UNIVERSITY OF CALIFORNIA
SANTA CRUZ

VISUALIZING MULTIMODAL UNCERTAINTY IN ENSEMBLE VECTOR FIELDS

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Brad Eric Hollister

June 2015

The Dissertation of Brad Eric Hollister is approved:

Professor Alex Pang, Chair

Professor Suresh Lodha

Dr. David Kao

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Visualizing Multimodal Uncertainty

in Ensemble Vector Fields

by

Brad Eric Hollister

Often times, simulations involve repeated runs where certain parameters, e.g. initial and boundary conditions, or model parameters are varied slightly, in order to capture the variability of the phenomenon being studied. The results are referred to as ensembles. Ensembles are very attractive since they represent both the data values and their uncertainty. Ensembles challenge us to extend traditional visualization assuming that the ensemble represents the distribution of all possible simulation outcomes given an input parameter space. Extending the traditional paradigm is also better suited for complex data associated with ensemble vector fields (EVFs). Derived features of the EVF allow for their summary visual analysis. This approach is related to traditional methods of visualization for crisp fields but require the definition and calculation of additional derived features of interest.

We first focus on a consolidated and extensible representation of EVF. A distinguishing aspect of this dissertation is the treatment of the values at each spatial point of the ensemble field as forming a probability distribution function (PDF) that need not conform to a Gaussian distribution. We present a new method for interpolation

of distributions of 2D vector fields, required for handling velocity distributions. We also include velocity probability density information from the EVF in the feature set of streamlines.

Another defining characteristic of this work is considering streamline information content and geometrically based streamline clusters as a derived feature of EVF. We apply a suitable and proven streamline clustering method first introduced to summarize regions of crisp vector fields. Our contribution is redefining this method for use in EVF, both for seed points over the spatial domain and for entire sub-regions of the EVF. We also show correlation between the associated cluster counts and streamline information content at seed points in the EVF.

Our goal is to enable simulation scientists and consumers of ensemble data sets, such as weather forecasters, to visualize areas of predicted flow that are improperly represented by a Gaussian simplification. The potential impact of this work ranges from better representation of current weather prediction forecasts for public consumption to the refinement of computational fluid dynamics (CFD) models.

Acknowledgments

I want to thank my advisor Alex Pang for providing guidance during our research. I also want to thank David Kao and Suresh Lodha for serving on my defense committee, and Thomas Peterka for providing advice while serving on my advancement committee. I wish to thank Pierre Lermusiaux and his group at MIT for sharing the ocean and lock-exchange data sets, and Harald Obermaier for providing the industrial stirring data set. Lastly, I want to thank Matthew Jee for providing LaTeX code for three of the diagrams (from specification) in chapter 6, and Mitchell Allen for his replication experiments of the *Bivariate Quantile Interpolation* algorithm introduced in chapter 3.

Chapter 1

Introduction

1.1 Motivation

Many applications in physical sciences, engineering, statistics, risk assessment and decision science, etc. use Monte Carlo methods to model phenomena with uncertainty. The input parameter space of the models is repeatedly sampled, and each sample set solved using a deterministic model, to produce a possible outcome of the model. Each possible outcome is called a realization, and the collection of realizations from repeated runs is called an ensemble.

An everyday example is the weather forecast. Forecasts are usually obtained by running Monte Carlo simulations on a number of weather models. Each model may in turn be run with a set of input parameter whose values are drawn from a probability distribution associated with each parameter. An ensemble weather forecast may produce

several fields such as temperature, humidity, pressure, and velocity. This gives rise to ensemble scalar fields and EVFs. Each one is a distribution of values about the scalar or vector variable at each location and time. Hence, ensembles encode both the data and the uncertainty about the data.

Treatment of the different fields of an ensemble depends on the cardinality of the field. Ensemble scalar fields may be summarized using parametric statistics in certain situations. For example, the mean field can be used as a proxy for the ensemble scalar field, while the standard deviation field may be used as a representation for the uncertainty of the scalar field. This works when the distribution can be adequately characterized by parametric statistics. However, this is not always the case. In situations where this assumption does not hold, non-parametric statistics may be computed and mapped visually, or the spatial distributions themselves may be displayed.

Treatment of ensemble vectors at discrete locations may proceed in a similar fashion. Ensemble mean and standard deviation at each location may be calculated and mapped to uncertainty vector glyphs. Note that while some assumptions are imposed on the input parameters e.g. Gaussian distribution, the EVF may not exhibit such properties. In fact, most of the interesting events happen when and where such assumptions fall apart.

1.2 Goals

The theme of this dissertation is to remove Gaussian assumption when applied to EVF visualization. As a start, we first consider velocity distributions at each EVF grid point of the ensemble simulation output. These distributions are multivariate. A two-dimensional multimodal velocity distribution is shown in Fig. 1.1. Our first goal is to evaluate and extend interpolation of non-parametric bivariate velocity PDF for efficient use. Using interpolated PDF, we then extend traditional streamlines to incorporate multimodal uncertainty encoded in the EVF. We take two approaches. One approach is to directly use velocity PDF for advection. Another approach adds feature information to each member streamline from the field PDF.

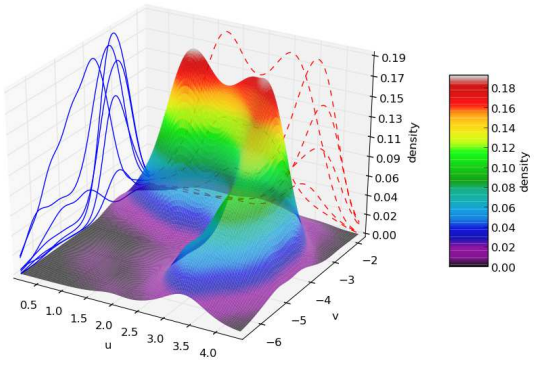


Figure 1.1: Kernel density estimate of a multimodal velocity distribution from an EVF grid point. The EVF is derived from an ocean current simulation at a constant pressure level. Marginal probabilities corresponding to the u and v velocity components are projected onto the side walls.

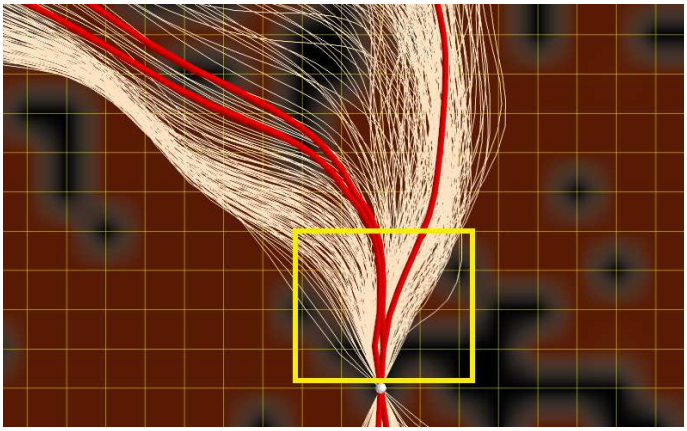


Figure 1.2: Three potential types of multimodality in an EVF: (1) The maroon areas in this EVF show multimodal velocity distributions. The red flow lines were integrated using peak velocities from PDF. (2) Bifurcating flow bundles from each realization are shown using a *spaghetti plot* from a single seed location. (3) The yellow box highlights an area of modal behavior in flow field as shown in schematic Fig. 1.4.

Figure 1.2 shows bundles of streamlines from each realization. These streamline clusters represent multimodality in the EVF as seen in a traditional *spaghetti plot*. As another goal, we characterize and quantify these modes for the entire spatial domain. This will allow inspection of the entire EVF with regard to such multimodal flow.

A related purpose to the previous goal will be to visualize flow similarity in sub-regions of the spatial domain. A schematic is shown in Fig. 1.3, where modes present in the realizations are combined in the EVF as their mean value. Figure 1.4 considers multiple modes in the EVF. By doing so, we acknowledge more than one primary flow direction. The possibility of multiple possible flow is lost when only summary statistics such as the mean field is used.

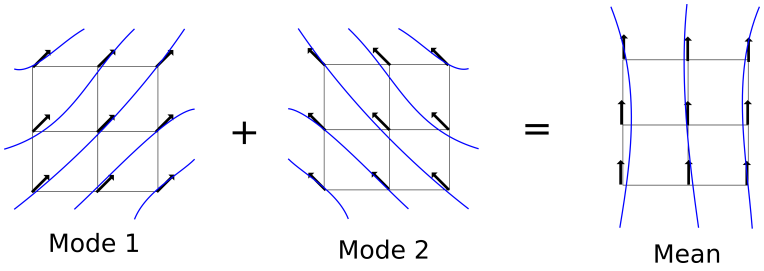


Figure 1.3: The average of two modes of regional flow in the EVF is taken to be the mean flow.

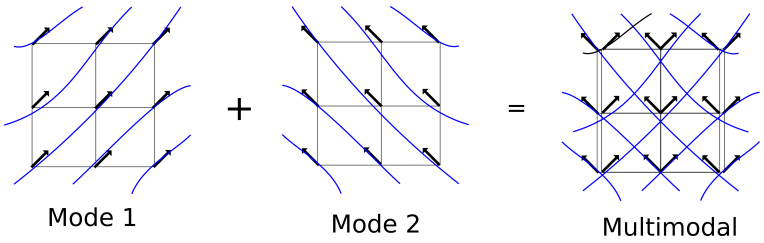


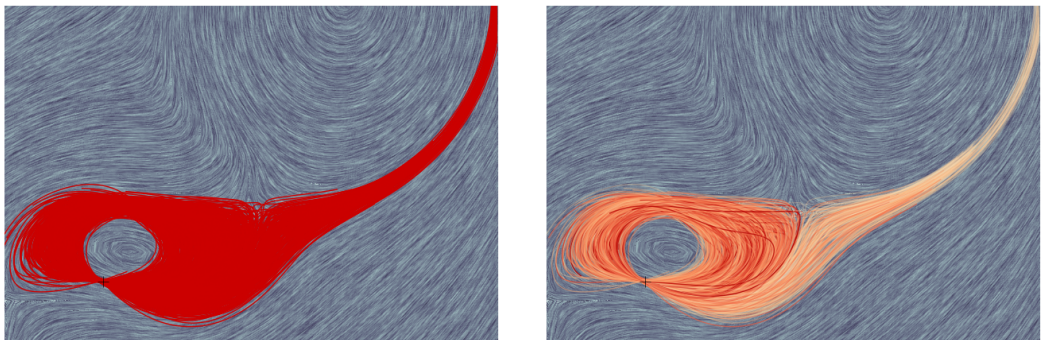
Figure 1.4: The union of two modes of regional flow in the EVF is taken to be a multimodal distribution.

1.3 Overview

Summarizing EVF is the purpose of this dissertation. Such summaries are formulated with particular goals, as discussed in section 1.2. Overcoming Gaussian simplification and unimodal assumptions are key contributions. As a preview, our result shown in Fig. 1.5 depicts non-parametric uncertainty from the EVF, and represents significant improvement over traditional methods.

Our first approach is to show EVF as a field of non-parametric PDF, and then observe EVF uncertainty expressed using velocity density estimation. As a second approach, we treated EVF as separate realizations of which we compare member streamlines. All methods presented are meant to provide analysis of different EVF aspects.

After describing related works in chapter 2, this dissertation is divided into four



(a) (b)

Figure 1.5: (a) Traditional "spaghetti" plot. (b) Streamlines rendered to show relative non-parametric uncertainty derived from the EVF.

primary chapters. Chapter 3 describes a novel interpolation method for bivariate probability density estimates. Chapter 4 provides applications and analysis of using non-parametric distributions in the context of ensemble visualization and reiterates our method of PDF interpolation in a larger context. The subsequent chapter presents an extension to traditional streamline visualization using the PDF from the EVF. In chapter 6, we provide methods to visualize EVF transport similarity both for the entire field and in selected regions. In the last chapter, we provide our conclusions and how each method in this dissertation has a common theme related to EVF analysis. We also suggest some possible directions for future research.

Chapter 2

Background

2.1 Crisp Vector Fields

Crisp vector fields represent certain vector fields. Numerous methods are available to visualize vector fields both from local and global viewpoints, as described in Laramee et al. [33]. Common methods of visualization are streamlines for steady flow and pathlines for unsteady flow. Pathlines are calculated using integration methods such as Euler, Runge-Kutta, etc. Stability of solutions is a key concern coupled with computational time and storage. In general, dense seeding is required to derive meaningful visualization using flowlines. Figure 2.1 depicts the directions that nvarious types of vector field visualizations can be performed.

Integration must proceed to sufficient stopping criteria to cover the vector field domain fully. Dense seeding can be used to expose critical points in the vector field, along

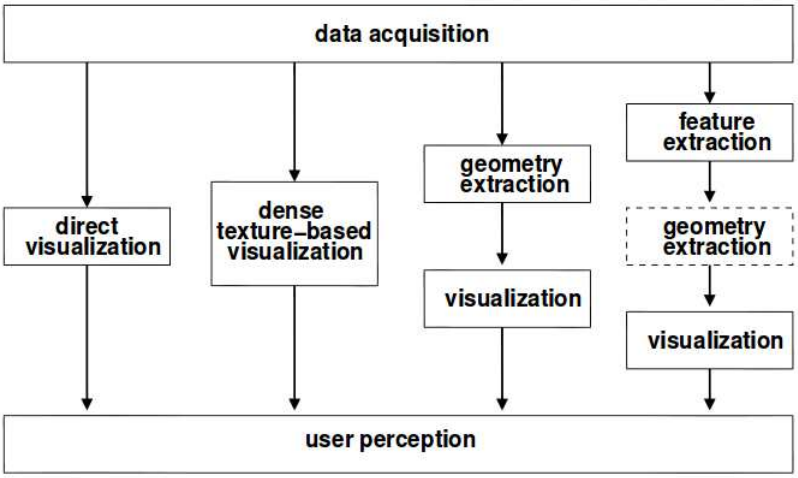


Figure 2.1: Classification of flow visualization techniques [33] - (left) direct, (middle-left) texture-based, (middle-right) based on geometric objects, and (right) based on geometric objects, and (right) feature-based.

with the classification of the critical point. Vector field singularity classes are: saddle points, attracting/repelling nodes, attracting/repelling focii, and centers. Such topological information is called *feature extraction* in vector fields, as Helman et al. discusses in [22].

Dense field visualization of flowlines is accomplished with methods that seek to simplify groupings of lines. A popular technique is line integral convolution (LIC) as first done by Cabral et al. [8] and its many more recent variants. For uncertain vector fields, such a technique is less useful, since LIC provides general notions of flow in a single image. LIC is often used as a reference field for comparison. That reference can be the mean or a single representative crisp realization. Laidlaw et al. have surveyed various other ways of visualizing crisp vector fields [31]. Kuhn et al. employ a camera-aligned method using triangle-strips to replace field lines [30]. Their width

is dependent on flowline density. This has been utilized in real-time using OpenGL to reduce streamline clutter.

2.1.1 Lagrangian Flow Classification

Lagrangian flow classification is based on material transport in vector fields, and thus provides a global picture of the vector field, see Sadlo et al. [73]. A displacement map, called the *flow map* Φ, is derived from the vector field using integration. Because a (possibly time-varying) crisp vector field can be described with the differential Eq. 2.1, the *flow map* is subsequently defined in Eq. 2.2.

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = v(x(t),t) \tag{2.1}$$

$$\Phi(x(t);T) = x(t+T) \tag{2.2}$$

Equation 2.2 describes the final location of a particle seeded at x at time t and advected for an interval T. The field is not required to be time-varying and in such a case, T simply refers to the number of integration steps forward or backward in the *flow map*.

2.1.2 Finite-time Lyapunov Exponent

Taking the largest eigenvalue of the left-Cauchy Green deformation tensor as in Eq. 2.3, we find the magnitude of the direction of greatest stretching in the flow medium at $x(t)$. The left-Cauchy Green deformation tensor removes effects of reference frame rotations as might be present in the *flow map* gradient.

$$\lambda_{max}(\nabla\Phi(x(t);T)^T \nabla\Phi(x(t);T)) \tag{2.3}$$

The magnitude of maximum expansion, λ_{max} in Eq. 2.4, is the largest eigenvalue from Eq. 2.3. The Finite-time Lyapunov Exponent (FTLE) is a logarithmic scaling of the magnitude of maximum expansion.

$$FTLE(x(t),T) = \frac{1}{T} log\sqrt{\lambda_{max}} \tag{2.4}$$

FTLE can be viewed as a scalar field over the vector field domain as seen with a tilted bar flow data set in Fig. 2.2. When the height ridges of this scalar field are found, we get a topological representation of the regions that share in either a contractive or expansive material property of the flow medium.

2.2 Ensemble Vector Fields

Ensemble vector fields (EVF) are uncertain vector fields derived from Monte Carlo simulations. Repeated runs of the same simulation, with varying simulation input pa-

Figure 2.2: FTLE computed for a tilted bar data set with total integration time of 1.0 second. From Schneider et al. [79].

rameters, produce member realizations that taken together, can be considered as a distribution of all possible, outcomes of the field for a given input parameter space. A time-varying flow field can be described as in Eq. 2.5.

$$v : \Omega \times I \rightarrow \mathbb{R}^d \tag{2.5}$$

Using the notation in Hummel et al. v is defined over a spatial domain $\Omega \subseteq \mathbb{R}^d$ [23]. The time interval is $I \subseteq \mathbb{R}$. An EVF is a set of m vector fields over the same spatial domain and the ensemble space can be considered to be the intersection of all such vector fields, $\Omega_{EVF} = \Omega_1 \cap .,. \cap \Omega_m$ and $I_{EVF} = \Omega_1 \cap ... \cap I_m$:

$$EVF : \{1, ..., m\} \times \Omega_E \times I_E \rightarrow \mathbb{R}^d \tag{2.6}$$

$EVF(i,.,.)$ corresponds to the i^{th} realization in our ensemble. We can see an example of particle transport in an ensemble (Fig. 2.3).

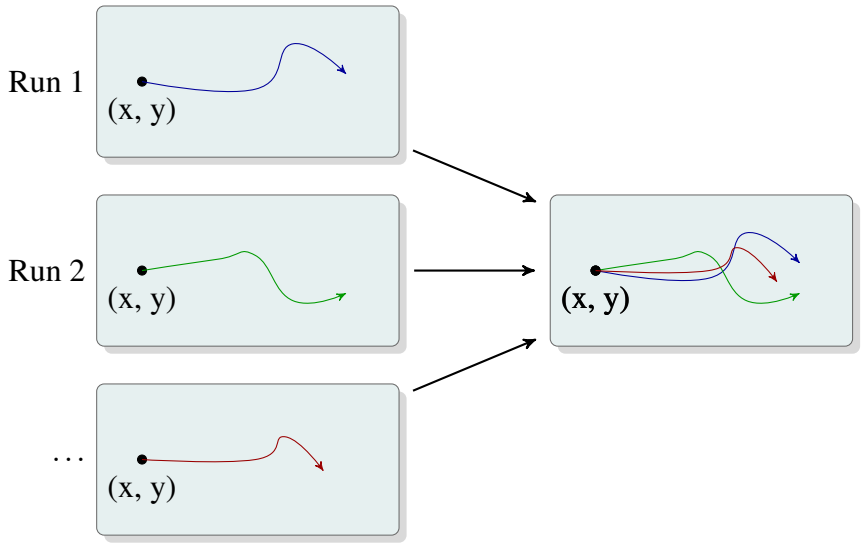


Figure 2.3: A particle started at identical positions in all vector fields of an ensemble is transported to different final positions. Different locations in the ensemble lead to stronger or weaker separation of particle positions. Notice the conceptual similarity between ensemble divergence and individual member flow field divergence.

2.2.1 Finite-time Variance Analysis

A probabilstic variant of FTLE is called the FTVA, Eq. 2.7. It takes the covariance matrix of particle positions advected over the ensemble domain from given seed locations. It was first presented by Schneider et al. and is shown in Fig. 2.4 [79].

$$FTVA(x(t),T) = \frac{1}{T} log \sqrt{\lambda_{max}(Cov(x(t);T))} \tag{2.7}$$

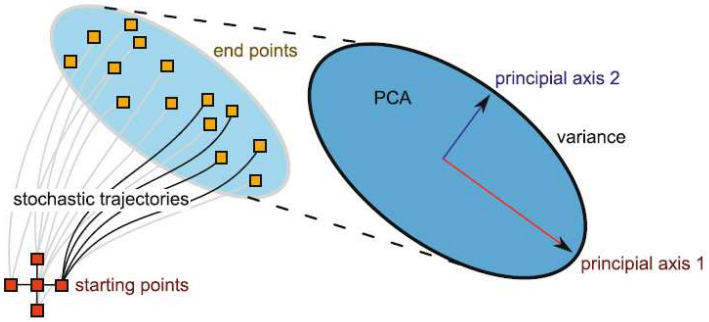


Figure 2.4: Stochastic integration from a starting point gives a distribution of end points due to uncertainty. A principal component analysis of the start and end point distributions provides information about the maximum amount of stretching [79].

2.3 Clustering

The primary aim of our research is to identify multimodal similarity (i.e., more than one cluster) in an EVF. To this end, we utilize multiple clustering algorithms. We provide background on relevant clustering algorithms both for *point data* and *trajectories*. The use of the term *trajectory* in this work is used interchangeably with the term *streamline* or *pathline*. Geometrically, a trajectory is represented as a polyline and the same clustering methodology can be applied.

We also endeavor to minimize the need for prior information or assumptions about the data, such as the number of possible clusters. Therefore, we omit discussion on *vector field k-means*, a partition based trajectory clustering algorithm by Ferreira et al. [15]. We do not use this approach as it requires an input parameter k which denotes the number of output clusters. While we use Expectation-maximization [2] (EM) for point data, where EM fits a specified number of radial basis functions (Gaussians), we are less certain as to the number of *similar* trajectories that may exist and so employ more

exploratory clustering approaches. It is worth considering the clustering survey papers by Ilango et al. [25] and Estivill-Castro [14] which outline some general strategies.

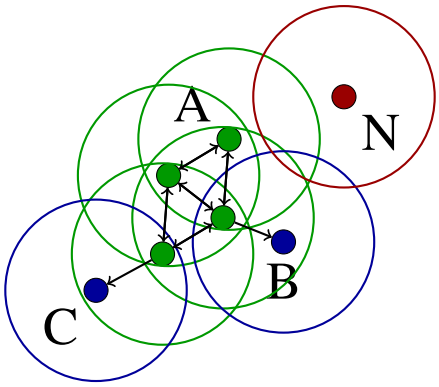


Figure 2.5: Illustration of DBSCAN cluster analysis requiring minimum points constituting a cluster to be three. Points around A are core points. Points B and C are not core points, but are density-connected via the cluster of A (and thus belong to this cluster). Point N is Noise, since it is neither a core point nor reachable from a core point. DBSCAN also requires a maximum distance parameter ε that determines density-connected points [13].

2.3.1 Point Data

Numerous point data clustering algorithms exist and can be sorted into the six categories: partitioning models (e.g., k-means), hierarchical models (e.g., BIRCH), density-based models (e.g., DBSCAN and OPTICS), grid-based models (e.g., STING), distribution models (e.g., EM) and graph-based models (e.g., minimum spanning tree), Pedregosa et al. [54].

We choose fitting Gaussian mixture models via EM and the density-based model, DBSCAN, as our initial clustering methods for point data. Because we are investigating material transport, such methods are most applicable as they represent clusters based

on Euclidean distance metrics. We provide a simple outline to the DBSCAN algorithm in Fig. 2.5, as it is extended in the implementation of TRACLUS, a trajectory clustering technique by Lee et al. [34].

2.3.2 Trajectories

This section briefly describes two prominent trajectory clustering algorithms, TRACLUS and a feature-based approach presented by Lu et al. [40].

TRACLUS

TRACLUS is an extension of DBSCAN for trajectories. It generalizes Euclidean distance to include parallel, angular, and linear distance for line segments. Using this generalized distance, representative streamlines are calculated for clusters, by averaging the individual line segments in a cluster. The results of using TRACLUS for a set of hurricane tracks can be seen in Fig. 2.6.

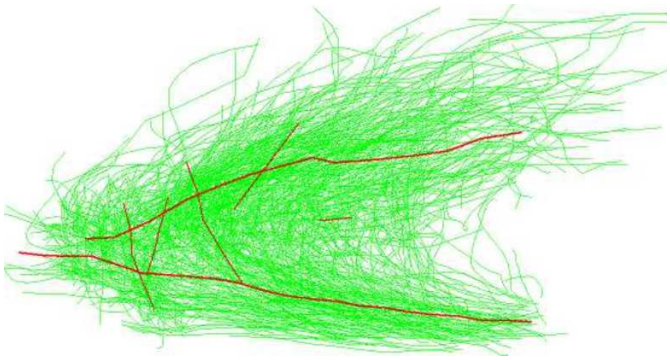


Figure 2.6: TRACLUS clustering result for a hurricane data set [34].

Feature-based Analysis

Another approach for streamline clustering is based on features. Lu et al. cluster streamlines regardless of their location in the simulation field [40]. They measure the amount of curvature or torsion in a streamline and then form sets based on the occurrence of those characteristics. An example is shown in Fig. 2.7.

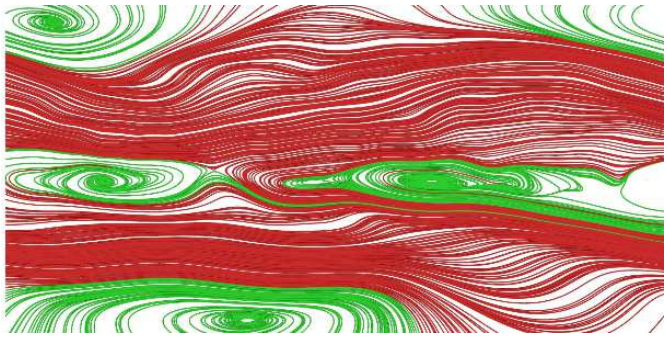


Figure 2.7: Clustering results based on curvature distribution. The green cluster corresponds to vortex flow and the red one corresponds to straight flow [40].

Chapter 3

Bivariate Quantile Interpolation

Probability distribution functions (PDFs) may be estimated from members in an ensemble. For an ensemble of 2D vector fields, this results in a bivariate PDF at each location in the field. Vector field analysis and visualization, e.g. streamline calculation, require an interpolation to be defined over these 2D density estimates. Thus, a non-parametric PDF interpolation must advect features as opposed to cross-fading them, where arbitrary modalities in the distribution can be introduced. This is already achieved for 1D PDF interpolation via inverse cumulative distribution functions (CDFs). However, there is no closed-form extension to bivariate PDF. This chapter presents one such direct extension of the 1D closed-form solution for bivariates. We show an example of physically-coupled components (velocity) and correlated random variables. Our method does not require a complex implementation or expensive computation as does *Displacement Interpolation* [4]. Additionally, our method does not suffer

from ambiguous pair-wise linear interpolants, as does *Multivariate Gaussian Mixture Model Interpolation* (see chapter 4).

3.1 Introduction

A fundamental operation used in most visualization algorithms is interpolation. Interpolation is used in workhorse visualization techniques such as marching cubes, direct volume rendering, and streamline generation, and many other popular algorithms. Performing interpolation is well defined when the data points and the interpolants are crisp. However, this is not the case when the data points consist of a distribution.

With increasing interest in representing uncertainty in modeling and simulation with techniques based on Monte Carlo methods, we are now faced with the challenge of analyzing and visualizing ensemble fields. Ensemble fields are made up of individual realizations, each a possible outcome, of the simulation. Assuming that the ensemble fields are defined over a regular Cartesian grid, a popular approach is to treat all the values at a given grid point from different realizations as a distribution. Recent works in this area have primarily assumed that the distribution follow a Gaussian distribution. Even more recent efforts have extended this to non-Gaussian distributions.

In this chapter, we extend a closed-form 1D probability distribution function interpolation method [68] that advects features for non-parametric probability distribution functions (PDFs). It is essentially a method that interpolates quantiles of the corresponding cumulative distribution functions (CDFs) and then solves for the interpolant

PDF. However, until now, there was no direct extension to bivariate distributions, which are needed to represent vector PDF interpolation.

This chapter addresses both physical vector fields (e.g. velocity, angular momentum) and a vector of scalar fields (e.g. two scalar fields, for instance temperature and humidity in a vector representation). Our interpolation method is general, and applies to both types of vectors. Our method is necessary for physical vector fields that cannot be decomposed into univariate distributions and for correlated random variables.

This work is motivated by the need for a non-parametric PDF interpolation that scales to large data sets by employing variable computational cost for required levels of accuracy. It is primarily applicable to multi-dimensional fields whose component random variables are correlated. Uncorrelated random variables may be treated as univariates.

3.2 Related Work

A nice overview of statistical techniques for spatial interpolation was presented by Myers [45]. The techniques range from simple linear models with no covariance, to those using spatial structure functions. The survey however does not include non-parametric distribution interpolation. The paper does claim that interpolation is a solution to an inherently ill-posed problem, namely that it is a problem of prediction with limited data. For that, multiple models with different purposes can be employed. A more detailed survey [37], but focusing on geostatistical applications, compare meth-

ods according to different criteria such as local vs. global support, deterministic vs. stochastic, univariate vs multivariate, linear vs. nonlinear, etc. Among the methods that consider stochastic data, they assume normal distribution.

Within the visualization community, there are also a number of recent publications that address stochastic interpolation. Scheuermann, et al. [78] present a form of Kriging interpolation of spatial data for Gaussian distributions using a parameter-based approach. This technique relies on computing a covariance matrix and that the underlying data be formed from a Gaussian process. Pfaffelmoser et al. [56] visualize isosurfaces via a raycasting scheme, and perform spatial interpolation assuming the data has a Gaussian distribution at each location. Likewise, Pothkow et al. [61] discuss isocontour visualization of normally distributed data. They interpolated between grid points using the 0th and 1st moments without spatial correlation considerations. Their subsequent work [63] considered the effects of spatial correlation in visualizing isosurfaces using probabilistic marching cubes. An alternative method of looking at global correlation structures in a hierarchical fashion was presented in [57].

When data do not follow a Gaussian distribution, a more general uncertainty model is needed. Liu et al. [38] propose a Gaussian mixture to represent the distribution of voxel values in air temperature data. They perform volume rendering on the data set and interpolate between pairs of a fixed number of Gaussian components along cast rays. In their study, they found that four Gaussian kernels are sufficient for a variety of data sets that they examined. In addition, they support stationary and anisotropic correlations in

the process, but at the expense of considering multimodal qualities of the probability distributions at grid points. For non-parametric representations of non-Gaussian distributions, operations on the distributions require different handling. Love, et al. [39] discuss two forms of a non-parametric interpolation method via convolution addition of probability distributions as well as bin-wise addition. Pohl, et al., [60] first transform the (discrete) distribution to Euclidean space via a set of Log Odds operations, where they can then be manipulated using conventional addition and multiplication. Results are then mapped back to probabilistic space via a reversible transform.

Uncertainty in vector fields is of great interest to at least two broad fields: environmental science e.g. oceanography and meteorology [36, 35], and fiber tracking of diffusion tensor magnetic resonance images (DT MRI). Both [80] and [3], discuss non-Gaussian methods in these areas of research. Otto, et al. present analysis of 2D [48] and 3D velocity fields [49] using particle advection, critical points, and segmentation of field topology. Petz et al. [55] also analyze uncertain velocity fields modeled as Gaussian random fields with spatial correlation.

There is a growing body of work on probabilistic fiber tracking. Unlike velocity fields, the tracks here represent fiber connectivity from one region to another and are obtained by integrating the major eigenvector field of symmetric DT MRI data set. The main source of uncertainty can be attributed to inadequate resolution in the data acquisition stage. However, there are numerous other sources as well [6]. While most of the earlier works on probabilistic fiber tracking delved on the inadequacy of the simple

tensor representation to show alternative trajectories due to multiple fiber populations within a cell, more recent works are based on high angular resolution diffusion imaging (HARDI) data which makes it is possible to describe fiber orientations using more sophisticated formulations such as spherical harmonics and multi-tensor representations. In a recent paper, Jiao et al. [26] describe a local, icon-based presentation of an ensemble field of fiber orientation distribution functions (ODF). The results of our work can be used towards spatial analysis of such ensemble fields, for example.

There is much interest in the meteorological community to provide better visualization of forecast data. Slingsby et al. [85], discuss how users interpret and use weather data, specifically hurricane data. Storm path information are examined from historical data. They draw attention to spatial and temporal clustering and its undervalued status among those currently employing such visualization software. Weather forecasts are usually based on an ensemble of predictions. For that, Potter et al. [64] describe a framework for viewing stochastic information from ensembles. This package allows for visualization of spaghetti plotting, etc. of weather data. Zhang, et al. [75] present Noodles, a software package for displaying uncertainty in streamlines and other weather data visualization for ensemble forecasting. Potter et al. [65] describe a software tool to visualize two-dimensional sets of distribution data. It displays a contour of field PDF values and allows for a normed difference between data PDFs and an ansatz selected by the user. More recently, Phadke et al. [58] present two novel visualization methods for ensembles. Primarily, they allow simultaneous viewing of multiple ensemble members.

They also present a technique called *Screen Door Tinting* which applies value changes to field points that show differences between ensembles.

From the point of view of users, Martin et al. [42] point out the difficulty of users to identify hurricane directional movement and speed from current data visualization, or directly on vector fields. In a similar study, Broad et al. [7], further emphasize interpretation and usage of complex weather data. They show how a general interpretation of a Gaussian distribution of hurricane direction prediction can lead to inaccurate views on the probability within a *cone of uncertainty*. Clearly, if multimodal velocity distribution is calculated with such a broad region of uncertainty using a Gaussian assumption, incorrect estimation of the probability of hurricane direction can occur, most specifically within the general population who can be greatly impacted by such interpretation. A non-Gaussian consideration for vector field visualization together with a redesigned visualization may rectify this issue to a degree.

The method presented by Liu et al. [38], which proposes a Gaussian mixture model, is insufficient for bivariate PDF. Despite the use of a fixed number of Gaussian basis functions for PDF estimates, the interpolation is only unambiguous for 1D PDF when pairing Gaussian components by the order of their mean parameter. For 2D Gaussian mixture models, there is no such ordering. It is possible to order bivariate Gaussian components based on their mean probability, but this does not follow from the 1D case of ordering based on the mean parameter value.

Displacement Interpolation, developed by Bonneel et al. [4], is a general method

for multivariate PDF interpolation. It is shown to reduce to the 1D PDF interpolation presented by Read [68]. It satisfies the advection of features by interpolating populations instead of cross-fading them. (Bonneel et al. provide an in-depth discussion of this property in their paper.) It is based on solving for intermediate solutions to the Earth Mover's Distance, a minimum cost problem of transforming one PDF into another. This method does not scale well to 2D field interpolation, however. It is computationally costly, with current CPU implementations (using compiled code) taking on the order of minutes to hours for interpolation between only two PDFs. In the form presented by Bonneel et al., it is developed only for interpolation between two PDFs.

3.3 Bivariate Quantile Interpolation

3.3.1 Derivation

We extend a CDF based interpolation method for use with bivariate PDF, which is needed for uncertain 2D velocity fields. The original 1D method was analytically derived in [68], and is shown below. Here, $F(x)$ is the CDF with its associated PDF, $f(x)$, as in Eq. 3.1.

$$F(x) = \int_{-\infty}^{x} f(h)dh \tag{3.1}$$

f_0 and f_1 are two known PDF used for the interpolation. Their CDF are F_0 and F_1, respectively. The quantile y corresponds to both x_0 and x_1 in Eq. 3.2 and Eq. 3.3.

$$F_0(x_0) = y \tag{3.2}$$

$$F_1(x_1) = y \tag{3.3}$$

$\bar{F}(\bar{x})$ is the interpolant CDF found from linearly interpolating between x_0 and x_1, shown in Eq. 3.4 and Eq. 3.5.

$$\bar{x} = (1-\alpha)x_0 + \alpha x_1 \tag{3.4}$$

$$\bar{F}(\bar{x}) = y \tag{3.5}$$

Using F^{-1}, we have $F_0^{-1}(y) = x_0$, $F_1^{-1}(y) = x_1$ and $\bar{F}^{-1}(y) = \bar{x}$. Substituting these results into equation 3.4 yields:

$$\bar{F}^{-1}(y) = (1-\alpha)F_0^{-1}(y) + \alpha F_1^{-1}(y) \tag{3.6}$$

Knowing that $dx = dF^{-1}(y)$, $dy = dF(x)$ and $dx/dy = (dy/dx)^{-1}$, we have:

$$\frac{dF^{-1}(y)}{dy} = \left[\frac{dF(x)}{dx}\right]^{-1} = \frac{1}{f(x)} \tag{3.7}$$

Thus, applying d/dy to Eq. 3.6, and solving for $\bar{f}(\bar{x})$ produces:

$$\bar{f}(\bar{x}) = \frac{f_0(x_0) f_1(x_1)}{(1-\alpha) f_1(x_1) + \alpha f_0(x_0)} \tag{3.8}$$

Our contribution is the novel extension to 2D PDF interpolation. Equation 3.9 represents the 2D conceptual extension of Eq. 3.8. The parameter $t \in [0, n]$ is introducted to provide a unique one-to-one correspondence between x and y pairs on the corresponding quantile curves from two bivariate PDFs f_0 and f_1, the known PDFs we interpolate from.

$$\bar{f}(\bar{x}(t_i), \bar{y}(t_i)) = \frac{f_0(x_0(t_i), y_0(t_i)) f_1(x_1(t_i), y_1(t_i))}{(1-\alpha) f_1(x_1(t_i), y_1(t_i)) + \alpha f_0(x_0(t_i), y_0(t_i))} \tag{3.9}$$

Additionally, α is the linear interpolation factor that determines the Euclidean distance in the scaled probability space of the interpolant $[\bar{x}(t_i), \bar{y}(t_i)]^T$. This relationship is expressed in Eq. 3.10.

$$\begin{bmatrix} x_0(t_i) \\ y_0(t_i) \end{bmatrix} + \alpha \begin{bmatrix} x_1(t_i) - x_0(t_i) \\ y_1(t_i) - y_0(t_i) \end{bmatrix} = \begin{bmatrix} \bar{x}(t_i) \\ \bar{y}(t_i) \end{bmatrix} \tag{3.10}$$

The parameter t, is taken as the fraction of the arc length of the rectified quantile curves from f_0 and f_1. The arc length L of curve C is defined as in Eq. 6.10 on the interval $[a, b]$. $ds^2 = dx^2 + dy^2$ for the infinitesimal line segment ds.

$$L(C) = \int_a^b ds = \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \tag{3.11}$$

For finite numerical approximations, where C is the image of a continuous function $l : [a,b] \to \mathbb{R}^n$, we have:

$$L(C) = \sup_{a=t_0<t_1<\cdots<t_n=b} \sum_{i=0}^{n-1} d(l(t_i), l(t_{i+1})) \tag{3.12}$$

All quantile curves are indexed with the same number n of finite t_i, regardless of the value of $L(C)$. Effectively then, each $[x(t_i), y(t_i)]^T$ pair between curves are the same fractional length of their curve.

Our method does not seek to minimize various metrics placed on mapped curve segments. For instance, we do not minimize distance in the sample space between paired samples on the quantile curves being interpolated but use the simpler heuristic of arc length parameterization.

For interpolation within a grid cell, Eq. 3.9 can be extended using bilinear interpolation via both α and β weights for the orthogonal directions of the grid. The α and β weights within the unit cell are shown in Fig. 5.1.

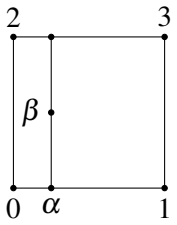


Figure 3.1: Unit cell interpolation using both α and β.

In Eq. 3.13, we show the interpolation solved for the unit cell case. For brevity, we omit the $(x(t_i), y(t_i))$ pairs associated with each PDF. Each vertex represents the

estimated PDFs from the ensemble for those locations. Setting either α or β to zero reduces to interpolation along a line.

$$\bar{f} = \frac{f_0 f_1 f_2 f_3}{f_1 f_2 f_3 + \alpha A + \beta B + \alpha\beta C} \tag{3.13}$$

A, B and C are shown in Eq. 3.14, Eq. 3.15 and Eq. 3.16, respectively.

$$A = f_0 f_2 f_3 - f_1 f_2 f_3 \tag{3.14}$$

$$B = f_0 f_1 f_3 - f_1 f_2 f_3 \tag{3.15}$$

$$C = f_1 f_2 f_3 - f_0 f_2 f_3 - f_0 f_1 f_3 + f_0 f_1 f_2 \tag{3.16}$$

3.3.2 Algorithm

The major steps of the quantile interpolation method are shown in Fig. 4.5.

Stages *gather samples* and *estimate density* are implementation specific. We do not cover their implementation details here and the user may choose varying approaches depending on the data. For example, kernel density estimation (KDE) [84] with different window settings can be used for density estimation.

For the *CDF calculation* stage, we collect (u, v) pairs for each requested quantile curve. We use u and v to refer to the components of a 2D velocity vector, in place of x

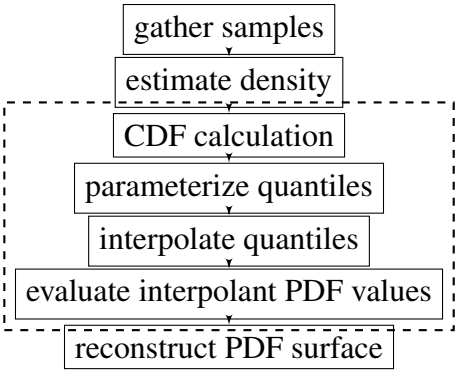


Figure 3.2: Quantile PDF interpolation method. Dashed outline signifies core method stages discussed.

Data: $dobj$, $quantiles$
Result: $qpts$
initialize $qpts$;
for $i = dobj.min\_u$ **to** $dobj.max\_u$ **do**
  for $j = dobj.min\_v$ **to** $dobj.max\_v$ **do**
    $d =$ density in region (min_u, min_v) to (i, j);
    foreach q *in* $quantiles$ **do**
      if $q - TOL \leq d < q + TOL$ **then**
        $qpts[q].append((i, j))$;
      end
    end
  end
end

Algorithm 1: CDF calculation.

and y from the previous section. The input to the routine is a data object that represents the density estimate. The object supports returning the maximum and minimum values for u and v and the density for given extents. The routine is shown in algorithm 1.

$quantiles$ is the set of quantiles. $dobj$ is the density object. $qpts$ is a dictionary of point lists, whose key is a quantile from $quantiles$ and whose value is a list of points. Each point is a (u, v) pair on the corresponding quantile within a tolerance TOL. TOL=$1 \div 2|quantiles|$, where $|quantiles|$ is the cardinality of $quantiles$. The in-

tervals from $dobj.min\_u$ to $dobj.max\_u$, and from $dobj.min\_v$ to $dobj.max\_v$, are both divided evenly by DIV, the number of divisions along each dimension. DIV can be tuned for desired resolution and CPU timings. We show our choice for DIV in table 3.1 under *integration mesh size*.

Data: $qpts$, $quantiles$
Result: $qcurves$
initialize $qcurves$;
foreach q *in* $quantiles$ **do**
  $cobj =$ interpolate curve for all pts in $qpts[q]$;
  foreach u *in evenly spaced NUM_PTS over interval* $[qpts[q][0], qpts[q][index\ at\ list\ length-1]]$ **do**
    $qcurves[q].append((u, cobj(u))$;
  end
end

Algorithm 2: Parameterize quantiles.

For the *parameterize quantiles* stage (see algorithm 2), we iterate through each member of $qpts$ and interpolate each individual curve using a curve object $cobj$, that can later be evaluated to obtain any v indexed by u.

This routine returns *qcurves*, a list of points from a parameterization of a curve represented by $cobj$. We approximate the parameter t_i in Eq. 6.11 by evenly dividing the entire interval of a quantile curve from an ortho-projection onto the u axis by NUM_PTS (the number of points chosen for parameterization). We then evaluate the $cobj$ from this interval of u values. We assume that the quantile curves are monotonically increasing over the interval.

For *interpolate quantiles* see algorithm 3. This routine loops through all members of $quantiles$ and interpolates each parameterized point between corresponding quantiles.

Data: $dobj0$, $dobj1$, $dobj2$, $dobj3$, $qcurves0$, $qcurves1$, $qcurves2$, $qcurves3$, α, β, $quantiles$
Result: $ipdf$
initialize $iqcurves01$;
initialize $iqcurves23$;
initialize $ipdf$;
foreach *q in quantiles* **do**
  foreach *idx in qcurves0*$[q]$ **do**
    $vec01 = qcurves1[q][idx] - qcurves0[q][idx]$;
    $iqcurves01[q].append(\alpha * vec01)$;
  end
  foreach *idx in qcurves2*$[q]$ **do**
    $vec23 = qcurves3[q][idx] - qcurves2[q][idx]$;
    $iqcurves23[q].append(\alpha * vec23)$;
  end
  foreach *idx in qcurves01*$[q]$ **do**
    $vec = qcurves23[q][idx] - qcurves01[q][idx]$;
    $ipt = \beta * vec$;
    $idens$ = evalPDF($dobj0$,$dobj1$,$dobj2$,$dobj3$,ipt,α,β);
    $ipdf.append((ipt.u, ipt.v, idens))$;
  end
end

Algorithm 3: Bilinear interpolation of quantile curves.

iqcurves store the interpolated points for interpolant quantiles. Using Eq. 3.10, we calculate $vec01$ and $vec23$. vec follows in a similar fashion for β. $dobj0$, $dobj1$, $dobj2$ and $dobj3$ are the density objects associated with each unit cell vertex in Fig. 5.1. $ipdf$ is returned and is a list of surface points on the interpolated PDF.

The *evalutate PDF values* stage is a direct calculation using Eq. 3.13, invoked during *interpolate quantiles* as the method *evalPDF*.

For the final *reconstruct PDF* step, a reconstruction of the PDF surface is performed using a suitable interpolation such as those available in SciPy [27] for irregular grid data. In this study, we tessellate the input point set to three-dimensional simplices, and

interpolate linearly on each simplex.

3.4 Results

Our implementation was written in Python, utilizing the SciPy package. All the computations were performed on the CPU. The computer system used for running the experiments was an Intel Core i7-3930k with 32 GB of RAM.

3.4.1 Synthetic Data

We construct a toy example consisting of a unimodal and bimodal distribution. Our mean parameter(s) for the 2D PDFs are the mean vector $\mu_i = [u, v]^T$, where u and v are the components aligned with the Cartesian x-y coordinate system. Spherical covariance matrices are used, i.e. the covariance matrix designation is a multiple of the identity matrix. The number of samples drawn from each distribution is 600 when estimating the PDF for interpolation.

The unimodal distribution is defined as:

$$\mathcal{N}_1(\mu_1, \Sigma_1), \mu_1 = \begin{bmatrix} 0 \\ \\ -1 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1 & 0 \\ \\ 0 & 1 \end{bmatrix} \tag{3.17}$$

The bimodal distribution is the sum of two bivariate normals, where the first is weighted 0.6 and the second is weighted 0.4:

$$\mathcal{N}_3(\mu_3,\Sigma_3), \mu_3 = \begin{bmatrix} 2 \\ \\ 1 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1 & 0 \\ \\ 0 & 1 \end{bmatrix} \tag{3.18}$$

$$\mathcal{N}_4(\mu_4,\Sigma_4), \mu_4 = \begin{bmatrix} -2 \\ \\ -1 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 1.5 & 0 \\ \\ 0 & 1.5 \end{bmatrix} \tag{3.19}$$

The parameters used for interpolation are in table 3.1. See table 3.2 for CPU timings. The results of interpolating between the synthetic PDFs are shown in Fig. 3.3.

Integration mesh size	200 x 200
Number of quantiles	≤ 100
Quantile curve interpolation	Linear
Number of points per quantile	150
PDF surface interpolation	Linear simplicial

Table 3.1: Parameters used for *bivariate quantile interpolation*.

CDF calculation	139.92
Quantile curve parameterization	0.02
Quantile curve interpolation	1.17
Interpolant PDF evaluation	6.22
PDF surface reconstruction	1.18

Table 3.2: Average CPU timings (in seconds) in toy example.

3.4.2 Application

Our ensemble data set covers a region of the Massachusetts Bay on the east coast of the United States of America [39] and is provided by Dr. Lermusiaux from MIT. The Massachusetts Bay volume in the study was divided into 53 x 90 grid with 16

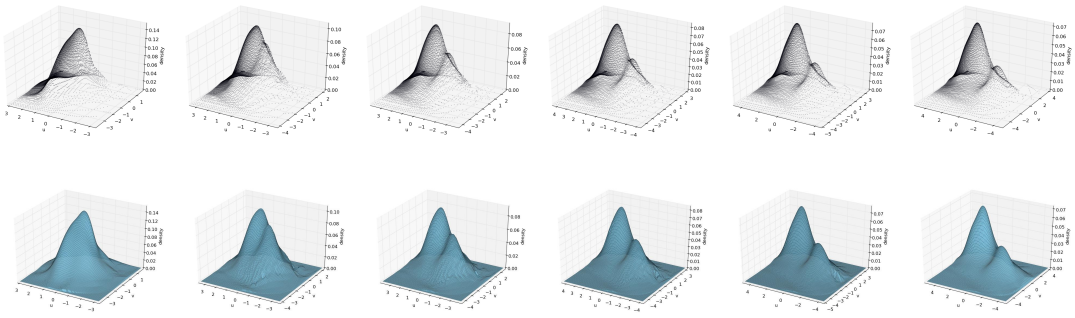


Figure 3.3: Interpolation from left ($\alpha = 0.0$) to right ($\alpha = 1.0$). Top row without surface interpolation. Bottom row with surface interpolation.

depths. The depths at these 53 x 90 grid points vary significantly: depths as shallow as 90 meters and as deep as 196 meters were recorded. Our data is representative the of environmental studies discussed in [36, 35].

We apply bivariate quantile interpolation to selected grid points over the spatial domain. We sub-sample at a quarter of the resolution of the original data, and keep the hidden data points as the "known" distribution to compare against our interpolants at $\alpha = 0.5$. For velocity fields, it is possible to interpolate over the temporal-domain as well. For instance, one could choose the same grid point but two different time-steps. Additionally, it is possible to interpolate over space and time. The interpolation is general and applicable to multiple scenarios. However, in this study, we show interpolation between velocity PDF separated by space for the same value of time.

We choose two pairs of representative examples from the data for velocity. The first pair is an interpolation well within the boundaries of the data set (at a depth of 90 meters). The second pair is an interpolation that includes multimodal distributions but is along the boundary of the data set (at the same depth level). These interpolations are

shown in Fig. 3.4 and Fig. 3.5 and referred to pair 1 and pair 2 in tables 3.3 and 3.4.

A third pair of PDFs use a vector of temperature and salt concentration (see Fig. 3.6 and tables 3.3, 3.4). The interpolation was performed at the same spatial location as the first pair of PDFs. These variables were tested for correlation using the Spearman rank-order correlation coefficient and the p-value to test for non-correlation [88]. For our data these are $\rho = -0.3093$ and p-value$= 8.946 \times 10^{-15}$.

Our metric for the variation between an interpolant and the known grid point density estimate is Earth Mover's Distance (EMD). EMD is a linear optimization initially developed for supply-demand transportation. EMD minimizes the cost of transforming one PDF into another by moving mass from one PDF to the other [72]. The transformation cost between two PDF P and Q is expressed by the following formulation:

$$EMD(P,Q) = min_{\{F=f_{ij}\}} \frac{\Sigma_{i,j} f_{ij} d_{ij}}{\Sigma_{i,j} f_{ij}} \tag{3.20}$$

where d_{ij} is a pre-defined ground distance between supplier i and consumer j, and $F = f_{ij}$ is a set of flows which defines the amount of mass transported from supplier i to consumer j. We use OpenCV's implementation of EMD, with the L2-distance parameter [5].

The EMD measured for our interpolation examples are listed in table 3.3. CPU performance is listed in table 3.4.

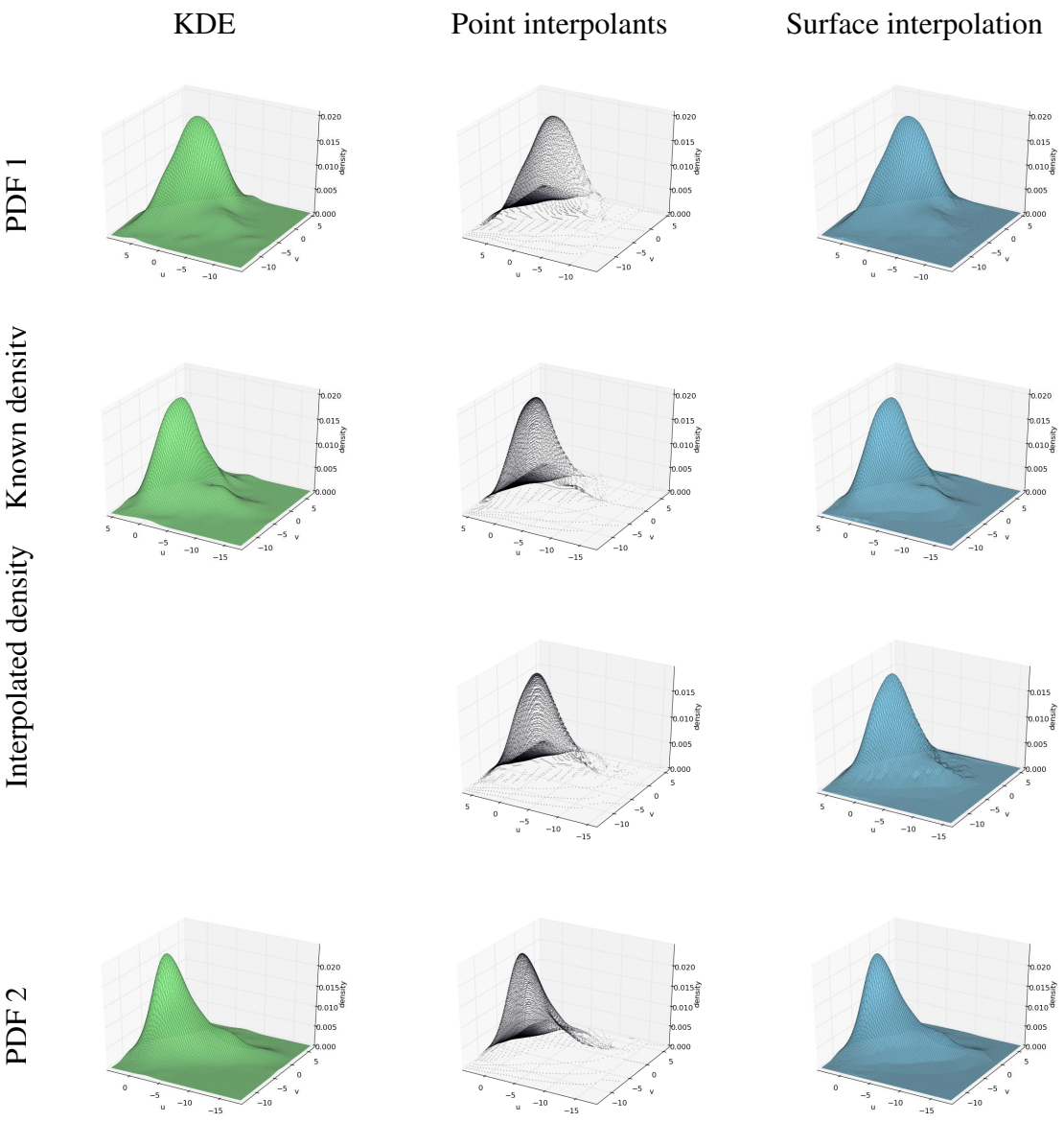


Figure 3.4: Pair 1 for simulation data using velocity components. Green distributions represent KDEs at grid points in data set. Blue distributions represent results of interpolation. The top row (PDF 1) and bottom row (PDF 2) contain the known distributions used for interpolation. We compare the second row density estimate with the third row containing the interpolant density.

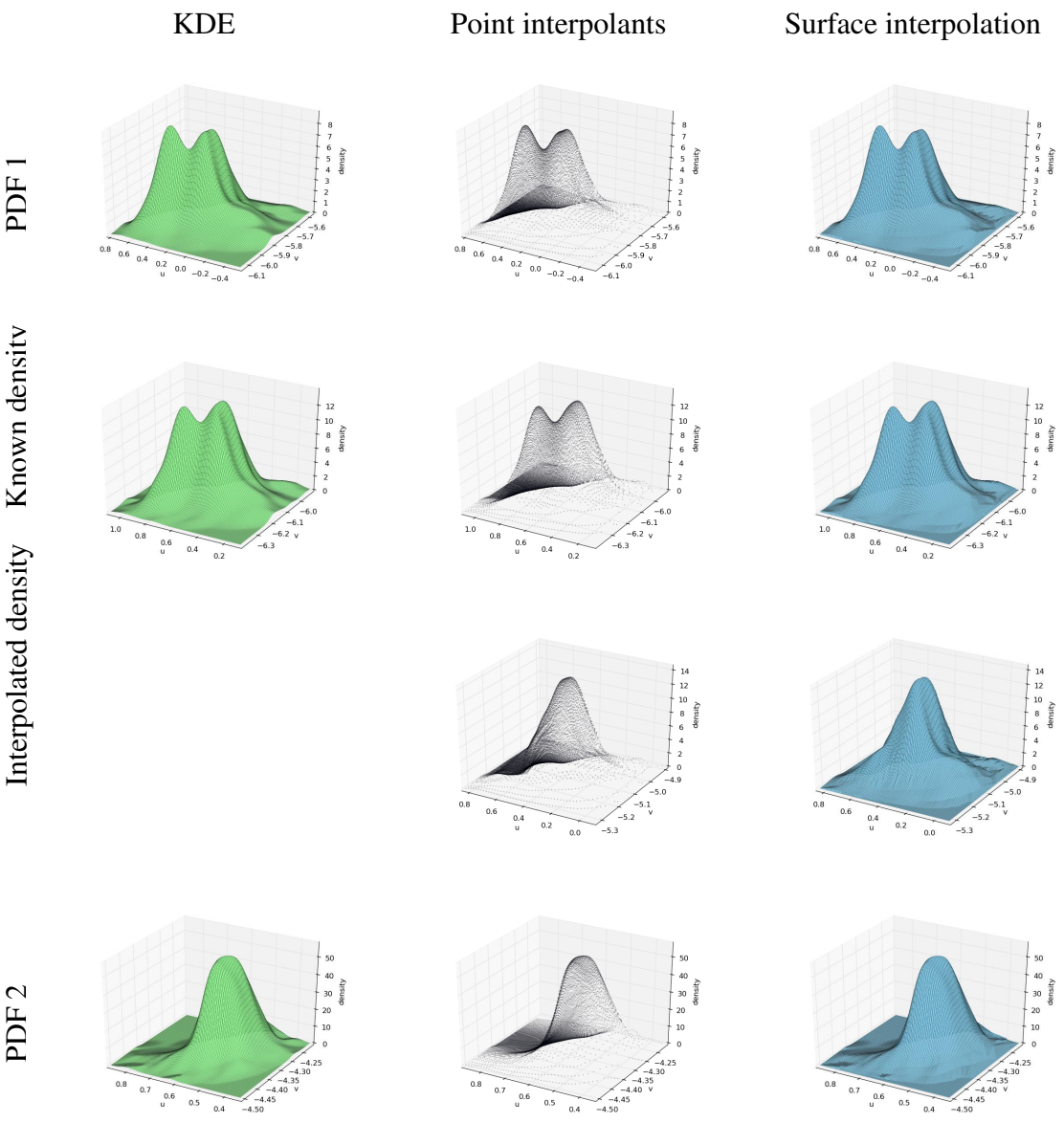


Figure 3.5: Pair 2 for simulation data using velocity components. Green distributions represent KDEs at grid points in data set. Blue distributions represent results of interpolation. The top row (PDF 1) and bottom row (PDF 2) contain the known distributions used for interpolation. We compare the second row density estimate with the third row containing the interpolant density.

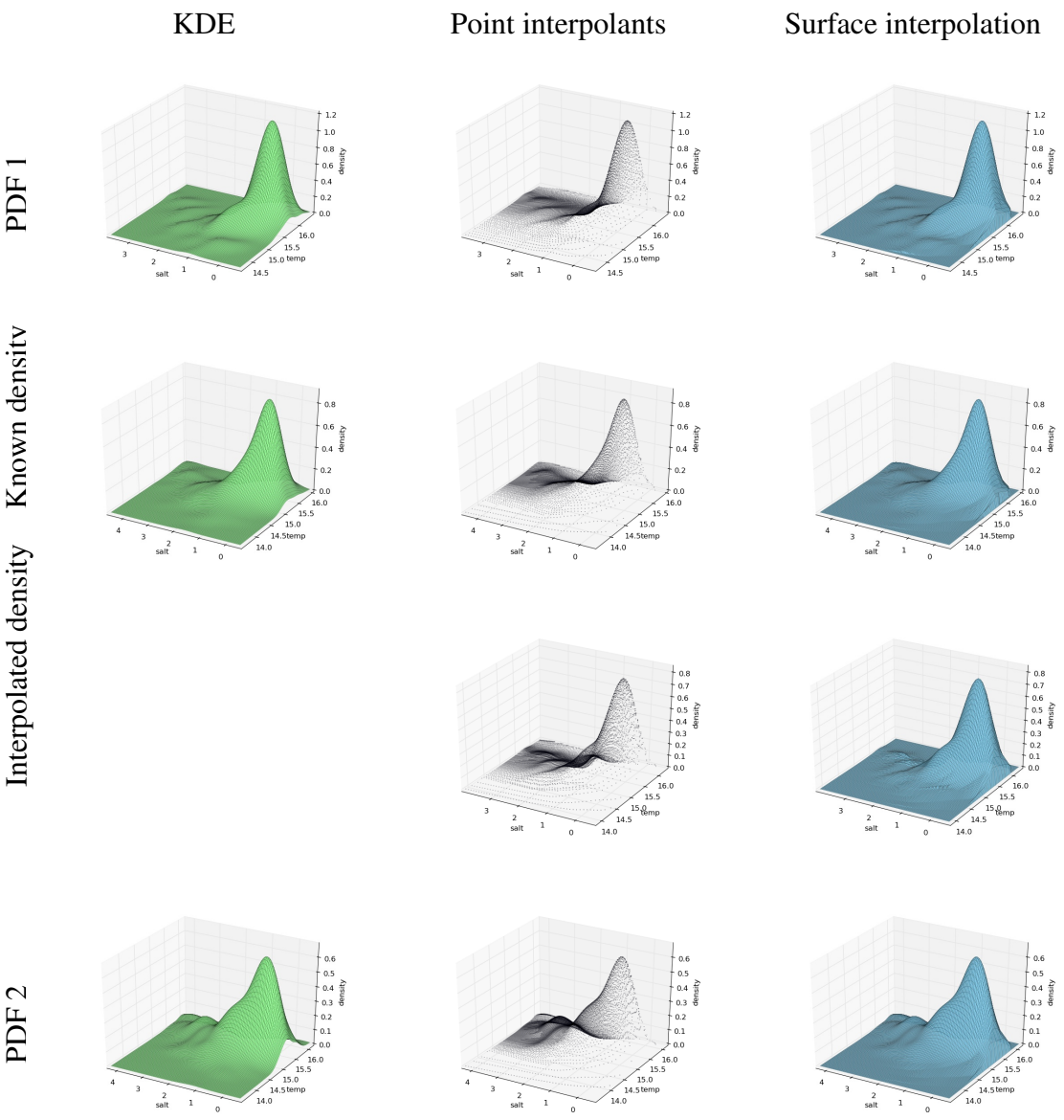


Figure 3.6: Pair 3 for simulation data using temperature and salt concentration. Green distributions represent KDEs at grid points in data set. Blue distributions represent results of interpolation. The top row (PDF 1) and bottom row (PDF 2) contain the known distributions used for interpolation. We compare the second row density estimate with the third row containing the interpolant density.

Distribution	Pair 1	Pair 2	Pair 3
PDF 1	0.397509	0.473298	1.040877
Know PDF	0.317712	0.290578	0.351446
Interpolated PDF	1.362007	3.958944	5.588929
PDF 2	0.542557	0.729264	0.568048

Table 3.3: Earth mover's distance measurements for simulation data shown in Fig. 3.4, Fig. 3.5, and Fig. 3.6. We compute EMD for the interpolant at $\alpha = 0.0$ in the entries of the row labeled PDF 1. Similarly, we compute EMD at $\alpha = 1.0$ in the row labeled PDF 2.

Algorithm Stage	Pair 1	Pair 2	Pair 3
Avg. CDF calculation per PDF	154.34	135.88	84.403
Parameterization	0.010	0.010	0.017
Curve interpolation	1.29	1.28	0.679
Interpolant PDF evaluation	6.17	6.09	5.678
PDF surface reconstruction	1.14	1.31	0.979

Table 3.4: CPU timings (in seconds) for simulation data.

3.5 Discussion

EMD values measured in table 3.3 show good results for pair 1, but the EMD measurement is higher for pair 2. The densities in pair 1 follow a smoother transition, while we miss the bimodal distribution in the second example. Minimal EMD difference is measured for surface interpolation alone in both cases. Pair 3 has slightly higher EMD values overall, while there is still good agreement with the known and interpolated distributions. This discrepancy is likely due to the slight clipping of the KDE (range values) versus the fill value of zero for surface interpolation.

It was also found that the number of samples in a density estimate increase CPU time for CDF calculation. This is due to the underlying implementation of SciPy and is not addressed in this chapter.

Various increases in efficiency could be gained by porting the density object implementation to compiled code. Interpolating points from quantile curves in parallel on the GPU is another possible way to decrease execution time. An implementation may also be chosen to store quantile curve calculations for increased interpolation efficiency.

Note that the number of quantiles may be less than or equal to the number requested, as shown in table 3.1. A chosen CDF integration mesh resolution is not always sufficient to capture the requested number of quantiles for a given distribution. Our implementation uses a fixed CDF integration mesh resolution.

Relaxing the assumption that quantile curves monotonically increase might allow better interpolation for cases where this is not always true. However, most distributions that we study have densities where this assumption is valid. In any case, this generality in the algorithm would increase execution time.

The interpolation methods presented in this chapter do not account for spatial covariance with surrounding grid point distributions. We interpolate unique surface values of individual PDFs which do not relate as a whole to surrounding PDFs when considered in isolation.

Interpolation is inherently ill-posed. The quality of the interpolants are dependent on the smoothness of the underlying field. Therefore, procedures for measuring the smoothness of ensemble data sets are important here, but also for calculating probabilistic gradient fields. Such gradients are not easily defined for ensembles using finite difference. In any case, such analysis constitutes further study.

3.6 Conclusion

We presented a direct extension of 1D PDF interpolation using quantile interpolation for the bivariate case. This interpolation is useful for interpolating within random fields whose components are inseparable, such as 2D velocity PDF and other correlated random variables. Further studies of visualization using interpolation will be facilitated by such interpolation. Under circumstances where multiple scalar fields are interpolated, a univariate approach is best for performance and to reduce over-smoothing in density estimation [81] if the fields are uncorrelated.

While *Gaussian Mixture Model Interpolation* is ambiguous with respect to its pairwise interpolants, we have provided a better alternative. For 2D vector fields, *Bivariate Quantile Interpolation* is faster than *Displacement Interpolation* and can be more easily implemented.

Chapter 4

Applications of Non-Gaussian Density Estimates

A typical assumption is that ensemble data at each spatial location follows a Gaussian distribution. We investigate the consequences of that assumption when distributions are non-Gaussian. A sufficiently acceptable interpolation scheme needs to be addressed for the interpolation of non-Gaussian distributions. We present two methods to calculate interpolations between two arbitrary distributions and compare them against two baseline methods. The first method uses a Gaussian Mixture Model (GMM) to represent distributions. The second method is a non-parametric approach that interpolates between quantiles in the cumulative distribution functions. The baseline methods for comparison purposes are: (a) using a Gaussian representation and interpolating the means and standard deviations, and (b) forming a new distribution based on the inter-

polation of individual realizations of the ensemble. We show that the two proposed non-Gaussian interpolation methods have the following behavior: the interpolated distributions do not decompose to more constituent Gaussian distributions than the highest modality of those being interpolated, and do not have variances less than the smallest variance from the grid points being interpolated. Finally, we compare these four interpolation methods when used in the analysis of scalar and vector fields of ensemble data sets, particularly in areas where the distribution is non-Gaussian.

4.1 Introduction

A fundamental operation used in most visualization algorithms is interpolation. Interpolation is used in workhorse visualization techniques such as marching cubes, direct volume rendering, and streamline generation, and many other popular algorithms. Performing interpolation is well defined when the data points and the interpolants are single valued, or crisp. However, this is not the case when the data points and the interpolants are multivalued, or consist of a distribution.

With increasing interest in representing uncertainty in modeling and simulation with techniques based on Monte Carlo methods, we are now faced with the challenge of analyzing and visualizing ensemble fields. Ensemble fields are made up of individual realizations, each a possible outcome, of the simulation. Assuming that the ensemble fields are defined over a grid, a popular approach is to treat all the values at a given grid point from different realizations as a multivalue or a distribution. Recent works

in this area have primarily assumed that the multivalue follow a Gaussian distribution. Even more recent efforts have tried to remove this assumption. In this chapter, we examine two alternative interpolation methods that support non-Gaussian distributions and compare them against two other baseline methods.

There are several reasons for considering a more general representation for multivalue aside from a Gaussian model. The assumption of a normal distribution neglects the possibility that the multivalue represents overlapping sub-populations of data, which by themselves can be considered Gaussian component distributions. These often arise in various situations such as sub-voxel material classification for volume rendering, and ambiguity in resolving fiber orientation during DT-MRI tractography. Often times, it is at these "mixing" regions where interesting things happen e.g. presence or absence of a boundary, crossing or divergence of a path, etc. The distributions at these regions exhibit multimodal profiles. Their consideration requires representation of these distributions as non-Gaussian.

In this chapter, we adopt the terms crisp to mean single valued, whereas multivalued is taken to mean a collection of values. The concept of multivalues is general enough to represent (i) the collection of values of a variable at a particular location as reported by different realizations in an ensemble, (ii) a probability distribution of the same set of values represented as a probability density function (PDF) that requires the area under the function to sum to one, and (iii) other representations e.g. as a signal. Using the operator based approach for manipulating multivalues linear interpolations can be

defined as:

$$M' = (1-\alpha)M_1 + \alpha M_2 \tag{4.1}$$

where M', M_1 and M_2 are multivalues, $\alpha \in [0,1]$. Note that $(1-\alpha)$ is a simple subtraction between 2 crisp values. The multiplication of a crisp value and a multivalue simply scales each member of the multivalue and results in a multivalue. On the other hand, the $+$ operator between two multivalues can be defined according to the needs of the application. Using this framework, one can also define and entertain other variations of simple linear interpolations e.g.

$$f(M') = (1-\alpha)f(M_1) + \alpha f(M_2) \tag{4.2}$$

where $f(.)$ operates on multivalue M, and $+$ is appropriately defined.

The two interpolation methods examined in this chapter define $f(.)$ as: (i) a gaussian mixture model to represent M, and (ii) different quantiles of the PDF representing M. We refer to interpolation using method (i) as *GMM PDF interpolation*, and method (ii) as *Quantile PDF interpolation*. These are described later in section 6.4 and Quantile PDF interpolation was the subject of chapter 3. The two baseline methods used to compare these interpolations are: (i) one that uses a Gaussian representation of M – interpolation is referred to as *Gaussian PDF interpolation*, and (ii) one that uses the raw multivalues – interpolation is referred to as *Ensemble PDF interpolation*.

There are three main considerations in formulating the interpolation methods. Firstly,

if additional modes are introduced during interpolation, this would imply that new sub-populations are somehow introduced during the process. While such populations may exist, there is nothing in the data set to suggest this. So, we impose the condition that the interpolation method cannot create additional modalities between known distributions. Secondly, a suitable interpolation method should not produce distributions that have variance less than the smallest variance from the grid points being interpolated between. As a contradiction, suppose that the interpolated distributions did in fact have variances less than those at the grid points. This is undesirable since the interpolated distributions should be less certain than at the observed grid point distributions, and should therefore not have variances that are smaller than those observed at the grid points. Thirdly, the method must naturally produce a total probability of 1.0. While one approach is to normalize the sum of components treated separately, we present more than one possible method that adheres to our specifications and that does not require explicit normalization. Therefore, a good interpolation method should ensure that: (i) no additional modes are introduced during the interpolation, (ii) the variance should not be smaller during interpolation, and (iii) interpolated results are also probability distributions.

4.2 Related Work

A nice overview of statistical techniques for spatial interpolation was presented by Myers [45]. The techniques range from simple linear models with no covariance,

to those using spatial structure functions. The survey however does not include non-parametric distribution interpolation. The paper does claim that interpolation is a solution to an inherently ill-posed problem, namely that it is a problem of prediction with limited data. For that, multiple models with different purposes can be employed. A more detailed survey, but focusing on geostatistical applications, compare methods according to different criteria such as local vs global support, deterministic vs stochastic, univariate vs multivariate, linear vs nonlinear etc. Among the methods that consider stochastic data, they assume normal distribution.

Within the visualization community, there are also a number of recent publications that address stochastic interpolation. Scheuermann, et al. [78] present a form of Kriging interpolation of spatial data for Gaussian distributions using a parameter-based approach. This technique relies on computing a covariance matrix and that the underlying data be formed from a Gaussian process. Pfaffelmoser et al. [56] visualize isosurfaces via a raycasting scheme, and perform spatial interpolation assuming the data has a Gaussian distribution at each location. Likewise, Pöthkow et al. [61] discuss isocontour visualization of normally distributed data. They interpolated between grid points using the 0th and 1st moments without spatial correlation considerations. Their subsequent work [63] considered the effects of spatial correlation in visualizing isosurfaces using probabilistic marching cubes. An alternative method of looking at global correlation structures in a hierarchical fashion was presented in [57].

When data do not follow a Gaussian distribution, a more general uncertainty model

is needed. Liu et al. [38] propose a Gaussian mixture to represent the distribution of voxel values in air temperature data. They perform volume rendering on the data set and interpolate between pairs of a fixed number of Gaussians components along cast rays. In their study, they found that four Gaussian kernels are sufficient for a variety of data sets that they examined. For non-parametric representations of non-Gaussian distributions, operations on the distributions require different handling. Love, et al. [39] discuss two forms of a non-parametric interpolation method via convolution addition of probability distributions as well as bin-wise addition. Pohl, et al. [60] first transform the (discrete) distribution to Euclidean space via a set of Log Odds operations, where they can then be manipulated using conventional addition and multiplication. Results are then mapped back to probabilistic space via a reversible transform. Read [68] delineates a method to interpolate histograms via quantiles.

Uncertainty in vector fields is of great interest to at least two broad fields: meteorological community and fiber tracking community. Most of the work to date assumes Gaussian random fields. Otto, et al. present analysis of 2D [48] and 3D velocity fields [49] using particle advection, critical points, and segmentation of field topology. Petz et al. [55] also analyze uncertain velocity fields modeled as Gaussian random fields with spatial correlation.

There is a growing body of work on probabilistic fiber tracking. Unlike velocity fields, the tracks here represent fiber connectivity from one region to another and are obtained by integrating the major eigenvector field. The main source of uncertainty

can be attributed to inadequate resolution in the data acquisition stage of diffusion tensor MRI. However, there are numerous other sources as well [6]. While most of the earlier works on probabilistic fiber tracking delved on the inadequacy of the simple tensor representation to show alternative trajectories due to multiple fiber populations within a cell, more recent works are based on high angular resolution diffusion imaging (HARDI) data which makes it is possible to describe fiber orientations using more sophisticated formulations such as spherical harmonics and multi-tensor representations. In a recent paper, Jiao et al. [26] describe a local, icon-based presentation of an ensemble field of fiber orientation distribution functions (ODF). The results of our work can be used towards spatial analysis of such ensemble fields, for example.

There is much interest in the meteorological communityto provide better visualization of forecast data. Slingsby et al. [85], discuss how users interpret and use weather data, specifically hurricane data. Storm path information are examined from historical data. They draw attention to spatial and temporal clustering and its undervalued status among those currently employing such visualization software. Weather forecasts are usually based on an ensemble of predictions. For that, Potter et al. [64] describe a framework for viewing stochastic information from ensembles. This package allows for visualization of spaghetti plotting, etc. of weather data. Zhang, et al. [75] present Noodles, a software package for displaying uncertainty in streamlines and other weather data visualization for ensemble forecasting. Potter et al. [65] describe a software tool to visualize two-dimensional sets of distribution data. It displays a contour of field PDF

values and allows for a normed difference between data PDFs and an ansatz selected by the user. More recently, Phadke et al. [58] present two novel visualization methods for ensembles. Primarily, they allow simultaneous viewing of multiple ensemble members. They also present a technique called "Screen Door Tinting" which applies value changes to field points that show differences between ensembles.

From the point of view of users, Martin et al. [42] point out the difficulty of users to identify hurricane directional movement and speed from current data visualization, or directly on vector fields. In a similar study, Broad et al. [7], further emphasize interpretation and usage of complex weather data. They show how a general interpretation of a Gaussian distribution of hurricane direction prediction can lead to inaccurate views on the probability within a "cone of uncertainty." Clearly, if multimodal velocity distribution is calculated with such a broad region of uncertainty using a Gaussian assumption, incorrect estimation of the probability of hurricane direction can occur, most specifically within the general population who can be greatly impacted by such interpretation. A non-Gaussian consideration for vector field visualization together with a redesigned visualization may rectify this issue to a degree. We hope that with the results presented in this chapter, we will be able to extend such visualizations to consider non-Gaussian mixing regions.

4.3 Gaussian Interpolation

In this section, we briefly summarize alternative strategies of performing spatial interpolation for distributions that are assumed to be Gaussian. In this discussion, we consider linear interpolation between two univariate Gaussian distributions. The interpolation parameter α indicates both the parameterized spatial distance and the parametric interpolation distance between the two distributions.

First, it is possible to interpolate Gaussian parameters: the mean, standard deviation (and other moments) independently. The interpolants remain Gaussian and can be reconstructed based on interpolated parameters. This method is simple yet allows for smooth translation of mode and smoothly varying moments as can be seen in Fig. 4.1. *Gaussian PDF interpolation* in this work refers to this variant of Gaussian interpolation.

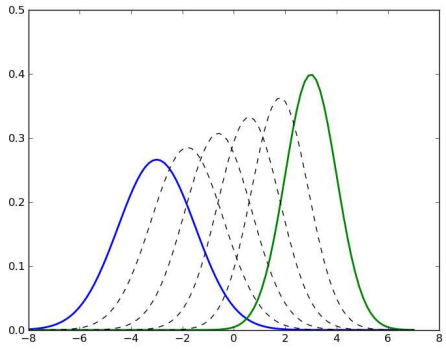


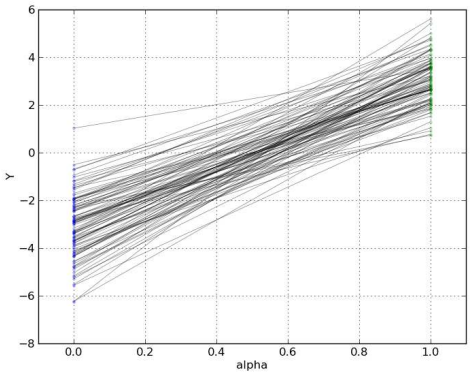
Figure 4.1: Intermediate interpolants (black dashed curves) travel from the blue to the green Gaussian curve.

When the distribution is represented by samples rather than by Gaussian parameters, another approach is to interpolate the samples directly rather than fitting it with a

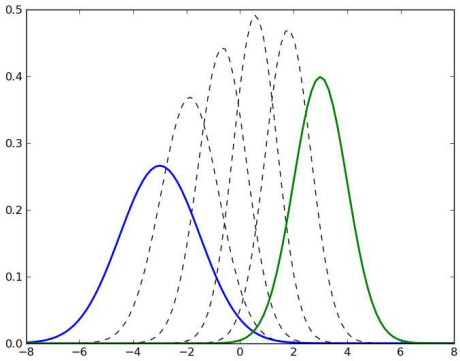
Gaussian first. Here, samples drawn from each distribution are interpolated independently. For a random variable B (representing samples drawn from the blue curve), let the random sample $Y_1, Y_2, ..., Y_n$ be n independent and identically distributed (i.i.d.) variables. Similarly, a random sample from G (representing samples drawn from the green curve) are the n i.i.d. variables $Y_{1+n}, Y_{2+n}, ..., Y_{2n}$. The total number of all possible sample interpolants is the count of all possible pairings between the members of the random samples, i.e. the cardinality of the Cartesian product: $|\{Y_1, Y_2, ..., Y_n\} \times \{Y_{1+n}, Y_{2+n}, ..., Y_{2n}\}|$, for any given $\alpha \in [0, 1]$. This method of PDF interpolation allows translation of mode but variance is potentially less than either the B or G distribution during interpolation. Figure 4.2 shows an instance of sample pairings between two PDFs and the resulting PDF interpolants. In this example, there are interpolants that have variance less than the distributions being interpolated.

Thirdly, there is "probabilistic interpolation," also referred to as histogram interpolation. This method normalizes the range of the grid point distributions. For each "bin," frequencies are interpolated. With this approach, the PDF at one grid point morphs into the the PDF at the other grid point. In Fig. 4.3, the interpolant at $\alpha = 0.5$ is bimodal.

This third method might be suitable for some applications, such as volume rendering materials where a cell might contain multiple materials. That is, when one considers the situation where the populations are predominantly of different types on either side of a boundary, but is made up of both populations at the boundary region, then interpolations that increase the modality of the distributions might be desirable. On the



(a) One set of sample pairs drawn independently from the distribution on the left (blue dots) and the distribution on the right (green dots).



(b) Intermediate interpolants (black dashed curves) show smaller variance than end points distributions.

Figure 4.2: Sample interpolation for a given instance of distribution sample pairings. (a) Shows pairings and (b) depicts interpolants with dashed lines.

other hand, when one considers the transport or transition of a population or mixture of populations e.g. volume of water at different temperatures, across some distance then we do not want to increase the modality of the interpolant distributions. In this work, we consider the latter design criterion as we consider interpolation of non-Gaussian distributions.

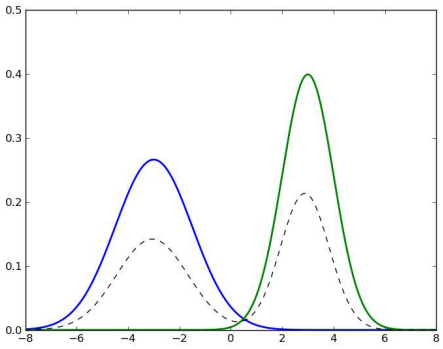


Figure 4.3: An interpolant can become multimodal between unimodal distributions as shown by the dashed black interpolant at $\alpha = 0.5$.

4.4 Non-Gaussian Interpolation

We present two techniques for the linear interpolation of PDFs as represented by a GMM and a non-parametric quantile model. These techniques directly apply to the standard unit reference cell, where each grid point represents a distribution from an ensemble.

4.4.1 GMM Interpolation

Our first approach is to linearly interpolate Gaussian parameters for a Gaussian Mixture Model (GMM) as outlined in Fig. 4.4. The final step may be optional depending on the application, as indicated by the dotted arrow and box. We describe fitting components and interpolating parameters in this section. Gathering samples is implementation specific and is influenced by the data source.

The *fit components* stage from Fig. 4.4 requires modeling the samples with Gaussian components.The GMM can be extracted using the Expectation-maximization (EM) al-

gorithm [1, 2, 67] in order to derive a mixture from the starting samples using m Gaussian components. The mixture is denoted as the random variable $V_{\mathbf{g}}$ located at grid point location $\mathbf{g}$, where $\mathbf{g} \subset \{\mathbf{p} | \mathbf{p} \subset \mathbb{R}^n, n \in \mathbb{N}, 0 < n \leq 3\}$. The GMM is determined by a linear combination of Gaussian basis functions Φ_i:

$$V_{\mathbf{g}} = \sum_{i=1}^{m} a_i \Phi_i \tag{4.3}$$

$$\sum_{i=1}^{m} a_i = 1 \tag{4.4}$$

$$\Phi_i = \mathcal{N}(\mu_i, \sigma_i^2) \tag{4.5}$$

In the next stage of the method, *interpolate parameters*, we first determine the how to pair each Gaussian component from different grid point distributions. For the separate grid points $\mathbf{g}_0$ and $\mathbf{g}_1$, whose Euclidean norm $||\mathbf{g}_0 - \mathbf{g}_1|| = 1$, we pair corresponding Φ_i from V_0 and V_1 (located at $\mathbf{g}_0$ and $\mathbf{g}_1$, respectively). The pairing heuristic for Gaussian components between each end point is based on a one-dimensional linear scale. For univariates, in order to minimize interpolation distance between the mean of paired Gaussian components, we allow sub-steps in which a possible re-pairing ranked by sorted Gaussian means takes place. In the multivariate case, we pair and sort based on the weight of each Gaussian.

We calculate α, and the interpolant Gaussian component parameters: $\bar{\mu}_i$, $\bar{\sigma}_i^2$ and their associated weights $\bar{a}_i$ using Eqs. 4.6 through 4.9. Another index is used for each

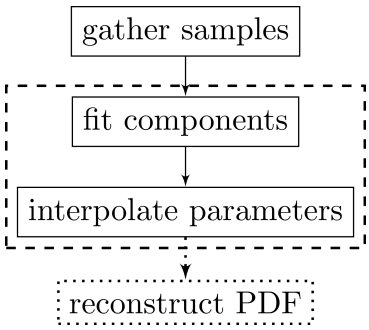


Figure 4.4: Gaussian Mixture Model interpolation method. Dashed outline signifies core method stages primarily discussed in this chapter. Dotted arrow and box signify optional stage.

component to denote which $V_{\mathbf{g}}$ it is from. Therefore, we have $\mu_{0,i}$, $\sigma^2_{0,i}$ and $a_{0,i}$ from V_0. $\mu_{1,i}$, $\sigma^2_{1,i}$ and $a_{1,i}$ are from V_1.

$$\alpha = ||\mathbf{p} - \mathbf{g}_0|| \tag{4.6}$$

$$\bar{\mu}_i = (1-\alpha)\mu_{0,i} + \alpha\mu_{1,i} \tag{4.7}$$

$$\bar{\sigma}^2_i = (1-\alpha)\sigma^2_{0,i} + \alpha\sigma^2_{1,i} \tag{4.8}$$

$$\bar{a}_i = (1-\alpha)a_{0,i} + \alpha a_{1,i} \tag{4.9}$$

Thus, our interpolant PDF is $\bar{V}_{\mathbf{p}}$ at location $\mathbf{p}$, defined on a line segment of unit length and with end points $\mathbf{g}_0$ and $\mathbf{g}_1$.

This interpolation method meets our design criteria. Interpolant PDFs will not have greater modality than end point distributions since we require a constant number of Gaussian components to be interpolated. Therefore no additional modes can be present

in the interpolants. Linear interpolation of variances from components produce GMM interpolants whose component variances are bounded by those at the end points. Mean interpolation difference is minimized for univariates. Probability interpolation difference between components is minimized for multivariates. The interpolated weights will always sum to one. This is ensured, as long as the total of the weights at every α equal one, as we require. Because EM only returns weights that sum to one, and we only make one-to-one pairings with a fixed and the same number of Gaussian components at each end point, then any number of re-pairings will also have total weights equal to one.

4.4.2 Quantile Interpolation

The quantile interpolation method overview is shown in Fig. 4.5. This method was first introduced in chapter 3.

Stages *gather samples* and *estimate density* are implementation specific. We do not cover their implementation details here and the user may choose varying approaches depending on the data. For example, kernel density estimation (KDE) with different window setting techniques can be used for density estimation.

During the *determine quantiles* stage, we compute the random value from the cumulative distribution function (CDF) that will return the desired quantile. The *interpolate quantiles* phase from Fig. 4.5 utilizes a linear interpolation between quantiles q_0 and q_1 of the cumulative density functions (CDF) of V_0 and V_1. This is expressed in Eq. 4.10

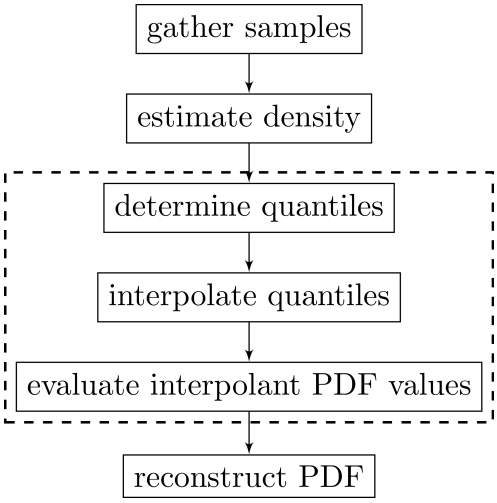


Figure 4.5: Quantile interpolation method. Dashed outline signifies core method stages discussed in the chapter.

and uses α from Eq. 4.6.

$$\bar{q} = (1-\alpha)q_0 + \alpha q_1 \tag{4.10}$$

In the *evaluate interpolant PDF values* step, both grid point distributions' quantiles evaluate to the same cumulative density of the interpolant CDF over the sample space variable s:

$$\int_{-\infty}^{\bar{q}} \bar{V}_{\mathbf{p}}(s)ds = \int_{-\infty}^{q_0} V_0(s)ds = \int_{-\infty}^{q_1} V_1(s)ds \tag{4.11}$$

Each interpolant probability value for the interpolant's qth quantile can be evaluated using the following expression (see [68] for a complete derivation):

$$\bar{V}_{\mathbf{p}}(\bar{q}) = \frac{V_0(q_0)V_1(q_1)}{(1-\alpha)V_1(q_1)+\alpha V_0(q_0)} \tag{4.12}$$

While we can find a unique random value to obtain a desired quantile for univariates, this is not true for the bivariate (or multivariate) case. For the bivariate case, the *determine quantiles* stage requires that we sum over the two-dimensional sample space of the PDF estimate in order to collect (u, v) sample pairs that correspond to the same cumulative density. We do this only at the end points $\mathbf{g}_0$ and $\mathbf{g}_1$. Note that integration of density is performed over a discretized grid and compared within a specified tolerance of the quantile value.

The result of the *determine quantile* step is a set of points that have the same quantile. These points form a curve which we parameterize and refer to as a quantile curve. In the *interpolate quantiles* stage, we take corresponding points (u_0, v_0) and (u_1, v_1) on the curves from $\mathbf{g}_0$ and $\mathbf{g}_1$, respectively and find $(\bar{u}, \bar{v})$ along a line between (u_0, v_0) and (u_1, v_1) depending on α. The resulting interpolant is obtained using Eq. 4.13.

$$\bar{V}_{\mathbf{p}}(\bar{u}, \bar{v}) = \frac{V_0(u_0, v_0)V_1(u_1, v_1)}{(1-\alpha)V_1(u_1, v_1)+\alpha V_0(u_0, v_0)} \tag{4.13}$$

For the final *reconstruct PDF* step, a reconstruction of the PDF curve or surface is performed using a suitable interpolation such as those available using [27]. For our study, we tessellate the input point set to n-dimensional simplices, and interpolate linearly on each simplex. Unlike the GMM method, PDF modes can only be estimated

with a continuous curve or surface. In the case of infinitely many interpolant PDF data points, the surface reconstruction approaches a true PDF.

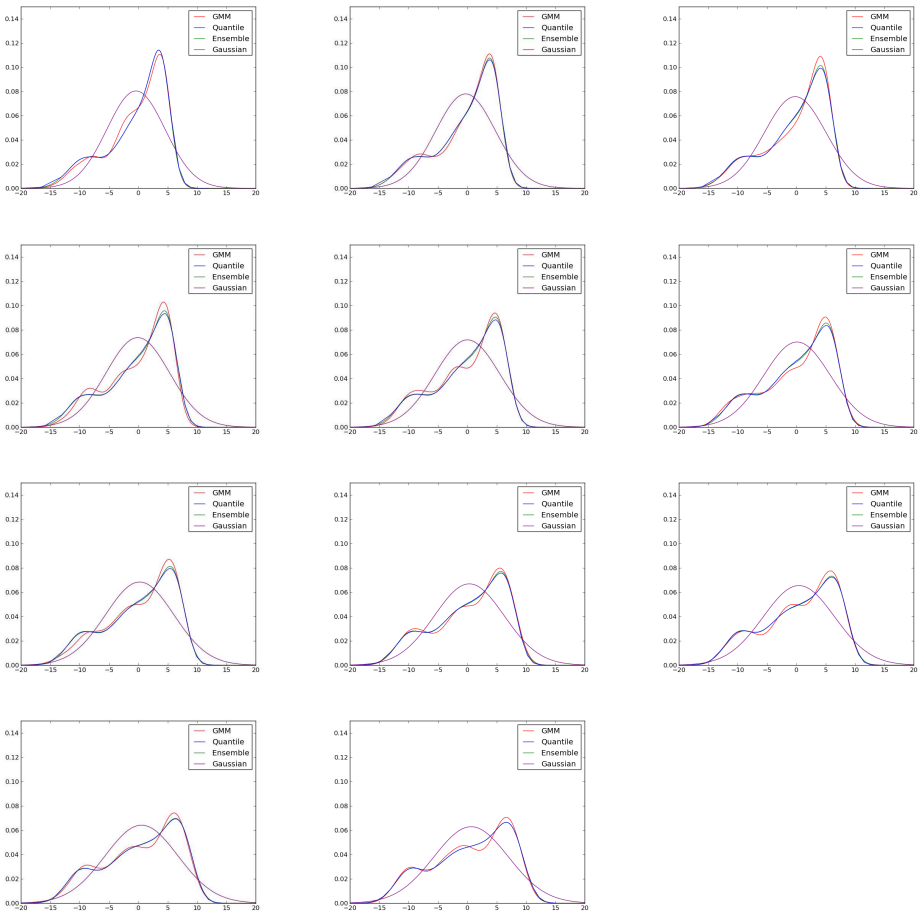


Figure 4.6: Univariate interpolant from α=0.0 to α=1.0: GMM (red), Quantile (blue), Ensemble (green) and Gaussian (purple).

Interpolant PDFs will not have greater modality than end point distributions. Inflection points on the CDFs will only split and merge corresponding to the modality at the end points. Linear interpolation of the quantiles ensures this. In order for additional modes to form at interpolants, quantiles would have to interpolate to values outside of the range set by the end point PDF quantile values during the interpolation. Since

this can not occur using linear interpolation, additional modes do not occur with this method.

Variance of the interpolants for Quantile interpolation is never greater than either end point distributions. The interpolants have quantiles located "between" the end point PDF quantiles in the associated sample space defined by the end point distributions. If the interpolated quantiles were to take on values outside of their bounds set by the end point PDFs, then the variance constraint would be violated. However, linear interpolation does not allow that to happen. It can also be shown that Quantile interpolation is similar to sample based interpolation discussed in section 4.3. The method interpolates paired samples based on *ordered* samples from both end point PDFs by cumulative density. In this way, no vertical cross-section of the interpolated samples has variance that is less than the least variance from either end point PDF in the interpolation.

4.5 Results

In the results below, we use four Gaussian components for GMM PDF interpolation as suggested by Liu et al. [38].

4.5.1 Ground "Truth" Comparison

We examine the behavior of our interpolation methods in Fig. 4.6 for a one-dimensional case between two non-Gaussian distributions. Six hundred samples are used to form a fixed-width kernel density estimate (FKDE [24]) at each end point. Our ground "truth"

is derived from a linear interpolation of realizations. We then form a non-parametric distribution of each ensemble member interpolant using FKDE.

Figure 4.6 qualitatively shows that both quantile and GMM PDF interpolations are quite similar to our ground truth ensemble PDF interpolation. On the other hand, the simple Gaussian PDF interpolation shows marked difference from our ground truth. To obtain a more quantitative measure, we calculate the symmetric Kullback-Leibler (SKL) divergence which gives us a measure of dissimilarity between two distributions. Eq. 4.14 is the SKL between probability distributions P and Q.

$$D_{\mathrm{SKL}}(P\|Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) P(i) + \sum_i \ln\left(\frac{Q(i)}{P(i)}\right) Q(i) \tag{4.14}$$

SKL is computed from $\alpha = 0.0$ to $\alpha = 1.0$ for each PDF interpolation method. For each method, we compute and average 100 such SKL comparisons to remove measurement noise due to sampling and EM fitting. Because the SKL results for Gaussian interpolants are an order of magnitude greater than both GMM and Quantile PDF interpolants, we show the Gaussian SKL measurements separately. In Fig. 4.7, we can easily see that Quantile interpolants (blue line) have the least SKL values, while both GMM (red line) and Gaussian (purple line) have larger entropies. The color scheme used for each PDF interpolation method in Figs. 4.6 and 4.7 are used for the remainder of this chapter.

Entropy at $\alpha = 0.0$ and $\alpha = 1.0$ are due entirely to the accuracy of the estimation and are not due to any of the interpolation methods. For intermediate α values, the SKL

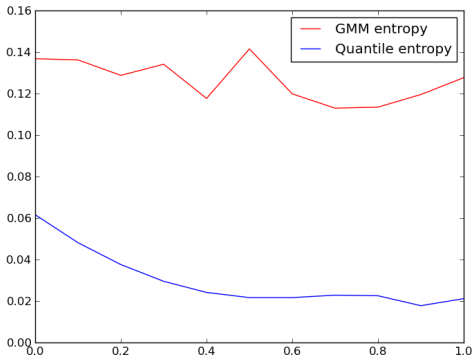
entropy is a combination of the entropy due to estimation errors and the entropy due to difference between the ensemble interpolant and the GMM, Quantile or Gaussian interpolant. Unfortunately, since density estimate and fitting of Gaussian components are needed to form the distributions at the end points, and we do not know how the estimation or fitting error varies as a function of α, we cannot distinguish between entropy due interpolation and those due to estimation or fitting.

Interestingly, as can be seen in Fig. 4.7 (b) for Gaussian interpolants, entropy at $\alpha = 1.0$ is less than any intermediate α. Quantile PDF interpolants are almost identical with ensemble interpolants and entropy is greatest at $\alpha = 0.0$ where estimation entropy is larger than for any interpolants. Quantile PDF interpolation effectively orders the samples by their cumulative probability. This corresponds closely with ensemble physical simulations per ensemble member.

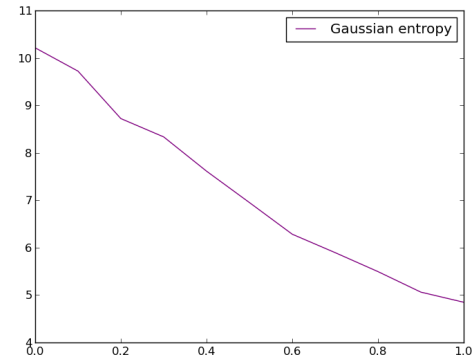
Figure 4.8 shows a linear interpolation between two bivariate distributions. At the top of the figure, we have a bimodal distribution and at the bottom of the figure, we have a unimodal distribution. Some tears on the interpolant PDF can be observed in column (b) due to insufficient data samples.

4.5.2 Synthetic Data

For covariant random variables, we describe interpolation in a synthetic velocity field where the velocity components are the bivariate random variables under consideration. In order to show the effect of considering a bivariate bimodal distribution when



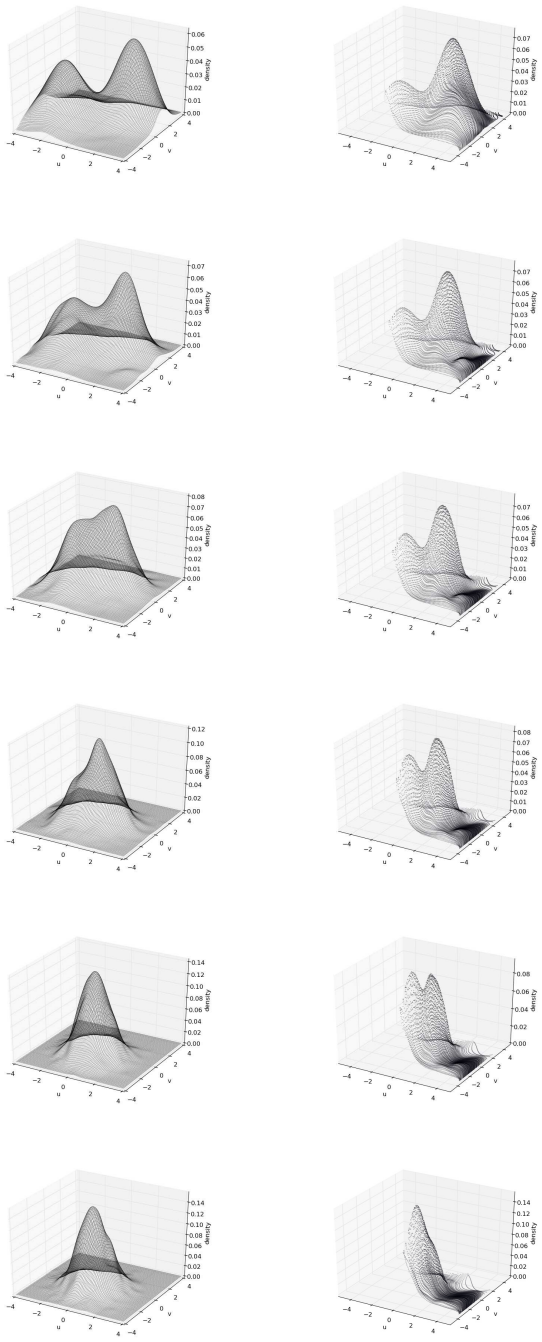
(a) GMM and Quantile SKL



(b) Gaussian SKL

Figure 4.7: Ten measurements of the SKL divergence for univariate interpolants from α=0.0 to α=1.0. Values are averaged from 100 independent comparisons. Entropy is shown on vertical axes and α on horizontal axes.

advecting in a velocity vector field, we construct a toy example consisting of a 3 x 3 grid where all grid points are defined as unimodal except the center grid point, which is defined by a bimodal distribution. Our mean parameter(s) for the velocity PDFs are the mean velocity vector $\mu_i = [u, v]^T$, where u and v are the velocity components aligned with the Cartesian x-y coordinate system. The left half and the top center of the grid is defined by a normal bivariate. Spherical covariance matrices are used, i.e. the covariance matrix designation is a multiple of the identity matrix.



(a) (b)

Figure 4.8: One-dimensional PDF interpolation using (a) GMM and (b) Quantile from a bimodal bivariate ($\alpha = 0.0$) at the top to a unimodal bivariate ($\alpha = 1.0$) at the bottom.

$$\mathcal{N}_1(\mu_1,\Sigma_1), \mu_1 = \begin{bmatrix} 0 \\ \\ -1 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1 & 0 \\ \\ 0 & 1 \end{bmatrix} \tag{4.15}$$

The right side of the grid is defined by:

$$\mathcal{N}_2(\mu_2,\Sigma_2), \mu_2 = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & 0 \\ \\ 0 & 1 \end{bmatrix} \tag{4.16}$$

And, the center grid point is the Gaussian mixture of the following two bivariate normals where the first is weighted 0.6 and the second is weighted 0.4:

$$\mathcal{N}_3(\mu_3,\Sigma_3), \mu_3 = \begin{bmatrix} 2 \\ \\ 1 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1 & 0 \\ \\ 0 & 1 \end{bmatrix} \tag{4.17}$$

$$\mathcal{N}_4(\mu_4,\Sigma_4), \mu_4 = \begin{bmatrix} -2 \\ \\ -1 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 1.5 & 0 \\ \\ 0 & 1.5 \end{bmatrix} \tag{4.18}$$

We show the results of interpolating between a bimodal and a unimodal bivariate distribution in Fig. 4.8. The Quantile interpolants can be seen to have more pronounced modal separation. There are two discernible modes in all Quantile interpolants while the GMM interpolants are smoother and most lack multimodality. One noticeable artifact with the Quantile interpolants are "missing" lower quantiles. See section 4.6 for more details.

For visualizing uncertain vector fields, particularly where the distributions are non-

Gaussian and more specifically multimodal, and therefore presenting multiple possible trajectories, we propose the use of *modal curves*. While spaghetti plots show bundles or clusters of (possibly intersecting) streamlines, we want modal curves to be parsimonious representations of the major trajectories of the flow, where major is taken to mean the top b most likely directions. That is, we allow modal curves to bifurcate, if along its path, the curve encounters a distribution that is significantly multimodal. To construct modal curves, we seed and advect massless particles much like conventional streamlines but using the interpolated PDF to make decisions. That is, we advect using the velocity corresponding to the highest peak of a bivariate (for 2D) distribution. Modal curves are allowed to bifurcate along PDF modes after a minimum number of advection steps. Advection is performed as usual, using the fourth-order Runge-Kutta method. Each branch is a separate traditional streamline in the sense that branches are seeded at the branch point and advected forward or backward in the velocity field using the same direction as the parent branch. In order to reduce clutter, we remove branches according to criteria outlined in algorithm 4. Figure 4.9 shows results using $b = 2$.

We prune branches that cross over one another with one exception. Modal curves do not prune themselves at crossings that occur between "root" curves. Up to two "root" modal flow curves may advect from the seed point in either forward or backward integration. Both will be of the same age, i.e. have the same total advection steps at the end of an update cycle.

Pruning is performed to disallow ambiguation of primary flow paths and to keep

```
while not at end of the branching modal flow curve list do
    advect current branch by taking vector from distribution that forms smallest angle
    between itself and previous velocity taken by current branch;
    if new advection position crosses branch that is older and it is not the root then
        mark current branch and all of its descendents for removal;
        continue;
    else
        mark modal flow curve that was crossed by current modal flow curve and all of
        its descendents for removal
    end
    if current modal flow curve's position prior to its own advection has encountered an
    interpolated multimodal distribution and its minimum number of advection steps
    have been reached for another bifurcation then
        create and advect new modal flow curve along remaining highest probable
        velocity and add new branch to list;
        if new advection position of new branch crosses another modal flow curve then
            remove new branch modal flow curve from list;
end
process modal flow curve branches marked for removal
```

Algorithm 4: Advection for modal flow curves

computation to a minimum while allowing "feeler" breadth-search paths earlier in advection which can then be discontinued. Thus, we allow for the greatest divergence of advections along modes in PDF interpolants.

The GMM modal curves shown in Fig. 4.9 (top) contain only two branched forward advected curves, while for the Quantile modal curves in Fig. 4.9 (bottom), there are three branches, two root branches and a third child branch. Through monitoring intermediate advections, it was noted that all child branches encountered intersections and were subsequently pruned for the GMM advection. This can be explained by considering the entropy inherent in the GMM PDF interpolation method. GMM based modal curves tend to have more "noise" associated with their paths due to variations

in Gaussian component parameter fitting (EM) at grid point PDFs. Thus, modal curves branching between maximal divergent branches (such as those shown at the bottom of Fig. 4.9) often are completely pruned. In the toy example, the Quantile PDF interpolation method when applied, preserved one of the child branches and was not pruned because its path did not coincide with the rightmost root curve. Depiction of the most divergent flow paths are still observed in both methods, however.

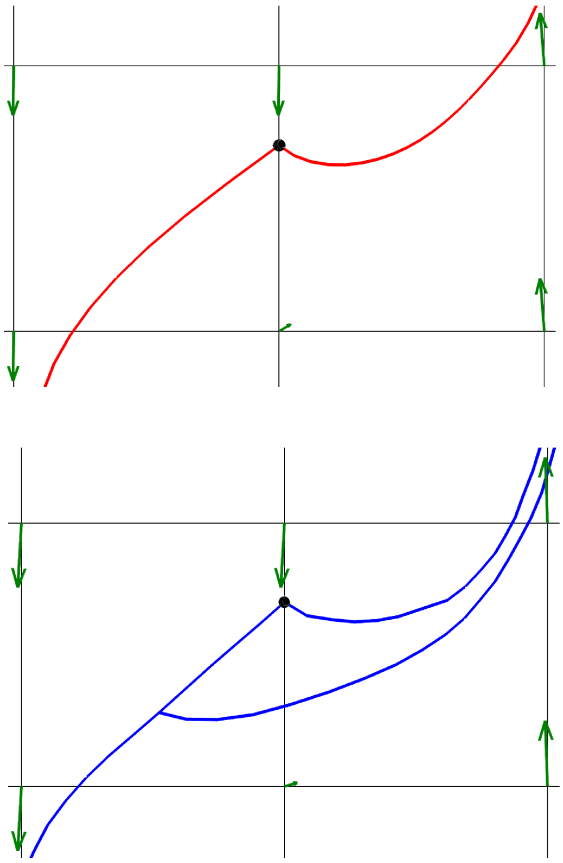


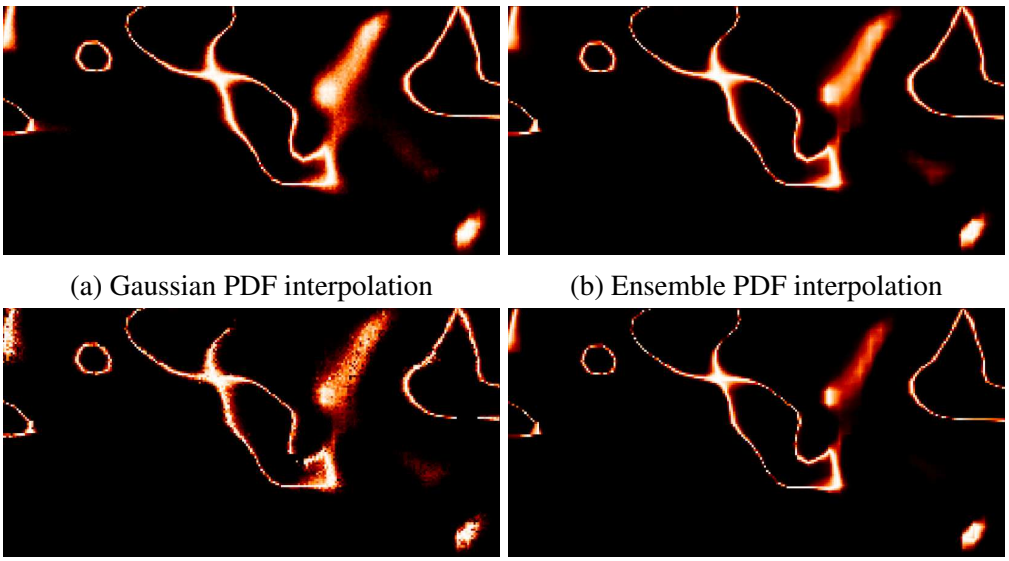
Figure 4.9: Toy example modal curves for (top) GMM and (bottom) Quantile PDF interpolation. Black dot denotes seed point. Mean vector is shown at grid points.

4.5.3 Simulation Data

Next, we provide verification of the interpolation methods and consideration of non-Gaussianity using simulation data. Our ensemble data-set covers a region of the Massachusetts Bay on the east coast of the United States of America [39] and is provided by Dr. Lermusiaux from MIT. The Massachusetts Bay volume in the study was divided into 53 x 90 grid with 16 depths. The depths at these 53 x 90 grid points vary significantly: depths as shallow as 90 meters and as deep as 196 meters were recorded. We use level zero, or the shallowest depth level in the ensemble and created visualizations using the temperature and velocity fields only.

The results of the GMM and Quantile PDF interpolation methods are shown for the level crossing probability (LCP) [61] at 35 degrees Fahrenheit (Fig. 4.10), using Eq. 4.20 in a mostly non-Gaussian region of the temperature field. Figure 4.11 shows the Shapiro-Wilk p-values for normality in the region where LCP is interpolated. Higher p-values of the Shapiro-Wilk test denote greater likelihood of a normal distribution. This region represents the lowest Gaussianity measured for the univariate temperature distributions at level zero of the ensemble data.

Quantile interpolated LCP matches closely with the Ensemble interpolated LCP. GMM interpolated LCP contains the most noise of all the interpolation methods and its probabilistic level set is also the most diffuse. The interpolated Gaussian assumption and the GMM interpolated LCP resemble each other more closely than do the Quantile and Ensemble interpolants.



(a) Gaussian PDF interpolation (b) Ensemble PDF interpolation

(c) GMM PDF interpolation (d) Quantile PDF interpolation

(e) LCP probability (left-to-right values range from 0.0 to 1.0)

Figure 4.10: LCP using (a) Gaussian, (b) Ensemble, (c) GMM and (d) Quantile PDF interpolation methods.

We use Eqs. 4.19 and 4.20 to calculate the LCP. Point $\mathbf{p}$ is a spatial location in the field, θ is the isovalue and $V_\mathbf{p}$ is a random variable at location $\mathbf{p}$. $V_\mathbf{p}$ is the interpolated temperature distribution at $\mathbf{p}$. Equation 4.20 is determined by considering whether the cumulative probability at the isovalue for the interpolated PDF is 0.5 at location $\mathbf{p}$. This formulation can be derived from [61].

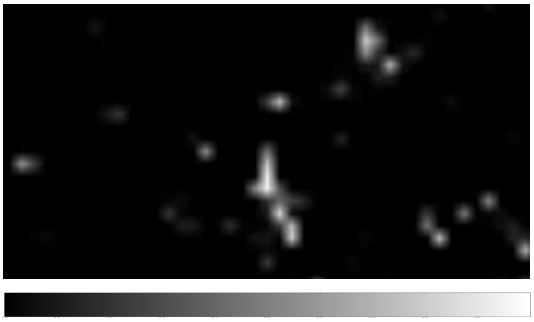


Figure 4.11: Temperature field Gaussianity as measured with Shapiro-Wilk test for normality. Shapiro-Wilk test produce p-values that range from 0.0 to 1.0. Higher p-values (white) denote greater likelihood of a normal distribution.

$$F_\mathbf{p}(\theta) = \int_{-\infty}^{\theta} V_\mathbf{p}(s)ds \tag{4.19}$$

$$\mathrm{LCP}_\mathbf{p} = 1 - F_\mathbf{p}(\theta)^4 - (1 - F_\mathbf{p}(\theta))^4 \tag{4.20}$$

Next, we examine the modal curves using all four methods and compare against the spaghetti plots in Fig. 4.13. The Gaussian modal curves (purple) tend to follow the primary bundle of the spaghetti plots but do not branch because of the single mode. The ensemble modal curves (green) show similar behavior but with branching. Sim-

ilarly, GMM (red) and Quantile (blue) modal curves bifurcate, but miss some of the streamline bundles of the spaghetti plots. The Quantile PDF interpolant modal curves have the closest paths in the rightmost part of the plot and GMM has a closer correspondence with the ensemble modal curves with its leftmost branches. There are two primary coherent bundles at the leftmost region of the spaghetti plots, where Quantile modal curves depict one bundle and GMM the other. Small variations in locality of the advections place both sets of modal curves closer to either streamline cluster and local modes dominate directional flow.

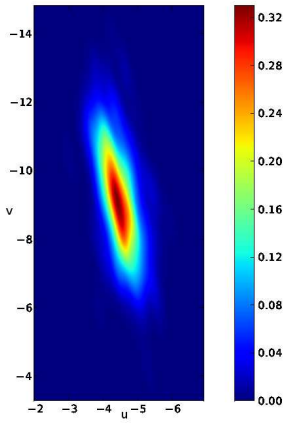
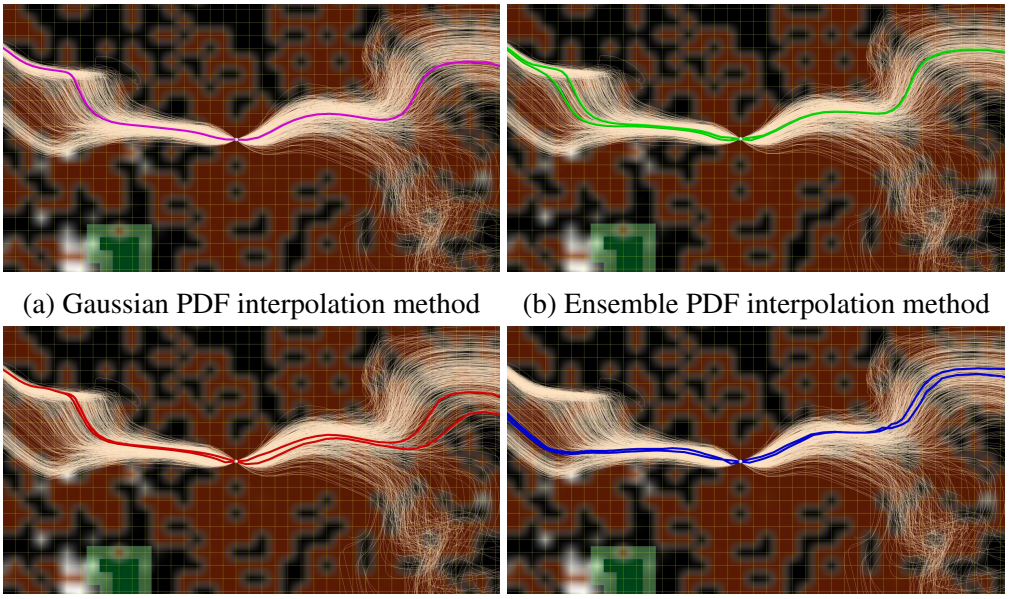


Figure 4.12: Representative non-Gaussian grid point (p-value = 4.6×10^{-4})

Note that the bivariate velocity Gaussianity is very low in our data set, where a typical example of a grid point distribution having relatively low variance along the direction of the minor eigenvector of its covariance matrix as compared to the major eigenvector direction (see Fig. 4.12). Also note that non-Gaussianity alone is not sufficient for deciding whether modal curves should bifurcate or not. We also need a test for multimodality. We achieve this based on size and separation of peaks. If one considers



(a) Gaussian PDF interpolation method

(b) Ensemble PDF interpolation method

(c) GMM PDF interpolation method

(d) Quantile PDF interpolation method

(e) Shapiro-Wilk p-value colorbar (left-to-right values range from 0.0 to 1.0)

Figure 4.13: Modal curves produced using (a) Gaussian, (b) Ensemble, (c) GMM and (d) Quantile PDF interpolation methods. White curves are spaghetti plots of streamlines. The greenish background represents land. The brownish-red background denotes bivariate multimodality greater than one. The black-gray-white background shows the p-values from the Shapiro-Wilk test (e), where higher p-values denote greater likelihood of a normal distribution. Most of the distributions in this region are multimodal non-Gaussian distributions.

multimodal marginal distributions individually, it is possible to generate samples that do not belong in the original bivariate distribution. Hence, it is important to consider the bivariate distribution itself rather than its marginals.

In Fig. 4.13, p-values are displayed for the Shapiro-Wilk test for Gaussianity along with modality from a Gaussian radial basis function (RBF) estimation. Each PDF has a set M of fitted Gaussian mean parameters. We calculate the greatest difference between any two Gaussian component means as a measure of multimodality. This is defined as follows: let $R = M \times M$, $r \in R$.

For all two-dimensional ensemble velocity values at a grid point, there are values: $u_{min}, v_{min}, u_{max}$ and v_{max} that represent the minima and maxima of the velocity components. Let the velocity sample extent γ, be defined as in Eq. 4.21.

$$\gamma = \| \left(|u_{max} - u_{min}|, |v_{max} - v_{min}| \right)^{\mathrm{T}} \| \tag{4.21}$$

Multimodality of PDF at a grid point is considered to be $true$ or $false$ depending on the following condition in Eq. 4.22, where our weighting factor is 0.10. This is a heuristic that ensures adequate separation of Gaussian components in the mixture.

$$multimodal = \begin{cases} true & \text{if } \max D > 0.10\gamma \\ false & \text{if } \max D \leq 0.10\gamma \end{cases} \tag{4.22}$$

The modal curves use only local ensemble information (PDF modes) for advection. Thus, they do not always bifurcate along bundles of ensemble streamlines. Figure 4.14

shows good separation along ensemble streamline bundles but was only reproducible with GMM PDF interpolation (likely due to over-smoothing of multimodality from density estimation with bivariates).

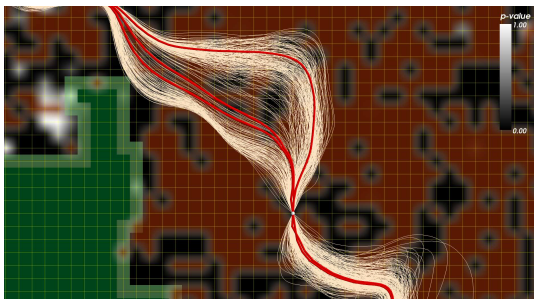


Figure 4.14: GMM modal curve exhibiting bifurcation with ensemble spaghetti plots.

We can also observe that modal curves do not always align themselves with regions of higher density of spaghetti plots. One of the contributing factors, if not the main contributing factor, is because we do not account for spatial covariance in our PDF interpolation. Streamlines in spaghetti plots are created from individual realizations where neighboring velocity information is available. The corresponding (i.e. pairing of) velocity information is lost in the PDF representation of the ensemble.

4.6 Discussion

Based on our limited investigation, Quantile interpolation is the method of choice for the case of univariate interpolation of non-Gaussian distributions since it provides the best SKL score when compared to the ensemble PDF interpolants as baseline.

Both GMM and Quantile PDF methods rely on having a good density estimate

either through EM or FKDE. However, Quantile PDF interpolation is particularly susceptible to the "curse of dimensionality" as one goes from univariate to multivariate interpolation. More data is needed to estimate the density. In our study, we use six hundred realizations for interpolating both univariate and bivariate joint distributions. Since PDF surface accuracy is proportional to the number of realizations, sample aliasing at lower frequencies may cause excess smoothing and can obscure modality. Aside from FKDE, there are other estimation methods such as adaptive kernel or projection pursuit density estimation [24] that can yield potentially better results with a limited number of samples for multivariates.

Limited samples also have adverse consequences during the integration stage for finding quantiles, where the sample space resolution needs to be increased in order to detect finer gradations of density per unit sample area. The complexity is proportional to n^d, where d is the dimension of the joint probability and n is the resolution of the sample space. Larger sample spacing can degrade high frequency probability surface detail. Such loss of detail may cause tearing in the reconstructed PDF because of incomplete quantile information during surface interpolation as can be seen in Fig. 4.8. This is not seen for univariates in our study but has been encountered for bivariates.

In contrast, because GMM will fit a given number of Gaussians to the data, GMM interpolation is less susceptible to over-smoothing of the density estimate due to lack of data. Hence it can detect modality (up to the number of Gaussian components) better than Quantile interpolation, but at the cost of accuracy associated with RBF. Another

consideration is that the GMM at each grid point can be performed in a preprocessing step and its interpolation will outperform Quantile interpolation in terms of fewer computations required per interpolant.

The interpolation methods presented in this chapter do not account for spatial covariance with surrounding grid point distributions. With GMM, we dismiss PDF-wide summary parameters that simplify covariance measurements and as a consequence we do not currently have heuristics for paired Gaussian component covariance. In the quantile case, we are interpolating unique surface values of individual PDFs which do not relate as a whole to surrounding PDFs when considered in isolation.

From our example of a two-dimensional univariate interpolation, we used LCP to visualize a probabilistic temperature field. Since LCP is determined based on the CDF, we can apply it directly to non-Gaussian fields.

4.7 Conclusion

This chapter investigated two PDF interpolation methods for both univariate and bivariate non-Gaussian distributions, in one and two dimensional space, and compared them against two baseline methods. The fundamental problem with PDF interpolation is that there is no unique path or set of intermediate interpolations between PDFs (especially in the more general case of non-Gaussian distributions). Our methods assume no prior knowledge of the ensemble data, in order to be more broadly applicable.

The interpolation methods presented in this chapter are designed to have certain

properties: variance should be bounded by the variances at grid points, no additional modes are introduced during interpolation, and the interpolants are PDFs. Using LCP and modal flow curves, we compared the results of the 4 interpolation methods on random fields exhibiting non-Gaussian distributions and their effects on the visualizations.

The Quantile PDF interpolation appears to offer the best fitting interpolants relative to the ensemble. However, it suffers from the "curse of dimensionality." Improvements to this method can come in the form of alternative ways to estimate density e.g. projection based methods that can capture multimodality with smaller sample sets. Hybrid methods that take advantage of both GMM and Quantile interpolation is also another area to be explored. We currently do not include spatial covariance in PDF interpolation, and is another area of further investigation. Also, while we started out focusing on non-Gaussian distributions, the modality of the distribution is perhaps more significant particularly. In the results presented here, we used an ad-hoc method for testing the modality of a distribution. There are more formal multimodality tests that can be incorporated in the future [17].

Ensembles, when considered as a random field of (simulation) measurements, instead of merely disparate parallel field data, offers promise for a much better insight into the nature of the ensemble when all members are visualized as their aggregate. Using interpolation on the grid point PDF directly provides a method for using the results of ensemble data in this more consolidated view. Additionally, if ensemble data can be stored as random field data exclusively, with better insight into the ensemble informa-

tion, this approach may prove more viable than conventional methods (spaghetti plots for example) which are in large use today. Finally, the results presented in this chapter is but the first step in analyzing and visualizing uncertainty in random fields.

Chapter 5

Streamline Likelihood

Traditional spaghetti plots from ensemble data provide no explicit information as to the likelihood of the realization flow paths. While intuitive assessment can be used when visualizing streamline density directly in such a plot, the display is often cluttered and difficult to interpret. We present a method to measure and visualize member streamline likelihood from an ensemble of vector fields. The method incorporates velocity probability density as a feature along each member streamline. We show visualizations of two different data sets using the proposed method.

5.1 Introduction

Ensemble vector fields (EVF) are common within the simulation community. Simultaneously rendering streamlines from multiple realizations leads to a "spaghetti" plot that is generally cluttered and difficult to interpret. Most current methodologies

summarize flow probability at the level of streamline geometry [43, 71]. These approaches are often restricted to parametric assumptions.

Our method uses the EVF to derive a varying feature along a streamline as a function of location. It is based on velocity density estimates from the EVF using non-parametric statistics. The sum of this feature along a streamline provides a streamline "likelihood" metric. This metric can then be used to compare streamlines from a data set. Our method allows users to affect rendering in *at least* two important ways: (1) clutter reduction and (2) uncertainty visualization.

5.2 Related Work

An overview of current methods for representing uncertainty in vector fields is given in [66]. Otto, et al. present analysis of 2D [48] and 3D velocity fields [49] with uncertainty approximated by Gaussian distributions. Our study uses non-parametric estimates of velocity. Pothkow et al. [62] discuss the application of non-parametric methods for uncertainty visualization. A variance based FTLE-like method for unsteady uncertain vector fields was first presented in [79]. Adaption of probabilistic and summary statistics are discussed in the survey paper [47]. Hummel et al. [23] was the first work to apply FTVA from [79] to address EVF visualization. Instead of analyzing only the particle deposition via FTVA, this chapter evaluates all locations along streamline data. Kuhn et al. [30] provided a method to render streamlines by scaling opacity over bill-board streamline segments. Although their method is quite effective

at reducing streamline clutter in crisp fields, our method uses uncertainty data to rank streamlines for rendering. Grottel et al. and Lampe et al. use non-parametric densities for scatter plots and trajectory data but not for EVF [19, 32]. Mirzargar et al. extend boxplots to curves for ensemble streamlines and hurricane track data [43]. Our method does not use streamline geometry to assign likelihood as they do. In [71], the authors present methods to visualize bundles of HARDI fibers using fiber encompassing hulls. Methods employing glyphs and information visualization techniques for ensembles are discussed in [76, 59, 12]. In [16], the authors cluster streamlines by fitting derived vector fields based on the streamline data itself. Our method works in reverse, where we derive features from the EVF and assign them to streamlines for further analysis.

5.3 Background

A time-varying flow field can be described as: $v : \Omega \times I \rightarrow \mathbb{R}^d$. v is defined over a spatial domain $\Omega \subseteq \mathbb{R}^d$, where the spatial dimension is d. The time interval is $I \subseteq \mathbb{R}$. An *ensemble E* is a set of m vector fields. The ensemble space is considered the intersection of all such vector fields, $\Omega_E = \Omega_1 \cap ... \cap \Omega_m$ and $I_E = I_1 \cap ... \cap I_m$.

$$E : \{1, ..., m\} \times \Omega_E \times I_E \rightarrow \mathbb{R}^d \tag{5.1}$$

We define the kernel density estimation (KDE) prior to discussing our KDE-based visualization. KDE is an often used approach to obtain a non-parametric estimation of

data density [52, 70]. Given a set of m univariate data samples x_i, $1 \leq i \leq m$, the KDE $f_h(x)$ is determined as:

$$\widehat{f}_h(x) = \frac{1}{mh}\sum_{i=1}^{m} K(\frac{x-x_i}{h}) = \frac{1}{m}\sum_{i=1}^{m} K_h(x-x_i) \tag{5.2}$$

based on a kernel function K and a bandwidth parameter h. The multi-dimensional KDE is defined in Eqs. 5.3:

$$\widehat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x}-\mathbf{x}_i), K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-\frac{1}{2}}\kappa(\mathbf{H}^{-\frac{1}{2}}\mathbf{x}) \tag{5.3}$$

where κ is a multi-variate kernel function that integrates to unity. $\mathbf{H}$ is a symmetric and positive definite bandwidth matrix. We use a Gaussian kernel with a bandwidth determined via Scott's Rule. Both selections are standard "rules of thumb" for multi-variate data sets [82].

5.4 Methods

We start with a set of streamlines (each with a corresponding member in the EVF). For the i^{th} streamline of m members, we consider it a set of points in Ω_{EVF}, $P_i = \{\mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_n\}$ with n integration steps. We can construct line segments between each point $\mathbf{p}_j$ and $\mathbf{p}_{j+1}$, for a poly-line representation: $S_i = \{l_1, l_2, ..., l_n\}$. Streamlines to be analyzed in the EVF belong to the set $S = \{S_1, S_2, ..., S_m\}$. For each $l_j \in S_i$, we seek to obtain the feature-vector $F_i = \{f(l_1), f(l_2), ..., f(l_n)\}$ corresponding to streamline S_i.

We do this for each $S_i \in S$.

Location $\mathbf{p}_{j-1} \in P_i$, is the starting point $\mathbf{p}$ for $l_j \in S_i$. We find the set of interpolated velocities $\{\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, ..., \bar{\mathbf{v}}_m\}$ at location $\mathbf{p}_{j-1}$, where $\bar{\mathbf{v}}_i$ is the i^{th} member's velocity at $\mathbf{p}_{j-1}$. We then compute a multi-variate KDE, with $\{\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, ..., \bar{\mathbf{v}}_m\}$, as the density estimate for the random variable V_{j-1}. V_{j-1} is the distribution of velocities at location $\mathbf{p}_{j-1}$. For the line segment $l_j \in S_i$, we obtain its feature from the EVF as: $f(l_j) = Prob(\mathbf{v}_i - \varepsilon \leq V_{j-1} \leq \mathbf{v}_i + \varepsilon)$, such that $0 \leq f(l_j) \leq 1$. ε should be a small relative to the range of samples.

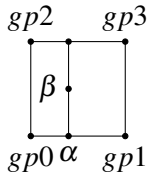


Figure 5.1: Unit cell interpolation using both α and β to interpolate within grid points $gp0...gp3$.

When deriving our probability density estimate for velocities, we first interpolate each velocity vector within a unit cell for each member i. See Fig. 5.1. Each grid point contains all the ensemble members' velocities at that point.

The features F_i are added to assign a ranking for each $S_i \in S$. We term the rank of streamline S_i as: *streamline likelihood*. The streamline rankings can be displayed, used to reduce the number of streamlines for visualization, or the user may focus visualization on a particular range of likelihoods. Our results display likelihood rank. To render S_i, a color-map for velocity density is used. We also render overall likelihood with transparency as a ratio of streamline likelihood to greatest likelihood in S. For

rendering individual streamlines, the line segment thickness is set to a minimum value. Its width is scaled by $\frac{F_i}{max(F)-min(F)}$, where F is from all S.

5.5 Experiments

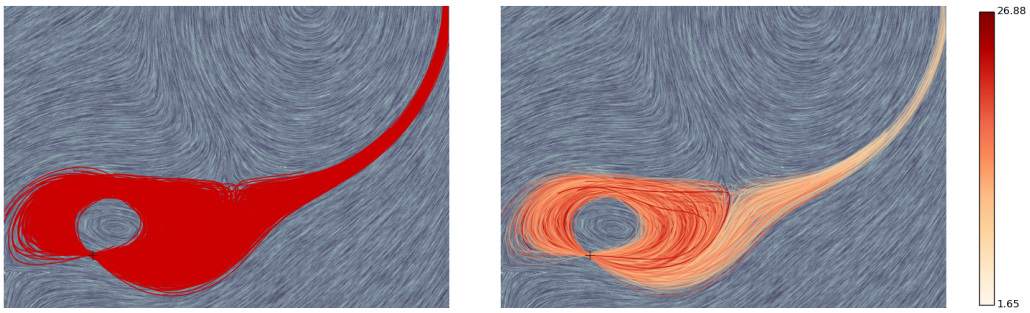
5.5.1 Implementation

Our results were obtained from code written in Python, utilizing the SciPy package [11]. The PC system used an Intel Core i7-3930k with 32 GB of RAM. All Python scripts were run as single-threaded processes.

5.5.2 Data Sets

Ocean Simulation Ensemble This data set covers a region of the Massachusetts Bay on the east coast of the United States of America [39, 35]. The Massachusetts Bay volume in the study was divided into 53 x 90 grid with 16 depths. The depths at these 53 x 90 grid points vary significantly: depths as shallow as 90 meters and as deep as 196 meters were recorded.

Lock-exchange Simulation Ensemble The initial conditions are heavy fluid on one side and light fluid on the other, separated by a barrier [86]. The lock-exchange data has the following parameters: 128 x 128 grid with velocity measurements, 1000 realizations.



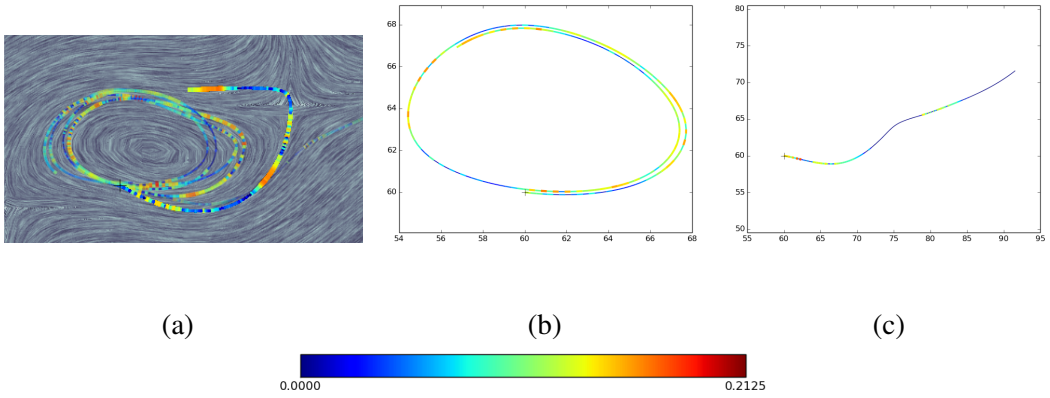
(a) (b)

Figure 5.2: Lock-exchange data. All 1000 streamlines seeded at coordinates $(60, 60)$. Background: mean vector-field LIC. (a) Conventional "spaghetti" plot. (b) Streamlines rendered to show relative likelihoods as derived from EVF. Color-bar applies to (b) only. Opacity is proportional to likelihood in (b).

5.5.3 Results

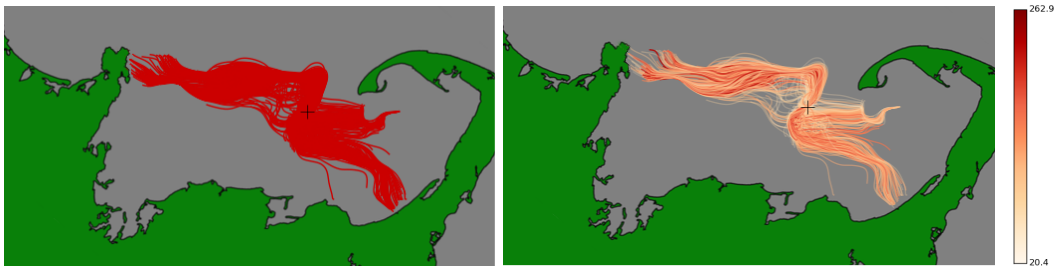
Figure 5.2a shows a traditional plot with no likelihood ranking as compared to Fig. 5.2b, that color codes the overall likelihood for each streamline. Streamlines are backward integrated. We show 1000 streamlines rendered simultaneously over the mean-field LIC. In Fig. 5.2b, there is a clear distinction in flow paths of individual streamlines. There is also a clear division in flow behavior not easily seen in Fig. 5.2a. Figure 5.2b shows that most of the streamlines flowing to the right have generally lower likelihoods. However, flow that circulates to the left contains streamlines with more variance in total likelihood. Streamline likelihood is not entirely defined by geometric location. Using our method, it is due to small variations in velocity probability along a streamline.

Figure 5.3 shows streamlines from Fig. 5.2 with local features. Each integration step has its velocity density rendered via color and thickness. The streamline in Fig.



(a) (b) (c)

Figure 5.3: Streamlines show velocity probability density feature along their trajectories. (a) Top one-percent with opacity scaled for overall likelihood, (b) higher-than-average member, and (c) lower-than-average member from Fig. 5.2.



(a) (b)

Figure 5.4: Ocean simulation data. All 600 streamlines seeded at coordinates $(48, 30)$. (a) Conventional "spaghetti" plot. (b) Streamlines rendered to show relative likelihoods as derived from EVF. Color-bar applies to (b) only. Opacity is proportional to likelihood in (b).

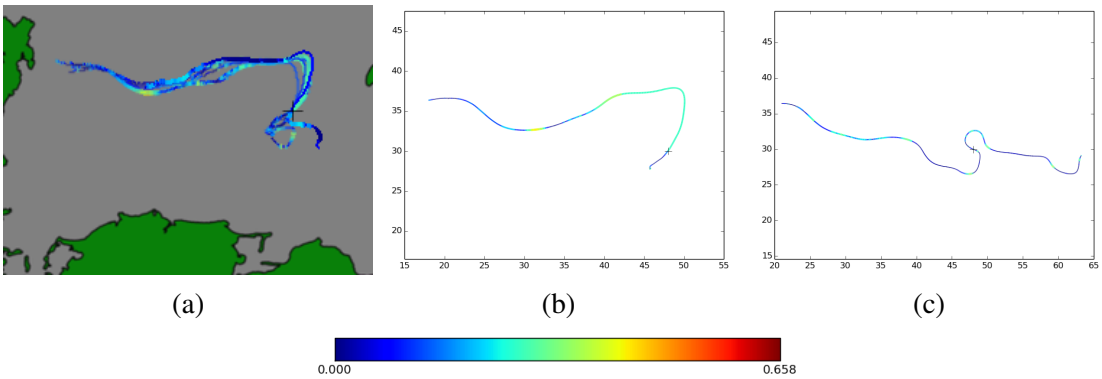


Figure 5.5: Streamlines show velocity probability density feature along their trajectories. (a) Top one-percent with opacity scaled for overall likelihood, (b) higher-than-average member, and (c) lower-than-average member from Fig. 5.4.

5.3b shows higher velocity density values along its path, while the streamline in Fig. 5.3c illustrates lower velocity densities along its trajectory. This information is not shown in the traditional plot of Fig. 5.2a.

Figure 5.4 and Fig. 5.5 show surface currents for the ocean data. Streamlines are both forward and backward integrated. There are streamlines with overall less likelihood that have similar partial trajectories with streamlines of higher likelihood. This view of streamline likelihood is not discernible using methods based entirely on streamline geometry. For instance, there are streamlines with low overall likelihood that would be included geometrically in the upper bundle to the right of the seed location in Fig. 5.4. Fig. 5.5 shows that streamlines terminating near the seed point tend to have higher overall likelihoods due to the high velocity probability density for those paths.

5.6 Conclusion

We have provided a method to extract features from an EVF for streamline analysis. The method is based on finding velocity density estimates at locations along streamlines and is not limited to parametric assumptions. Member vector fields are readily stored as an ordered list and thus KDE are easily constructed. While we have shown results for 2D data sets, it should be straightforward to extend our method to 3D vector fields.

Besides the use cases shown in the results, another application of our method includes reduction in storage of pre-computed streamline data. Finite-time Variance Analysis [23], and other statistical post-processing, can benefit by using our method to remove less significant streamlines from consideration.

For future work, we plan to investigate additional rendering approaches using streamline likelihood along with further exploration of the potential use cases already mentioned.

Chapter 6

Transport Similarity

Currently, there is no method for visual analysis of ensemble vector fields (EVF) that provide identification of flow trends and general flow similarity over the extent of transport across ensemble members. Finite-time Variance Analysis (FTVA) provides flow structure information only on particle distributions at the termination of streamline integration. In this chapter, we first present a flow structure based on streamline clustering. Second, we discuss a method using streamline clustering to provide information of flow coherence at corresponding spatial regions in the EVF. We consider the regions where bifurcation in flow trends among the EVF members occur. We will also discuss how both methods can be used as a sequential framework for EVF analysis, by using the results of the scalar flow structure to find regions of member flow dissimilarity for further analysis.

6.1 Introduction

Ensembles of vector field data, as produced via *Computational Fluid Dynamics* (CFD) simulations, are now common within the simulation community, in order to represent the output of a fluid model using distributions of input parameters [86]. The variation in parameter selection can represent uncertainty about boundary conditions, densities or other relevant input.

As a consequence, we are now faced with the challenge of analyzing and visualizing ensemble vector fields (EVF). EVF are made up of individual realizations, each a possible outcome, of the simulation. Flow has traditionally been visualized by advection of mass-less particles, e.g streamline integration, in a certain vector field. There are many methods to analyze single instance vector fields and quantify their flow.

When extending those methods to ensembles, multiple problems arise. For one, statistical variation likely exists between the members of an EVF. A key visualization problem is first detecting and then displaying that variation. Most importantly, we want to draw attention to significant trends among members. Modes of flow coherence (e.g., trends) should ideally be considered over the full extent of flow: (1) initially, for identification, between the entire paths of each particle's movement sharing a common seed location within the field and then (2) subsequently, within known regions of the field where the modes of variation are clearly evident, as determined from the results of the initial consideration.

Until now, such methods as Finite-time Variance Analysis (FTVA) [79] have been

employed to quantify global flow variation in an ensemble. Such methods only investigate variation in the flow through a given seed location at the termination of integration, via the principal components of the covariance matrix computed from the positions of particle deposition. Transport separation, however, may occur anywhere along streamlines with a common seed over the ensemble. FTVA, therefore, overlooks potentially important bifurcation between members.

In this work, we provide the following contributions:

- We utilize proven and efficient streamline clustering methods [9] to characterize, on the scale of the entire field, the flow coherence and bifurcation of the ensemble.
- We quantify via a two-stage streamline clustering method using representative streamlines from their cluster, the degree of flow coherence in regions of known bifurcation across the ensemble members.
- We show how both methods can be used together by first employing the flow structure to identify potential bifurcation and then the exploration of the regions of bifurcation.

6.2 Related Work

Much work had been done to define and identify global features of flow fields for crisp vector fields. Relevant publications are summarized here.

Lagrangian Coherent Structures (LCS) are a broad class of feature identification for the fluid medium [53]. Perhaps the first notable example is the Finite-time Lyaponov Exponent (FTLE) fields [21] for steady and unsteady vector field visualization.

Generalization of LCS has been discussed in depth [29]. Frameworks for flow field structure definition and visualization have been laid out in [74]. There, the authors discuss pathline predicate definitions relevant for given investigations of flow phenomena.

A variance based FTLE-like method for unsteady uncertain vector fields was first presented in [79]. This method reports the spatial second moment of particle destination, using the principal components of their covariance matrix as a result of initial uncertainty in the vector field. Theisel et al. [50], [51] examined uncertain vector field topology using Gaussian uncertainty. Analysis of streamline separation at infinity using time-discrete Markov Chains was explored in [69], in order to remove the finite-time requirements from [79].

While the papers discussed so far did not utilize EVF, adaption of probabilistic and summary statistics are discussed in the survey paper [47]. Hummel et al. [23] was the first work to apply FTVA from [79] to address EVF visualization. Their paper also used a Minimum Spanning Tree (MST) to detect and visualize trends in particle destinations at finite-time.

With novel numerical schemes to generate ensemble data using non-Gaussian input parameters [86] and [77], techniques to show the subtle variation and modality in output EVF is becoming increasing needed from the visualization community.

Similar to our work but not appropriate for flow trend detection, are several streamline clustering methods. In [34], the authors extend the point-based clustering algorithm called Density-based Spatial Clustering of Applications with Noise (DBSCAN) to line segments. They applied this method to find representative trajectories in hurricane track data. In [40], the authors use curvature distribution of a field of streamlines to find shape similarity. Neither of these studies are ensemble based, but use crisp vector fields.

Chen et al. [9] provide an efficient two-stage streamline clustering method based on spatial properties. The first-stage groups streamlines using k-means for feature vectors comprised of the start-point, mid-point, and destination-point of streamlines. Their second-stage finds sub-clusters from the first-stage, based on linear and angular entropy. They summarize flow in regions by finding representative streamlines closest to cluster centroids. Evaluation of fiber clustering methods for diffusion tensor imaging is discussed in [44]. It was from this study that [9] gave an approximate and efficient method.

Guo et al. outline a framework in [20] to provide an interactive assessment of ensemble variation. They call their system eFLAA (ensemble Flow Line Advection and Analysis). They present a novel parallel computation for calculating streamline spatial difference over an ensemble and then visualizing the differences. They compute various features of their ensembles (e.g., carbon dioxide concentration) along streamlines whose variation meets a given threshold.

Mirzargar et al. [43] extend boxplots to curves. They apply their method to quan-

tify and visualize ensemble streamlines and hurricane track data. While they show the band-depth for individual streamlines, they do not delineate bifurcation between member streamlines. Their method is not directly applicable to a dense-field summary of streamline data.

6.3 Background

We briefly describe the current methods for extracting flow structure from crisp vector fields and EVF. We also discuss information entropy as related to streamline identification and its potential use for EVF statistics.

6.3.1 Flow Classification

Flow classification is based on material transport in vector fields, and thus provides a global picture of the vector field. The *flow map* Φ is derived from the vector field using integration.

$$\Phi(\mathbf{x}(t);T) = \mathbf{x}(t+T) \tag{6.1}$$

Equation 6.1 describes the final location of a particle seeded at $\mathbf{x}$ at time t and advected for an interval T. The field is not required to be time-varying and in such a case, T simply refers to the number of integration steps forward or backward in Φ.

6.3.2 Finite-time Lyapunov Exponent

Taking the largest eigenvalue of the right-Cauchy Green deformation tensor, Eq. 6.2, we find the magnitude of the direction of greatest stretching in the flow medium at $\mathbf{x}(t)$. The tensor removes effects of reference frame rotations in $\nabla\Phi$.

$$\lambda_{max}(\nabla\Phi(\mathbf{x}(t);T)^T \nabla\Phi(\mathbf{x}(t);T)) \tag{6.2}$$

The finite-time Lyapunov exponent is a logarithmic scaling of the maximum direction (Eq. 6.3).

$$FTLE(\mathbf{x}(t),T) = \frac{1}{T} log\sqrt{\lambda_{max}} \tag{6.3}$$

FTLE is a scalar field over the vector field domain. Finding its height ridges provides a topological skeleton of the regions in contraction or expansion.

6.3.3 Ensemble Vector Fields

Ensemble vector fields (EVF) are uncertain vector fields derived from variations between multiple instances (or runs) of an experimental/observation space (i.e., a container or geographical volume for inspection and the related starting conditions, computational model, and fluid characteristics). Repeated runs of the same simulation, with varying simulation input parameters, produce member realizations that taken together can be considered as a distribution of all possible outcomes of the field for a given set

of parameters. For the purposes of this study, we limit our definition of an EVF to the definition given in Hummel et al. [23].

In that definition, a time-varying flow field can be described as in Eq. 6.4, where v is defined over a spatial domain $\Omega \subseteq \mathbb{R}^d$.

$$v : \Omega \times I \rightarrow \mathbb{R}^d \tag{6.4}$$

The time interval is $I \subseteq \mathbb{R}$. An EVF is a set of m vector fields over the same spatial domain and the ensemble space can be considered to be the intersection of all such vector fields, $\Omega_{EVF} = \Omega_1 \cap ... \cap \Omega_m$ and $I_{EVF} = I_1 \cap ... \cap I_m$.

$$EVF : \{1, ..., m\} \times \Omega_{EVF} \times I_{EVF} \rightarrow \mathbb{R}^d \tag{6.5}$$

$EVF(i, ., .)$ corresponds to the i-th realization in our ensemble. We can see an example of particle transport in an ensemble (Fig. 6.1).

6.3.4 Finite-time Variance Analysis

A probabilistic variant of FTLE is called the FTVA, Eq. 6.6. It takes the covariance matrix of particle positions advected over the ensemble domain from given seed locations. It was first presented by Schneider et al. [79].

$$FTVA(\mathbf{x}(t), T) = \frac{1}{T} log \sqrt{\lambda_{max}(Cov(\mathbf{x}(t); T))} \tag{6.6}$$

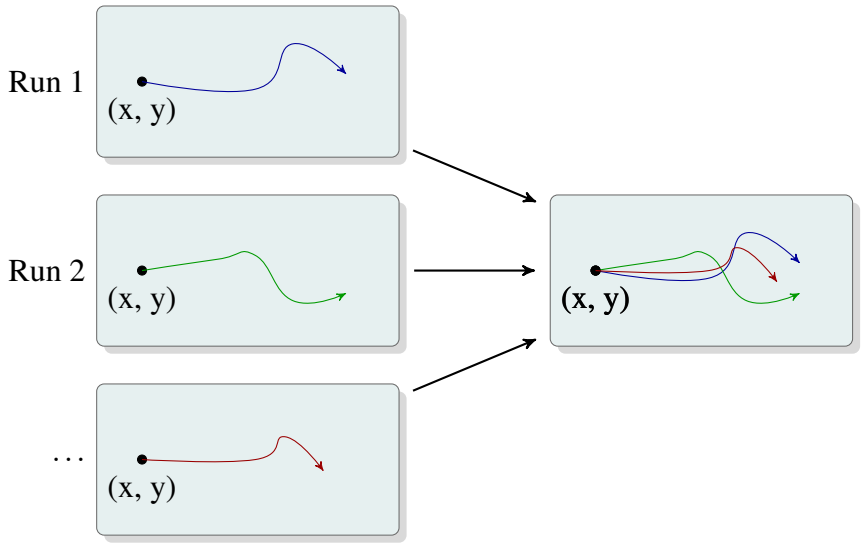
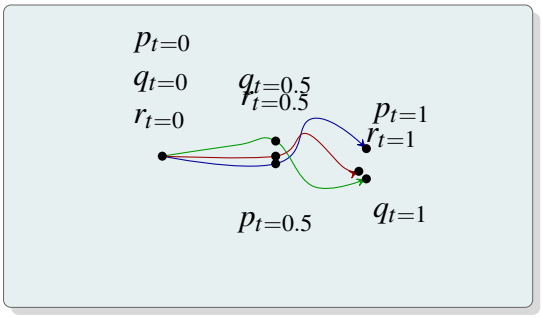


Figure 6.1: Streamlines seeded at the same positions in all members of the EVF have different transport paths. Seeds in the EVF lead to stronger or weaker path trends. Note that this is similar to FTVA for EVF, but that streamlines may terminate with weak separation but have strong separation anywhere along their trajectories. Here, the green streamline branches from the blue and red streamlines, but all terminate with weak variance.

6.3.5 Streamline Information Entropy

Many works have used information theory [83] applied to streamline geometry [18], [40], [41] for the purposes of selecting streamlines. In this study, we are interested in summarizing streamlines from the EVF with a common seed. We use this summary in two ways. First, it is used to weight the sampling frequency of points along streamlines (i.e., a higher sampling frequency captures greater streamline variability). Second, we utilize entropy as a reference map to better understand the overall variation in streamline geometry from the EVF.

We use both linear and angular streamline entropy [9]. Equation 6.7 represents the linear entropy [18], E_L, of single streamline. L_S is its total length and m the number of



Feature Vectors

p: $< p_{t=0}, \ldots, p_{t=0.5}, \ldots, p_{t=1} >$

q: $< q_{t=0}, \ldots, q_{t=0.5}, \ldots, q_{t=1} >$

r: $< r_{t=0}, \ldots, r_{t=0.5}, \ldots, r_{t=1} >$

$\ldots$

Figure 6.2: Shown here are three example streamlines all starting at the same location. We use at least the beginning, middle, and end locations. Other points used in the feature vector are evenly spaced over the approximated arc length and registered.

positions available from the numerical integration. D_j is the length of the j-th segment.

$$E_L = -\frac{1}{log_2(m+1)} \sum_{j=0}^{m} \frac{D_j}{L_S} log_2 \frac{D_j}{L_S} \tag{6.7}$$

Equation 6.8 represents the angular entropy [41], with A_j the angle of the line segment j, L_A is the total angular variation along the streamline (e.g. the sum of the absolute values of the A_j), and E_A the total angular entropy for the streamline.

$$E_A = -\frac{1}{log_2(m)} \sum_{j=0}^{m-1} \frac{A_j}{L_A} log_2 \frac{A_j}{L_A} \tag{6.8}$$

Both of these metrics summarize the degree of variation in a streamline over its entire path.

6.4 Methods

We first describe our method for extracting a cluster-based flow structure from an EVF. Second, we provide an exploratory region-based EVF similarity metric based on

the same underlying streamline clustering method.

In section 6.5, we show how the results from our flow structure can guide a user to probe more deeply into the regions that give rise to global bifurcation in transport among the members.

6.4.1 Cluster-based Flow Map

For a seed in the simulation domain Ω_{EVF}, we define a feature vector to represent each streamline. We sample position as a spatial feature. The number of features included are at a minimum the initial, middle, and terminal positions of a streamline. Streamline clusters are found for each seed in Ω_{EVF}, where a velocity value has been stored from the simulation. This result is similar to Φ. The *cluster map* Φ_C, is represented as:

$$\Phi_C(\mathbf{x}) = |C_S|, C_S = \{c_1, ..., c_n\} \tag{6.9}$$

where $\mathbf{x}$ is the location of the seeded streamlines, C_S is the set of all streamline clusters c_i, i is an integer such that $0 \leq i \leq n$, and n the number of clusters. $|C_S|$ is the cardinality of the finite set C_S. Set c_i contains the similar streamline feature vectors seeded at $\mathbf{x}$.

We use the mean linear $\overline{E}_L$, and mean angular entropy $\overline{E}_A$ of a population of streamlines to determine the frequency of sampling. The following steps are performed in computing $\Phi_C(\mathbf{x})$ for each $\mathbf{x}$:

Step 1 *Lookup precomputed* $\overline{E}_L$ *for* $\mathbf{x}$.

Step 2 *Lookup precomputed* $\overline{E}_A$ *for* **x**.

Step 3 *Calculate the number of streamline sample points,* $\propto \left(\overline{E}_L + \overline{E}_A\right)$.

Step 4 *For each streamline, assign a feature vector.*

Step 5 *Perform DBSCAN on all streamline feature vectors.*

Step 6 *Record the number of clusters found in* Φ_C.

The number of regularly sampled features is proportional to the mean linear and angular entropy (see step 3 above). We linearly interpolate the number of samples between a minimum and a maximum positive integer and take the floor of the result. The upper-limit on the number of samples is dependent on the data or user constraints. The α for interpolation is equal to the ratio of the average of the linear and angular entropy (at the seed) to the absolute value of the difference between the maximum and minimum total entropy (linear and angular entropy combined) from the data set.

Because we desire to detect bundles of streamlines that may start out together, diverge, and finally converge over the ensemble members, we need to sample spatial features that are registered between the streamlines. Note that our method of clustering is inspired by [9]. They found sub-clusters based on entropy from initially grouping streamlines sampled at three spatial locations each. We use streamline entropy to determine sample frequency for streamlines at a seed. In Fig. 6.2, the blue and red streamlines are spatially similar. However, if the minimum three points are used for the feature vector, all streamlines in the example would be found in a single cluster.

6.4.2 Spatial Feature Registration

Streamline registration of spatial features is accomplished via an approximation of arc length. t is a real number on the interval $[0,1]$ and is considered a fraction of the total arc length of a curve (streamline). The arc length L of curve S is defined as in Eq. 6.10 on the interval $[a,b]$. $ds^2 = dx^2 + dy^2$, for the infinitesimal line segment ds.

$$L_S = \int_a^b ds = \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx \tag{6.10}$$

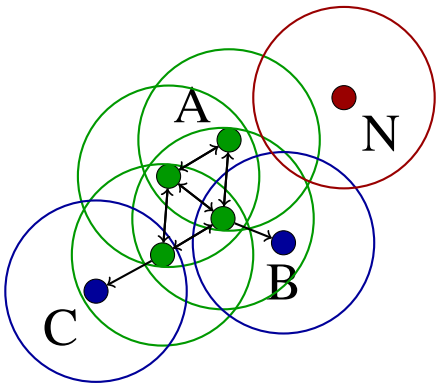


Figure 6.3: Illustration of DBSCAN cluster analysis requiring minimum points constituting a cluster. Points around A are core points. Points B and C are not core points, but are density-connected via the cluster of A (and thus belong to this cluster). Point N is Noise, since it is neither a core point nor reachable from a core point. DBSCAN also requires a maximum distance parameter ε that determines density-connected points [13].

S is the streamline from which we have a set of points derived from numerical integration in the vector field. l can be considered an ordered list of those points and can be accessed by index i. For our finite approximation, when n is the number of points from integration, we have:

$$L_S = \sum_{i=0}^{n-1} dist(l(i), l(i+1)) \tag{6.11}$$

where *dist* is the Euclidean distance between two points. Parameter t then, is the fraction of L_S we wish to consider for comparison between a registered set of streamlines.

6.4.3 Cluster Parameter Selection

Hummel et al. used a MST for terminal point trend clustering [23]. That study reported using a fraction of the average length of streamlines for the minimum distance between clusters.

We apply DBSCAN to assign cluster labels to member streamlines. Refer to Fig. 6.3 for the algorithm description. DBSCAN takes two parameters: ε, the maximum distance between features in a cluster, and *minPts*, the minimum number of data points in a cluster. The value for ε can be chosen by using a k-distance graph, plotting the distance to the $k = minPts$ nearest neighbor. Good values of ε are where this plot shows a strong bend. If ε is chosen too small, a large part of the data will not be clustered. Whereas for a too high value of ε, clusters will merge and the majority of objects will be in the same cluster [13].

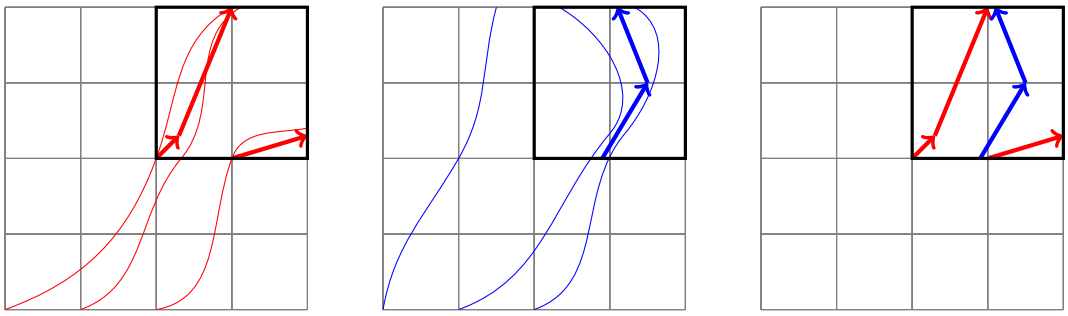
We, however, take an approach similar to [23], setting the minimum distance between clusters to be related to their spatial domain. We use five percent of the diagonal distance across the full simulation domain as ε. For p-values, most authors refer to statistically significant as $P < 0.05$ [46]. Thus, five percent presents itself as a good "rule-

of-thumb" for the fraction of the domain. We do not use the length of the streamlines themselves because we apply our clustering to multiple points along the steamlines. ε needs to be a function of the spatial domain size instead.

Ester et al. recommends $minPts \geq D + 1$, where D is the dimension of the data set [13]. Karami et al. provide adaptive strategies for parameter selection but at significant computational overhead [28]. In our study, *minPts* is set to five percent of the training data set size (e.g. the number of streamlines for a seed).

6.4.4 Region-based EVF Flow Similarity

In Fig. 6.4, EVF exhibit regional flow coherency when representative flow lines for the region can themselves be clustered.



(a) (b) (c)

Figure 6.4: Schematic for observing regional clustering across ensemble members. (a) and (b) represent separate realizations with the upper quadrant (heavy outline) considered. (c) EVF union of members (a) and (b). Arrows are representative flow for the region.

We summarize the possible combinations of coherence over the ensemble members in Fig. 6.5. The *lower-left quadrant*: coherent flow in individual members and

among members. *Lower-right*: incoherent flow in members but coherent among members. *Upper-left*: coherent flow in members but incoherent among members, and in the *upper-right*, incoherent flow in individual members and among members.

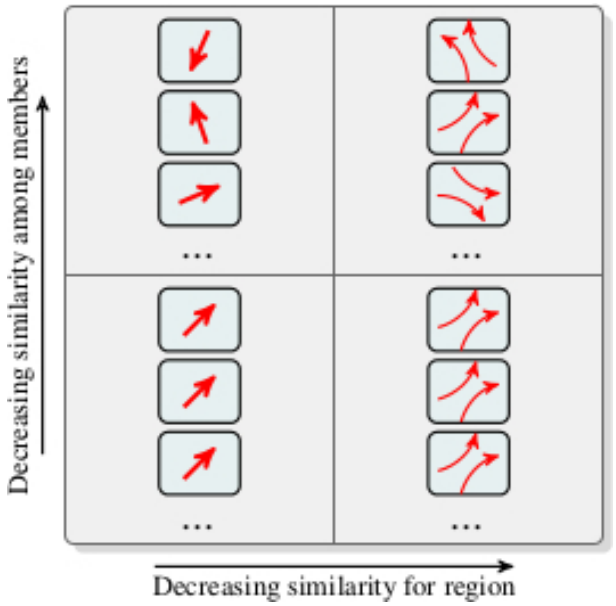


Figure 6.5: Matrix showing primary combinations of EVF flow similarity. Each box shows hypothetical representative flow (arrows) for a given region in a member of the vector field.

We utilize the following steps to summarize flow in a region from the EVF. After steps 1 through 4 are complete, ensemble flow coherence is visualized in a region using representative streamlines.

Step 1 *Define a spatial region ($\subset \Omega_{EVF}$) for inspection.*

Step 2 *Gather precomputed streamline segments spanning the region.*

Step 3 *For each member, cluster streamline segments.*

Step 4 *Assign a representative streamline per cluster by using the streamline closest to the cluster centroid via Euclidean distance.*

6.5 Experiments

6.5.1 Implementation

Our results were obtained from code written in Python, utilizing the SciPy package, Sci-kit Learn [27], and HDF5 [10] via H5py [11]. The PC system used an Intel Core i7-3930k with 32 GB of RAM. All Python scripts were run as single-threaded processes. Tables 6.1 and 6.2 show compute times for algorithms used in this study. Time spend on file I/O is excluded. We omit timings for regional analysis, since compute times vary widely based on dimensions of the selected area.

data set	resolution	members	time steps	flow map	FTVA
Lock	128x128	20	1100	30375.94s	206.69s
Ocean	53x90	30	1100	9285.76s	54.09s
Stir	152x152	15	1100	32126.12s	335.73s

Table 6.1: Timings for flow maps and FTVA pre-computation for the data sets in this study. Number of members reflects the members used in the computations and not necessarily the total available members. In cases where less members are used than available, those members used were randomly chosen from the available set. Compute times are dependent on number of ensemble members and field resolutions.

6.5.2 Data Sets

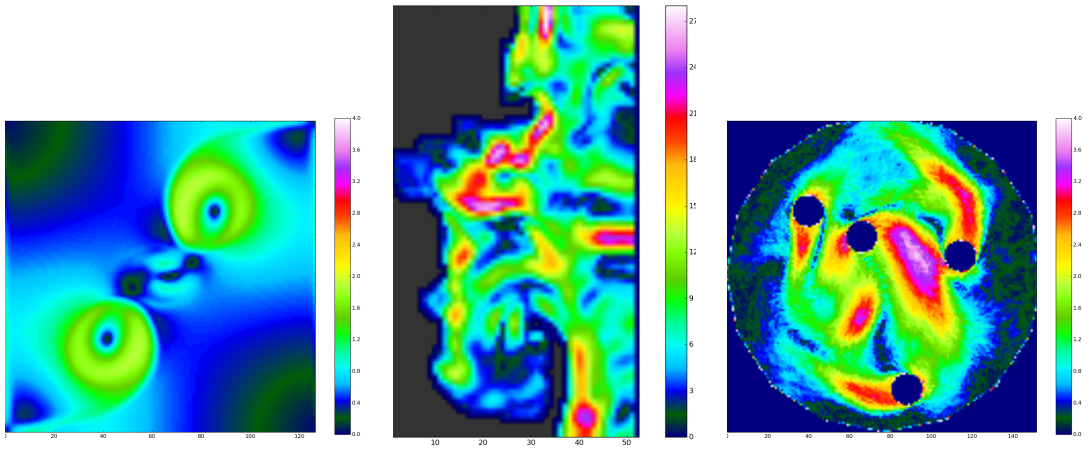
Lock-exchange The initial conditions are heavy fluid on one side and light fluid on the other, separated by a barrier (the lock) [86]. At initial time, that barrier is removed, and the flow is allowed to evolve. See Fig. 6.6a. Initial uncertainty originates from not knowing the position of the interface between the two fluids. In other words, the

data set	term.	3 pts.	13 pts.	var. pts.	entropy
Lock	389.23s	22523.08s	23900.47s	23086.69s	20359.93s
Ocean	97.32s	4590.12s	4704.98s	3581.92s	11842.21s
Stir	602.34s	31761.07s	32555.85s	14291.28s	28710.68s

Table 6.2: Timings for pre-computation of clustering for terminal points (term.) and multiple streamline samples (3 pts., 13 pts., and variable pts. between 3 and 13) for the data sets in this study. Included is the total calculation time of the linear and angular entropy pre-computations. Compute times are dependent on number of ensemble members and field resolutions. Identical resolution and number of members used for these timings are shown in table 6.1.

volumes of heavy and light fluid on each side is not exactly known, and the initial barrier slides left and right accordingly. At the start of the simulation, the probability distribution of the position of the barrier is Gaussian. Therefore, after infinite time, it is expected that the barrier is characterized by a similar Gaussian distribution, but with the light fluid on top of the heavy one, and with the variance of distribution stretched if the size of the whole lock domain is not square. However, the probability distributions of the interface or the dominant dynamics in between this start and infinite time are not assumed Gaussian. The lock-exchange data has the following parameters: 128 x 128 grid with velocity measurements, 1000 realizations.

Ocean This data set covers a region of the Massachusetts Bay on the east coast of the United States of America [39, 35]. See Fig. 6.6b. The Massachusetts Bay volume in the study was divided into 53 x 90 grid with 16 depths. The depths at these 53 x 90 grid points vary significantly: depths as shallow as 90 meters and as deep as 196 meters were recorded. The important visualization concern for this data set is understanding where ocean current streamlines seeded at the same location split into distinct paths



(a) (b) (c)

Figure 6.6: Single member velocity magnitude fields from, (a) Lock-exchange data, (b) Ocean data, and (c) Industrial Stirring data.

in different realizations. For example, streamlines may deviate geometrically between their common seed positions and their individual termination position in a set of streamlines from multiple realizations, but still have similarly located terminal positions. See Fig. 6.7.

Industrial Stirring The stirring data set is a set of 15 two-dimensional flow fields resulting from the simulation of mixing in a stirring apparatus [23]. See Fig. 6.6c. The device consists of two counter-rotating pairs of mixing rods that stir a medium in a cylindrical tank. The ensemble was generated by slightly varying the viscosity of the fluid to investigate mixing quality of the device for a range of different fluids. The primary question for this data set regards the effectiveness of the stirring process. An ensemble visualization is expected to be able to identify regions where the mixing quality is high or low throughout the ensemble.

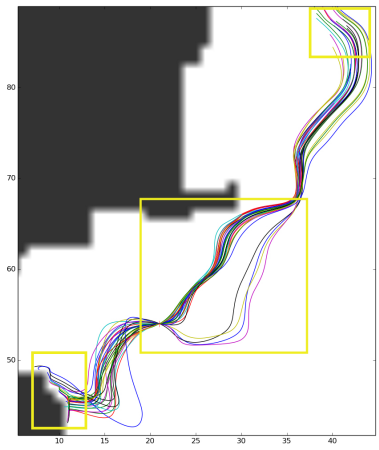


Figure 6.7: Member streamline bifurcation between members in ocean data set. Seed location is at the red cross marker. Streamlines separate along their trajectories forming two distinct clusters as seen in the central border selected region in yellow. However, the distribution of their terminal positions alone (FTVA) do not account for these separate bundles, especially as seen in the spread of the terminal positions in the upper-right and lower-left of the Fig. (additional yellow boxes).

6.5.3 Results and Analysis

This study does not use individual member variances (FTLE) in the consideration of FTVA [23], but compares our new visualizations to FTVA only. Using FTLE generalizes the application of FTVA to sensitivity between otherwise identical simulation runs (where variations due to numerical error and other noise-based variation is potentially present). Perhaps a more informative metric on FTVA, and streamline clustering in general, is streamline entropy, as discussed in section 6.4. Thus, our visualizations refer to both average linear and angular entropy maps, as well as FTVA maps, for interpretation of streamline clustering and sampling frequency for individual streamlines.

Lock-exchange The first data set to be evaluated is the lock-exchange simulation

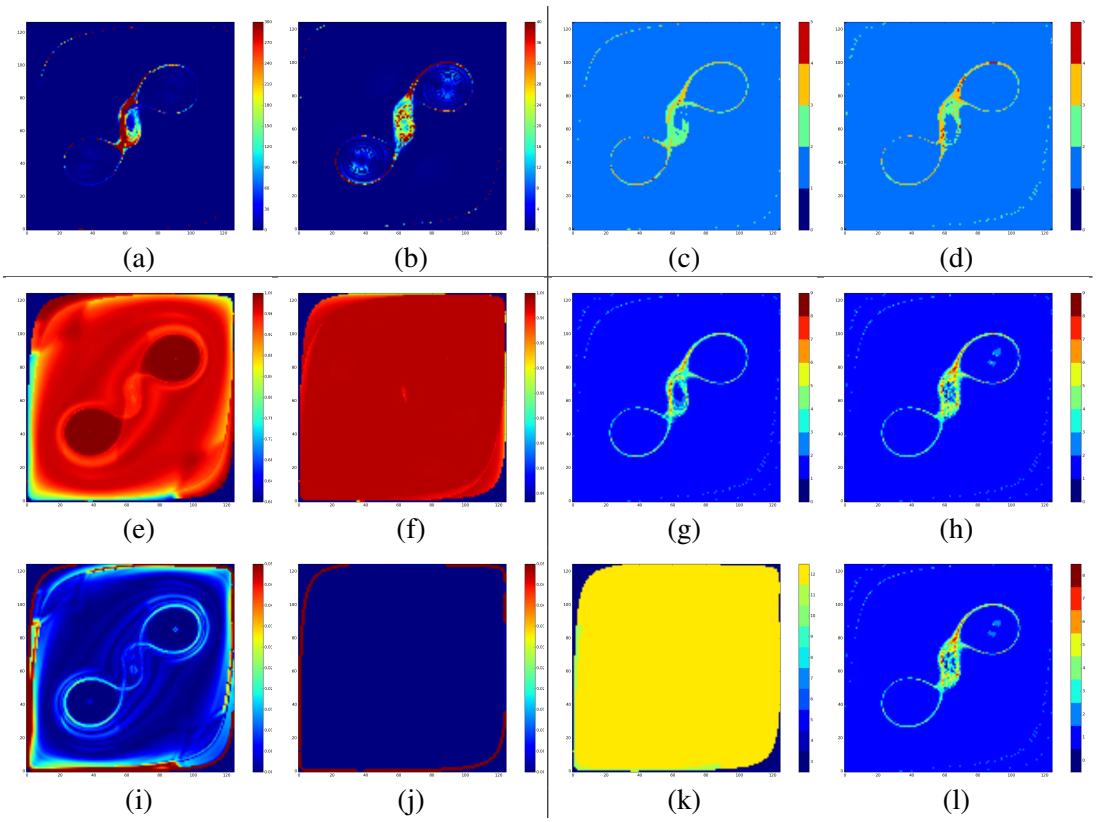
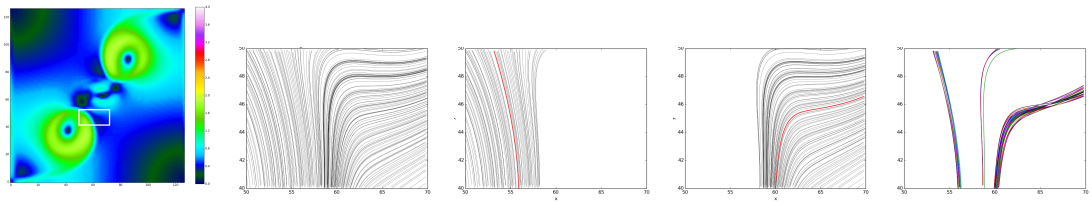
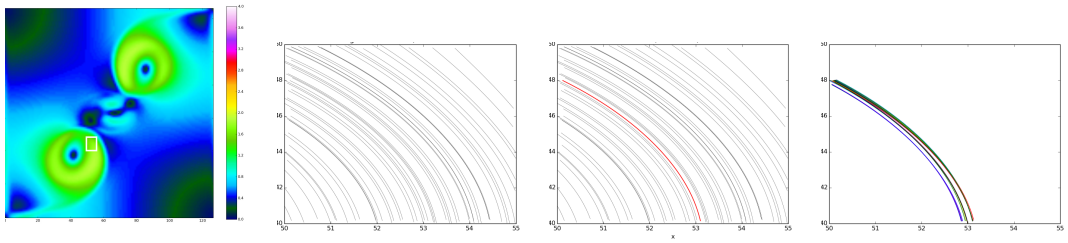


Figure 6.8: Comparison of transport visual summaries for the lock-exchange data set. Methods from [23] are along first row separated by the horizontal line. The vertical line separates entropy maps on the left and cluster results on the right half of the Fig. (a) FTVA for forward integrated streamlines. (b) FTVA for backward integrated streamlines. (c) Number of trend clusters from terminal positions in forward integration. (d) Number of trend clusters from terminal positions in backward integration. (d) Map of average linear streamline entropies for ensemble. (e) Map of average angular streamline entropies for ensemble. (f) Streamline clusters sampled at three points per streamline. (g) Streamline clusters sampled at ten additional points per streamline. (h) Gradient magnitude for linear entropy map. (i) Gradient magnitude for angular entropy map. (j) Sample map, i.e. the map of the points sampled on each streamline for their corresponding seed location. (k) Cluster map for streamlines sampled variably based on entropy. (Note: color bars for sample and cluster maps contains discrete colors labeled from top to bottom in increasing order.)



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

Figure 6.9: Streamline clusters for an incoherent flow region in lock-exchange data set. (a) Region location (shown by white box selected rectangle) from lock-exchange velocity magnitude field. (b) All streamlines from a single member. (c) First cluster from (a) with representative streamline. (d) Second cluster from (b) with representative streamline. Representative streamlines are highlighted in red. (e) Plot of representative streamlines for 20 members, each a random color.



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

Figure 6.10: Streamline clusters for a coherent flow region. (a) Region location (shown by white box selected rectangle) from lock-exchange velocity magnitude field. (b) All streamlines from single member. (c) Single cluster with representative from (b). Representative streamlines are highlighted in red. (d) Plot of representative streamlines for 20 members, each a random color.

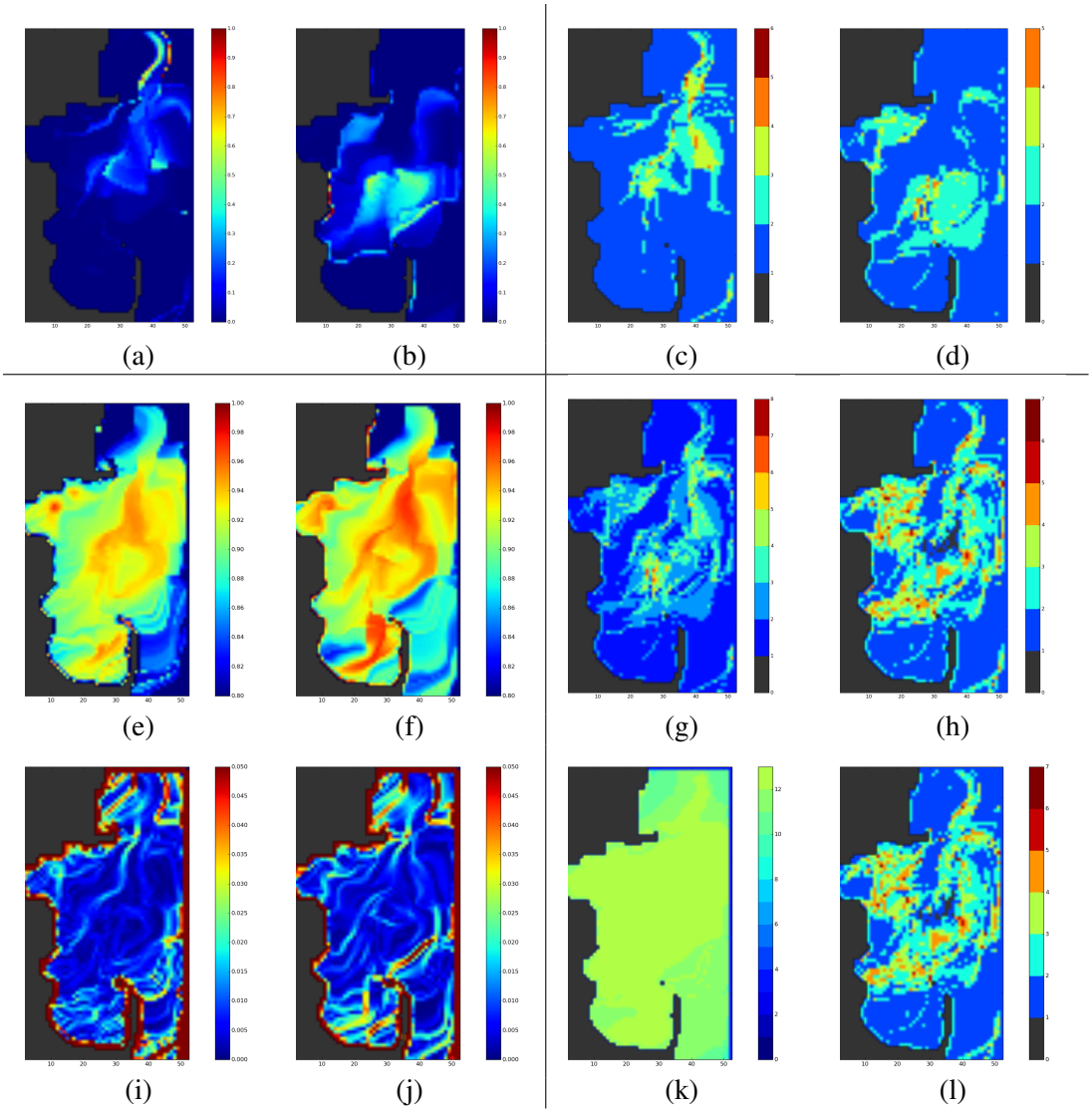


Figure 6.11: Comparison of transport visual summaries for the Massachusetts Bay data set at surface level. Methods from [23] are along first row separated by the horizontal line. The vertical line separates entropy maps on the left and cluster results on the right half of the Fig. (a) FTVA for forward integrated streamlines. (b) FTVA for backward integrated streamlines. (c) Number of trend clusters from terminal positions in forward integration. (d) Number of trend clusters from terminal positions in backward integration. (e) Map of average linear streamline entropies for ensemble. (f) Map of average angular streamline entropies for ensemble. (g) Streamline clusters sampled at three points per streamline. (h) Streamline clusters sampled at ten additional points per streamline. (i) Gradient magnitude for linear entropy map. (j) Gradient magnitude for angular entropy map. (k) Sample map, i.e. the map of the points sampled on each streamline for their corresponding seed location. (l) Cluster map for streamlines sampled variably based on entropy. (Note: color bars for sample and cluster maps contains discrete colors labeled from top to bottom in increasing order.)

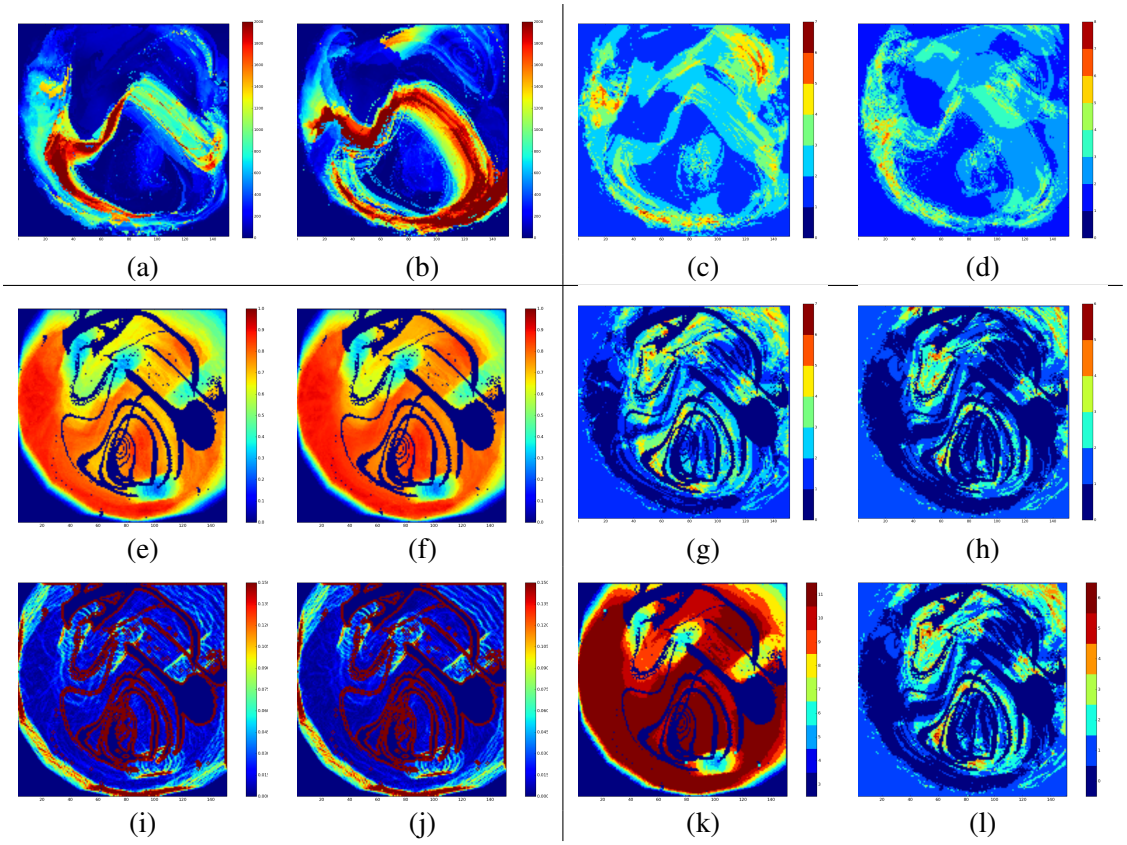


Figure 6.12: Comparison of transport visual summaries for the industrial stirring data set. Methods from [23] are along first row separated by the horizontal line. The vertical line separates entropy maps on the left and cluster results on the right half of the Fig. (a) FTVA for forward integrated streamlines. (b) FTVA for backward integrated streamlines. (c) Number of trend clusters from terminal positions in forward integration. (d) Number of trend clusters from terminal positions in backward integration. (e) Map of average linear streamline entropies for ensemble. (f) Map of average angular streamline entropies for ensemble. (g) Streamline clusters sampled at three points per streamline. (h) Streamline clusters sampled at ten additional points per streamline. (i) Gradient magnitude for linear entropy map. (j) Gradient magnitude for angular entropy map. (k) Sample map, i.e. the map of the points sampled on each streamline for their corresponding seed location. (l) Cluster map for streamlines sampled variably based on entropy. (Note: color bars for sample and cluster maps contains discrete colors labeled from top to bottom in increasing order. Also notice that all fields shown in this Fig. are slightly truncated in their upper right corner from Fig. 6.6c. We use the intersection of the simulation region for all members in the ensemble.)

ensemble. The most interesting aspects of the flow occur at the mixing interface between the two fluids. Figure 6.8a and Fig. 6.8b, show FTVA applied for forward and backward integration, respectively. Figure 6.8c and Fig. 6.8d display the terminal position clusters. Our methods of clustering entire streamlines are shown in Fig. 6.8f (three sample points for all streamlines) and Fig. 6.8g (ten additional sample points along each streamline). Similar flow patterns are seen using both methods, although our method captures aspects of both terminal end point distributions with either streamline sampling frequency. As we increase the sampling rate used in Fig. 6.8f to the one used in Fig. 6.8g, there are areas where cluster counts increase and are not seen using terminal positions alone. These clusters arise due to variations captured by using more samples and thus detect trends of overall streamline geometry.

When consulting the linear entropy map (Fig. 6.8d), the pattern of the flow field where both variance and distinct flow trends emerge is summarized for the lock-exchange data. Small field differences occurring in angular entropy are seen in Fig. 6.8e, providing a nearly constant field except near the transitions in entropy in the upper-left and lower-right corners of the domain. This indicates that the number of clusters and variances we see in the flow occur primarily from variation along streamline lengths, i.e. their linear entropy. However, when both linear and angular entropy inform the sampling frequency for streamlines, the overall higher magnitude of angular entropy (in this example) dominates the influence on sampling frequency for streamlines in the interior of the domain. The cluster map using variable sampling between a minimum of three

samples and a maximum of thirteen samples (Fig. 6.8k), shows a result consistent with the uniform sampling in Fig. 6.8h. (The number samples for streamlines at each seed location is shown in Fig. 6.8j.) Finally, we see from the magnitude of the gradient of the linear entropy map (Fig. 6.8h), that a larger number of clusters are found for streamlines with seed locations near the gradient ridges. Where linear entropy changes over the domain, we see streamline geometry variance over the ensemble members (and thus streamline trends).

We now investigate regions from the lock-exchange simulation domain. We show this for two separate regions using the method outlined in section 6.4.4. We can see regional clustering in Fig. 6.9, for a region exhibiting incoherent flow patters within each member of the region. This is similar to the lower-left quadrant of Fig. 6.5. The flow is simplified using representative streamlines for the region. If we track the streamlines as entering from the bottom of the selected region, some of the representative streamlines flow more from top to bottom that veering to the right or left. Thus, we consider three distinct flow trends from the members of the ensemble for this region.

In contrast, Fig. 6.10 displays more regional coherence of the type shown in the lower-left quadrant of Fig. 6.5. The region has a single representative streamline per member. The summary streamlines also show little variation as shown in Fig. 6.10d. There is coherence both within the region per member, and between members, for a strong overall coherence in the ensemble.

This similarity is different than that shown in Fig. 6.8. In the full-field analysis, we

do not know where in the field trends occur, only that they do for particular seeds. When applying streamline clustering for a region, we show the EVF giving rise to trends seen mapped to seed locations as in Fig. 6.8. However, this insight is limited to the region itself and the trends produced by flow through the region may be mapped to more than one seed, either within the region itself or outside it. We will next show two more data sets, using our method applied to the entire field as we did for the lock-exchange in Fig. 6.8.

Ocean Figure 6.11 is analogous to Fig. 6.8, but shows results for the ocean data. The primary variance occurs in the central region of the simulation domain for both integration directions. This is somewhat intuitive, since streamlines seeded there have the potential to cover a larger area and thus their terminal positions to differ over greater distances. The trend/clustering analysis for terminal points is shown in Fig. 6.11c and Fig. 6.11d, for forward and backward integration respectively.

Our streamline clustering method provides a much higher sensitivity for visualizing trends in the streamlines than conventional FTVA. The number of clusters increase from Fig. 6.11f to 6.11g at the higher sampling frequency. This is due to detecting more variation on the streamlines and seeing a higher resolution of the trends. Figure 6.11a through Fig. 6.11d fail to detect most of the flow behavior that occurs near the upper coastal region and the flow trends present there, i.e. flow bundles that separate along the intermediate positions of the streamlines but have similar positions at their terminal positions. See Fig. 6.7 for an example of this.

In Fig. 6.11d and 6.11e, it can be seen that streamlines seeded centrally have higher average streamline entropy (linear and angular). Again, we see more clusters near the ridges in the gradient magnitudes of Fig. 6.11h and Fig. 6.11i. Near the center of the domain, the number of the clusters drops to zero in Fig. 6.11g (the higher streamline sampling frequency). The lack of trends for these seeds is not seen in Fig. 6.11c and Fig. 6.11d (and in the lower sampling rate of our method in Fig. 6.11f). Our method uncovers the highly variable and chaotic flow mapped to this seeding region. This behavior is also shown when adaptive sampling is applied in Fig. 6.11k.

Industrial Stirring Figure 6.12 applies the same method to the industrial mixing simulation ensemble. As discussed in [23], the design of the stirring machinery shows needed improvement due to the low variance in much of the domain via FTVA. This is corroborated and repeated here in Fig. 6.12a and Fig. 6.12b. The trend analysis from [23] additionally shows much of the domain possessing at least two clusters of terminal particle positions for both the forward and backward integration.

Our method shown in Fig. 6.12d through Fig. 6.12k, sharply contrasts parts of the previous analysis from [23]. We find even in regions of high variance, little evidence of good transport. As can be seen in Fig. 6.12d and Fig. 6.12e, there are irregular domain regions showing very low average linear and angular streamline entropy. (This is most evident in the ovoid structure to the far-right middle section of the domain.) Interestingly, the region along the lower-left of the cylindrical tank possesses high average entropy, but little to no clusters (see Fig. 6.12f, Fig. 6.12g and Fig. 6.12k). This

would appear to contradict the regions with low entropy and also no clusters, except for the fact that we had already observed that regions with a high number of trends generally occur at the ridges of the gradient magnitudes of the entropy maps. We see that this region with high entropy in the lower-left of the domain also exhibits low gradient magnitude (not a region containing a ridge) and thus agrees with the earlier assessment.

There is little difference between the average linear and angular entropy maps for this data set (Fig. 6.12d and Fig. 6.12e) and this signature may be useful for classifying such overall behavior. In regions of the flow that both have low average entropy and low levels of cluster count, we would want to improve the overall transport. This analysis may suggest that a potential geometrical or material design might be implemented to prevent lack of agitation at the fluid and paddle points of contact, since this behavior is consistent across the ensemble where fluids of varying parameters of viscosity were used in the simulation.

6.6 Conclusion

In this chapter, we first presented a flow structure based on streamline clustering over their spatial extent. Using the mean linear and angular streamline entropy maps, we showed that where variations in entropy is greatest, there is in general a correspondingly high number of clusters for those streamlines.

Preliminary results revealed that related methods of trajectory similarity/clustering did not capture the behavior of spatial bifurcation or flow bundling as we had antici-

pated. For example, TRACLUS [34] is a direct extension of DBSCAN to line-segment data. TRACLUS tends to cluster trajectories without regard to individual path integrity, and often finds patterns in partitioned segments of the initial streamlines instead.

We followed our analysis of flow structure by investigating flow coherence at regions of bifurcation in a 2D EVF. Finally, we discussed how both methods can be used in a sequential framework for EVF analysis. The methods presented here are not limited to steady-flow. For the purpose of clarity in this initial study, we chose to focus on a single time-step in the simulation.

Future work will employ better adaptive strategies for streamline sampling frequency and incorporate multiple similarity metrics. Additionally, new methods of region analysis over the entire simulation domain may prove useful via algorithmic versus manual inspection.

Chapter 7

Conclusions and Future Work

Our work has been concerned with summarizing EVF. Such summaries must be created with a particular set of goals, since there is no one all-encompassing view. Our first approach was to show EVF as a field of non-parametric PDF, and then observe EVF uncertainty expressed using velocity density estimation. As a second approach, we treated EVF as separate realizations of which we compare member streamlines. All methods presented in this work have had the purpose of providing analysis and visualization of similarity (or difference) within an EVF.

7.1 Summary

Chapter 3 discussed our interpolation method, *Bivariate Quantile Interpolation*, to address multivariate data for EVF. The method computes faster than other methods (*Displacement Interpolation*), and our method is meant to address issues of EVF vi-

sualization. There are multiple directions to pursue for further investigation of PDF interpolation. This problem is not directly related to ensemble visualization, but visualization research will be the beneficiary of future developments. Some of the areas exposed by our work are discussed in section 7.2.

Chapter 4 applied *Bivariate Quantile Interpolation* to EVF, and compared the results with a Gaussian Mixture Model PDF interpolation. With visualization as the focus, other aspects of general function interpolation were not considered in our studies. We addressed the uncertainty in streamlines that flow through a given location in chapter 5. We provided a method to reduce clutter in the traditional "spaghetti" plot and to rank streamlines from the member realizations based on their probability derived from the vector field PDF.

In chapter 6, we applied streamline clustering to an entire EVF. This analysis and visualization was both for flow through regularly spaced seeds in the EVF and for specified regions. The approach taken for both analyses was similar, and cluster analysis assigned cluster centroids as representative of the cluster streamlines. However, while our analysis for streamlines through single locations over the field provided a full EVF summary, our regional analysis did not. We suggest potential methods for investigating region EVF similarity at the end of the next section.

7.2 Future Work

Gradients provide information about how a field changes over its spatial domain. The gradient of a scalar field produces a vector field. Similarly, vector field gradients are second-order tensor fields. It is not obvious how to express gradients of PDF fields as is now done with finite difference for scalar or vector fields. How might this new type of gradient be calculated? How could the result be used in an analysis of PDF interpolants? Love et al. [39] provide a statistical analysis of operators for multivalue data sets. Such operators are statistical summaries of sample data. Their statistical evaluations show the sensitivity of the operators to a particular data set. Thus, a similar set of statistical measures for the usefulness of a particular interpolation method for multivariate distributions would be useful as well.

Another problem left unsolved is to calculate possible error for non-parametric interpolants. It is likely to be at least approximately the sum of possible error of all the linear interpolation for sample pairs in the non-parameteric interpolants that comprise the KDE's. Once a definition of a PDF gradient is found, there is the application of finding the largest gradients (distance between interpolators) and showing that possible error for interpolation is proportional to that interval, as scalar linear interpolation analysis shows for scalar fields. The primary goal of linear interpolation is to find "least" distance travel for interpolants. If they are scalar, this is a simple Euclidean line. For PDF, we are summarizing a population of values (some of which may be vectors).

EVF exhibit variation between member realizations. Aspects in their variation can

be derived in multiple ways, such as in Finite Time Variance Analysis (FTVA) [23] for the entire field, using Curve Box-plots [43] for flow through a single location in the field, or via the methods of chapters 5 and 6. Another possible approach is to consider all streamlines in the EVF and their intersection with cells of the spatial domain.

This potential algorithm would start with all streamlines seeded at each grid cell from the EVF. An outline is shown in algorithm 5. A streamline should at least have the following meta-data: *seed grid cell id* and a *realization id*. This is represented with a three-tuple, i.e. $(x, y, member)$.

```
foreach streamline in all streamlines do
    foreach point on current streamline do
        lookup cell containing point and record streamline at cell
    end
end
```

Algorithm 5: Algorithm outline for determining streamlines passing through a cell in the EVF.

Gathering the three-tuples for each cell is a precomputation step just as streamline generation, but separate from it. We can then derive various similarity metrics based on a cell's list of three-tuples.

More importantly, we can look at the streamline geometry for the set of streamlines that enter a cell. For example, we could consider the curvature of all streamlines found entering a cell, in the region of the cell. The higher the variance of the curvature, the less agreement we have in the EVF at that cell. Even more simply, instead of curvature, using just the direction vector (could be scalar if 2D flow, i.e. 0 to 360 degrees). This would allow a check whether there is parallel or crossing flow. It is worth exploring

various rendering approaches, but as a first step, HyperLIC [87] may allow the capture of the principal directional flow in a cell for the EVF.

Bibliography

[1] Tatiana Benaglia, Didier Chauveau, David R. Hunter, and Derek Young. mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29, 2009.

[2] Jeff Blimes. A gentle tutorial for the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. International Computer Science Institute, 1998. http://crow.ee.washington.edu/people/bulyko/papers/em.pdf.

[3] Marc Bocquet, Carlos A Pires, and Lin Wu. Beyond gaussian statistical modeling in geophysical data assimilation. *Monthly Weather Review*, 138(8):2997–3023, 2010.

[4] Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. *ACM Transactions on Graphics (TOG)*, 30(6):158, 2011.

[5] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] R. Brecheisen. *Visualization of Uncertainty in Fiber Tracking Based on Diffusion Tensor Imaging*. PhD thesis, Technische Universiteit Eindhoven, 2012. Department of Biomedical Engineering.

[7] K. Broad, J. Leiserowitz, J. Weinkle, and M. Steketee. Misinterpretations of the cone of uncertainty in Florida during the 2004 hurricane season. *American Meterological Society*, pages 651–667, May 2007.

[8] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 263–270. ACM, 1993.

[9] Cheng-Kai Chen, Shi Yan, Hongfeng Yu, Nelson Max, and Kwan-Liu Ma. An illustrative visualization framework for 3d vector fields. In *Computer Graphics Forum*, volume 30, pages 1941–1951. Wiley Online Library, 2011.

[10] Christian M Chilan, M Yang, Albert Cheng, and Leon Arber. Parallel i/o performance study with hdf5, a scientific data package. *TeraGrid 2006: Advancing Scientific Discovery*, 2006.

[11] Andrew Collette. *Python and HDF5*. " O'Reilly Media, Inc.", 2013.

[12] Ismail Demir, Christian Dick, and Rüdiger Westermann. Multi-charts for comparative 3d ensemble visualization. 2014.

[13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based

algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[14] Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, 2002.

[15] Nivan Ferreira, James T Klosowski, Carlos Scheidegger, and Claudio Silva. Vector field k-means: Clustering trajectories by fitting multiple vector fields. *arXiv preprint arXiv:1208.5801*, 2012.

[16] Nivan Ferreira, James T Klosowski, Carlos E Scheidegger, and Cláudio T Silva. Vector field k-means: Clustering trajectories by fitting multiple vector fields. In *Computer Graphics Forum*, volume 32, pages 201–210. Wiley Online Library, 2013.

[17] N.I. Fischer, E. Mammen, and J.S. Marron. Testing for multimodality. *Computational Statistics & Data Analysis*, 18(5):499 – 512, 1994.

[18] Shiho Furuya and Takayuki Itoh. A streamline selection technique for integrated scalar and vector visualization. *Vis Š08: IEEE Visualization Poster Session*, 2(4), 2008.

[19] S Grottel, J Heinrich, D Weiskopf, and S Gumhold. Visual analysis of trajectories in multi-dimensional state spaces. In *Computer Graphics Forum*. Wiley Online Library, 2014.

[20] Hanqi Guo, Xiaoru Yuan, Jian Huang, and Xiaomin Zhu. Coupled ensemble flow line advection and analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2733–2742, 2013.

[21] George Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D: Nonlinear Phenomena*, 149(4):248–277, 2001.

[22] James Helman and Lambertus Hesselink. Representation and display of vector field topology in fluid flow data sets. *Computer*, 22(8):27–36, 1989.

[23] Mathias Hummel, Harald Obermaier, Christoph Garth, and Kenneth I Joy. Comparative visual analysis of lagrangian transport in cfd ensembles. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2743–2752, 2013.

[24] Jeng-Neng Hwang, Shyh-Rong Lay, and Alan Lippman. Nonparametric multivariate density estimation: A comparative study. *IEEE Transactions on Signal Processing*, 42(10):2795–2810, 1994.

[25] V Ilango, R Subramanian, and V Vasudevan. Cluster analysis research design model, problems, issues, challenges, trends and tools. *International Journal on Computer Science and Engineering*, 3(8):3064–3070, 2011.

[26] F. Jiao, J.M. Phillips, Y. Gur, and C.R. Johnson. Uncertainty visualization in HARDI based on ensembles of ODFs. In *Proceedings of the 5th IEEE Pacific Visualization Symposium (PacificVis 2012)*, pages 193–200, February 2012.

[27] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[28] Amin Karami and Ronnie Johansson. Choosing dbscan parameters automatically using differential evolution. *International Journal of Computer Applications*, 91(7):1–11, 2014.

[29] Jens Kasten, Ingrid Hotz, and Hans-Christian Hege. On the elusive concept of lagrangian coherent structures. In *Topological Methods in Data Analysis and Visualization II*, pages 207–220. Springer, 2012.

[30] Alexander Kuhn, Norbert Lindow, Tobias Günther, Alexander Wiebel, Holger Theisel, and Hans-Christian Hege. Trajectory density projection for vector field visualization. *EuroVis-Short Papers*, pages 31–35, 2013.

[31] David H Laidlaw, Robert M Kirby, Cullen D Jackson, J Scott Davidson, Timothy S Miller, Marco Da Silva, William H Warren, and Michael J Tarr. Comparing 2d vector field visualization methods: A user study. *Visualization and Computer Graphics, IEEE Transactions on*, 11(1):59–70, 2005.

[32] Ove Daae Lampe and Helwig Hauser. Interactive visualization of streaming data with kernel density estimation. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 171–178. IEEE, 2011.

[33] Robert S Laramee, Helwig Hauser, Helmut Doleisch, Benjamin Vrolijk, Frits H Post, and Daniel Weiskopf. The state of the art in flow visualization: Dense and

texture-based techniques. In *Computer Graphics Forum*, volume 23, pages 203–221. Wiley Online Library, 2004.

[34] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[35] Pierre FJ Lermusiaux. Uncertainty estimation and prediction for interdisciplinary ocean dynamics. *Journal of Computational Physics*, 217(1):176–199, 2006.

[36] Pierre FJ Lermusiaux, Ching-Sang Chiu, Glen G Gawarkiewicz, Phil Abbot, Allan R Robinson, Robert N Miller, Patrick J Haley, Wayne G Leslie, Sharanya J Majumdar, Alex Pang, et al. Quantifying uncertainties in ocean predictions. Technical report, DTIC Document, 2006.

[37] Jin Li and Andrew Heap. A review of spatial interpolation methods for environmental scientists. Technical Report GeoCat 68229, Geoscience Australia, 2008.

[38] Shusen Liu, Joshua A. Levine, Peer-Timo Bremer, and Valerio Pascucci. Gaussian mixture model based volume visualization. *IEEE Symposium on Large Data Analysis and Visualization*, pages 73–77, 2012.

[39] Alison Love, David L. Kao, and Alex Pang. Visualizing spatial multivalue data. *IEEE Computer Graphics and Applications*, 25(3):69–79, May/June 2005.

[40] Kewei Lu, Abon Chaudhuri, Teng-Yok Lee, Han-Wei Shen, and Pak Chung

Wong. Exploring vector fields with distribution-based streamline analysis. In *PacificVis*, pages 257–264, 2013.

[41] Stephane Marchesin, Cheng-Kai Chen, Chris Ho, and Kwan-Liu Ma. View-dependent streamlines for 3d vector fields. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1578–1586, 2010.

[42] J. Martin, E. Swan II, R. Moorehead, Z. Liu, and S. Cai. Results of a user study on 2D hurricane visualization. *Eurographics/IEEE Symposium on Visualization*, 27(3):991–998, 2008.

[43] Mahsa Mirzargar, Ross Whitaker, and Robert Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. 2014.

[44] Bart Moberts, Anna Vilanova, and Jarke J van Wijk. Evaluation of fiber clustering methods for diffusion tensor imaging. In *Visualization, 2005. VIS 05. IEEE*, pages 65–72. IEEE, 2005.

[45] Donald Myers. Spatial interpolation: An overview. *Geoderma*, pages 17–28, 1994.

[46] Regina Nuzzo. Statistical errors. *Nature*, 506(13):150–152, 2014.

[47] Harald Obermaier and Kenneth I Joy. Future challenges for ensemble visualization. *Computer Graphics and Applications, IEEE*, 34(3):8–11, 2014.

[48] M. Otto, T. Germer, and H. Theisel. Uncertain 2D vector field topology. *Computer Graphics Forum (Proceedings of Euro-graphics 2010, Norrkpping, Sweden)*, 29(2):347–356, 2010.

[49] M. Otto, T. Germer, and H. Theisel. Uncertain topology of 3D vector fields. *Pacific Visualization Symposium (PacificVis)*, pages 67–74, 2011.

[50] Mathias Otto, Tobias Germer, Hans-Christian Hege, and Holger Theisel. Uncertain 2d vector field topology. In *Computer Graphics Forum*, volume 29, pages 347–356. Wiley Online Library, 2010.

[51] Mathias Otto, Tobias Germer, and Holger Theisel. Uncertain topology of 3d vector fields. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 67–74. IEEE, 2011.

[52] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, pages 1065–1076, 1962.

[53] Thomas Peacock and George Haller. Lagrangian coherent structures: The hidden skeleton of fluid flows. *Physics today*, 66(2):41–47, 2013.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[55] Christopher Petz, Kai Pöthkow, and Hans-Christian Hege. Probabilistic local features in uncertain vector fields with spatial correlation. *Computer Graphics Forum*, 31(3):1045–1054, 2012.

[56] T. Pfaffelmoser, M. Reitinger, and R. Westermann. Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. *Eurographics/IEEE Symposium on Visualization (EuroVis 2011)*, 30(3):951–960, 2011.

[57] T. Pfaffelmoser and R. Westermann. Visualization of global correlation structures in uncertain 2D scalar fields. *Computer Graphics Forum*, 31:1025–1034, 2012. doi: 10.1111/j.1467-8659.2012.03095.x.

[58] M. Phadke, L. Pinto, O. Alabi, J. Harter, R. Taylor, X. Wu, H. Petersen, S. Bass, and C. Healy. Exploring ensemble visualization. *VDA*, pages 82940B–82940B–12, 2012.

[59] Madhura N Phadke, Lifford Pinto, Oluwafemi Alabi, Jonathan Harter, Russell M Taylor II, Xunlei Wu, Hannah Petersen, Steffen A Bass, and Christopher G Healey. Exploring ensemble visualization. In *IS&T/SPIE Electronic Imaging*, pages 82940B–82940B. International Society for Optics and Photonics, 2012.

[60] Kilian M. Pohl, John Fisher, Sylvain Bouix, Martha Shenton, Robert W. McCarley, W. Eric Grimson, Ron Kikinis, and William M. Wells. Using the logarithm of odds to define a vector space on probabilistic atlases. *Medical Image Analysis*, 11:465–477, 2007.

[61] K. Pöthkow and Hege H. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1393–1406, October 2011.

[62] Kai Pöthkow and Hans-Christian Hege. Nonparametric models for uncertainty visualization. In *Computer Graphics Forum*, volume 32, pages 131–140. Wiley Online Library, 2013.

[63] Kai Pöthkow, Britta Weber, and Hans-Christian Hege. Probabilistic marching cubes. In *Proceedings of the 13th Eurographics / IEEE - VGTC conference on Visualization*, EuroVis'11, pages 931–940, Aire-la-Ville, Switzerland, Switzerland, 2011. Eurographics Association.

[64] K. Potter, A. Wilson, P. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. Johnson. Ensemble-Vis: A framework for the statistical visualization of ensemble data. In *IEEE Workshop on Knowledge Discovery from Climate Data: Prediction, Extremes.*, pages 233–240, 2009.

[65] Kristin Potter, Robert M. Kirby, Dongbin Xiu, and Chris R. Johnson. Interactive visualization of probability and cumulative density function. *International Journal for Uncertainty Quantification*, 2(4):397 – 412, 2012.

[66] Kristin Potter, Paul Rosen, and Chris R Johnson. From quantification to visualization: A taxonomy of uncertainty visualization approaches. In *Uncertainty Quantification in Scientific Computing*, pages 226–249. Springer, 2012.

[67] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[68] A.L. Read. Linear interpolation of histograms. *Nuclear Instruments and Methods in Physics Research*, pages 357–360, 1999.

[69] Wieland Reich and Gerik Scheuermann. Analysis of streamline separation at infinity using time-discrete markov chains. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2140–2148, 2012.

[70] Murray Rosenblatt et al. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

[71] Diana Röttger, Daniela Dudai, Dorit Merhof, and Stefan Müller. Bundle visualization strategies for hardi characteristics. In *Advances in Visual Computing*, pages 326–335. Springer, 2012.

[72] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

[73] Filip Sadlo and Ronald Peikert. Visualizing lagrangian coherent structures and comparison to vector field topology. In *Topology-Based Methods in Visualization II*, pages 15–29. Springer, 2009.

[74] Tobias Salzbrunn, Christoph Garth, Gerik Scheuermann, and Joerg Meyer. Pathline predicates and unsteady flow structures. *The Visual Computer*, 24(12):1039–1051, 2008.

[75] Jibonananda Sanyal, Song Zhang, Jamie Dyer, Andrew Mercer, Philip Amburn, and Robert Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, 2010.

[76] Jibonananda Sanyal, Song Zhang, Jamie Dyer, Andrew Mercer, Philip Amburn, and Robert J Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1421–1430, 2010.

[77] Themistoklis P Sapsis and Pierre FJ Lermusiaux. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Physica D: Nonlinear Phenomena*, 238(23):2347–2360, 2009.

[78] S. Schlegel, N. Korn, and G. Scheuermann. On the interpolation of data with normally distributed uncertainty for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2305 – 2314, December 2012.

[79] Dominic Schneider, Jan Fuhrmann, Wieland Reich, and Gerik Scheuermann. A variance based ftle-like method for unsteady uncertain vector fields. In *Topo-*

logical Methods in Data Analysis and Visualization II, pages 255–268. Springer, 2012.

[80] Christian Schoelzel, Petra Friederichs, et al. Multivariate non-normally distributed random variables in climate research–introduction to the copula approach. *Nonlin. Processes Geophys.*, 15(5):761–772, 2008.

[81] David W Scott. Feasibility of multivariate density estimates. *Biometrika*, 78(1):197–205, 1991.

[82] David W Scott. *Multivariate density estimation: theory, practice, and visualization*, volume 383. John Wiley & Sons, 2009.

[83] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[84] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[85] A. Slingsby, J. Strachan, P. Vidale, and J. Dykes. Discovery exhibition: Making hurricane track data accessible. http://www.discoveryexhibition.org/uploads/Entries/Slingsby_2010_HurricaneTrackData.pdf, 2010. Discovery Exhibition.

[86] MP Ueckermann, Pierre FJ Lermusiaux, and TP Sapsis. Numerical schemes for

dynamically orthogonal equations of stochastic fluid and ocean flows. *Journal of Computational Physics*, 233:272–294, 2013.

[87] Xiaoqiang Zheng and Alex Pang. Hyperlic. In *Visualization, 2003. VIS 2003. IEEE*, pages 249–256. IEEE, 2003.

[88] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. CRC Press, 2010.