Vision*. Springer, 2004.
- [26] Barrett O’Neill. *Elementary Differential Geometry*. Academic Press, New York, second edition, 1997.
- [27] Peter Oswald. Smoothness of a nonlinear subdivision scheme. In *Curves and Surface Fitting*, pages 323–332, Saint-Malo, 2003. Nashboro Press.
- [28] Peter Oswald. Smoothness of nonlinear median-interpolation subdivision. *Advances in Computational Mathematics*, 20:401–423, 2004.
- [29] Jong-Shi Pang and Thomas Pok-Yin Yu. Continuous m-estimators and their interpolation by polynomials. *SIAM Journal of Numerical Analysis*, 42(3):997–1017, 2004.
- [30] Ravi Ramamoorthi and Alan H. Barr. Fast construction of accurate quaternion splines. In *SIGGRAPH '97 Proceedings*, pages 287–292, 1997.
- [31] Peter J. Rousseeuw and Gilbert W. Bassett. The mediant: A robust averaging method for large data sets. *J. Amer. Statist. Assoc.*, 85:97–104, 1990.

- [32] Michael Spivak. *A Comprehensive Introduction to Differential Geometry*. Publish or Perish Inc., Houston, third edition, 1999.
- [33] Jean-Luc Starck, Fionn Murtagh, Emmanuel J. Candes, and David L. Donoho. Gray and color image contrast enhancement by the curvelet transform. *IEEE Transactions on Image Processing*, 12(6):706–717, 2003.
- [34] Gilbert W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory*. Academic Press, Boston, 1990.
- [35] Johannes Wallner and Nira Dyn. Convergence and  $C^1$  analysis of subdivision schemes on manifolds by proximity. *Computer Aided Geometric Design*, 22, 2005. To appear.
- [36] Gang Xie and Thomas Pok-Yin Yu. On a linearization principle for nonlinear  $p$ -mean subdivision schemes. In *Advances in Constructive Approximation*, pages 519–533, 2004.
- [37] Gang Xie and Thomas Pok-Yin Yu. Smoothness analysis of nonlinear subdivision schemes of homogeneous and affine invariant type. *Constructive Approximation*, 2004.
- [38] Thomas Pok-Yin Yu. *New Developments in Interpolating Wavelet Transforms*. PhD thesis, Scientific Computing and Computational Mathematics, Stanford University, 1997.