Extracting Jacobi Structures in Reeb Spaces

Amit Chattopadhyay\textsuperscript{1}, Hamish Carr\textsuperscript{1}, David Duke\textsuperscript{1} and Zhao Geng\textsuperscript{1}

\textsuperscript{1}School of Computing, University of Leeds, UK

Abstract
Jacobi sets have been identified as significant in multi-field topological analysis, but are defined in the domain of the data rather than in the Reeb Space. This distinction is significant, as exploiting multi-field topology actually depends on the projection of the Jacobi set into the Reeb Space, and the details of its internal structure. We therefore introduce the Jacobi Structure of a Reeb Space which describes this, explain its relationships with both the Jacobi Set and Fiber Analysis in mathematical topology, give an algorithm for computing the Jacobi Structure recursively using a Multi-Dimensional Reeb Graph and illustrate it using an early implementation in VTK.

Keywords: Topological analysis, Multi-Field, Reeb space, Jacobi set, Multi-Dimensional Reeb Graph, JCN

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types

1. Introduction
Much of the data that appear in scientific experiments and simulations are multivariate in nature and involve multiple scalar fields. Topological analysis of such data aims to reveal interesting features useful to the domain scientists. Until now, multi-field topological analysis is primarily based on Jacobi sets—a generalization of critical features in domains of multi-fields [EH04, EHNP04, BBD\textsuperscript{+}07, NN09]. Although a Reeb space structure has been introduced as an extension of the Reeb graph for capturing the multi-field topology [EHP08], there is a lack of efficient data-structure and algorithm for computing Reeb spaces in practical applications. Unlike the Reeb graph, the Reeb space structure is not simple, but it is quite predictable that the Reeb space is much more valuable than the Jacobi set for the multi-field topological analysis. Recently, a Joint Contour Net (JCN) graph structure has been proposed for approximating the Reeb space [CD13, DCS\textsuperscript{+}12]. However, a clear relationship between the Jacobi set and the Reeb space is still missing and worth investigating for understanding the Reeb space.

In this article, we introduce a Jacobi Structure in the Reeb space which relates the Jacobi set with the Reeb space. The Jacobi structure is characterised as the projection of the Jacobi set in the Reeb space and captures the exact location of topological changes. Clearly, computing the Jacobi structures helps understanding the Reeb spaces and fiber-topology of multi-fields.

Contribution. In this article, we contribute as follows:

\begin{itemize}
  \item We introduce the Jacobi structures as the critical features in Reeb spaces and relate them with the Jacobi sets and the singular fibers of multi-fields.
  \item We provide an efficient algorithm for computing the Jacobi structure by constructing a Multi-Dimensional Reeb Graph (MDRG) from the Joint Contour Net, itself a quantized approximation of the Reeb space.
  \item We report preliminary implementation results for simulated multi-field data sets, using the VTK library.
\end{itemize}

Outline. In the next section, we give a necessary background and an overview of the multi-field topology. In section 3, we describe our algorithm for computing the Jacobi structures in the Reeb spaces. Finally, section 4 shows some implementation results of the Jacobi structures from some simulated data and section 5 concludes with a summary.

2. Necessary Background
Over the last two decades, scalar topology has been used to support scientific data analysis and visualization, in particular through the use of the Reeb graph and its specialization, the contour tree [vKvOB\textsuperscript{+}97, CSA03, HSKK01, EHMP04, PBMS07]. The subject of multi-field topology in the data analysis is rather new. In this section we briefly describe the multi-field topological analysis and the existing tools for capturing them, such as the Jacobi sets and the Reeb spaces. In the end, we introduce the Joint Contour Net data-structure that approximates the Reeb space.
**Multi-Field Analysis.** A multi-field on \( \mathbb{R}^d \) with \( r \) component scalar fields \( f_i : \mathbb{R}^d \to \mathbb{R} \) is a map \( f = (f_1, f_2, \ldots, f_r) : \mathbb{R}^d \to \mathbb{R}^r \). In differential topology, \( f \) is considered to be a **smooth map** where all its partial derivatives of any order are continuous. A point \( x \in \mathbb{R}^d \) is called a **singular point** (or critical point) of \( f \) if the rank of its differential \( df_x \) is less than \( \min\{d, r\} \) where \( df_x \) is the \( d \times r \) matrix whose columns are the gradients of \( f_1 \) to \( f_r \) at \( x \). And the corresponding value \( f(x) = c = (c_1, c_2, \ldots, c_r) \) in \( \mathbb{R}^r \) is a **singular value**, otherwise \( x \) is called a regular point and \( f(x) \) is a regular value.

A **fiber** or a level set of a map corresponding to a value \( c \) is the inverse image \( f^{-1}(c) \). The inverse image of a singular value is called a singular fiber and the inverse image of a regular value is called a regular fiber. If each fiber of a smooth map contains at most one singular point then it is called a stable map, otherwise if at least one fiber contains more than one singular point then the corresponding map is an unstable map [Sae04]. Figure 1a is an example of a stable map, as here each singular fiber is a singular point. Figure 1c is an example of an unstable map, because of the singular fibers which pass through four saddles. From the pre-image theorem [GP74], generically a regular fiber \( f^{-1}(c) \) is a \( (d - r) \)-manifold for the regular value \( c \). We note for \( d < r \), \( f^{-1}(c) \) is an empty set. Again a fiber \( f^{-1}(c) \) can be considered as the intersection of the fibers of the component scalar fields \( f_1^{-1}(c_1), f_2^{-1}(c_2), \ldots, f_r^{-1}(c_r) \) and a connected component of this intersection is called a joint contour [CD13]. Alternatively, the joint contours \((f_1, f_2, \ldots, f_r)\) can be considered as the contours of a component field \( f_i \), restricted to the joint contours of the remaining component fields. This is a key idea that we use in building our MDRG data-structure. Topology of the singular fibers and their complete classifications have been studied in the literature for smooth maps from \( \mathbb{R}^3 \) to \( \mathbb{R}^2 \) and from \( \mathbb{R}^4 \) to \( \mathbb{R}^3 \) [Sae04].

**Jacobi Set.** The Jacobi set \( J_f \) of a multi-field \( f : \mathbb{R}^d \to \mathbb{R}^r \) is defined by the set \( J_f := \{ x \in \mathbb{R}^d \mid \text{rank } df_x < \min\{d, r\} \} \) [EH04]. In otherwords, this is the set of singular points of the multi-field. An alternative way to describe the Jacobi set is by the set of critical points of one component field (say \( f_i \)) of \( f \) restricted to the intersection of the level sets of the remaining component fields. Edelsbrunner et al. studied properties of the Jacobi set for \( r \) Morse functions [EH04]. The Jacobi set is symmetric with respect to its component fields. Generically, the Jacobi set of two Morse functions is a smoothly embedded 1-manifold where the gradients of the functions become parallel. Although, in general Jacobi sets are not sub-manifolds of the domain of the multi-field \( f \), but are the disjoint union of sub-manifolds of the domain [EH04]. The red dotted lines in Fig 1a and Fig 1c show the Jacobi sets of a stable and an unstable multi-field, respectively.

**Reeb Space.** As the Reeb graph of a scalar field, the Reeb space parametrizes the joint contours of a multi-field. In other words, each point in the Reeb space corresponds to a joint contour of the multi-field and vice versa. The Reeb space of a multi-field \( f = (f_1, f_2, \ldots, f_r) : \mathbb{R}^d \to \mathbb{R}^r \) is denoted as \( \text{RS}[f_1, f_2, \ldots, f_r] \). Generically, when \( r \leq d \) the Reeb space of \( f \) consists of a collection of \( r \)-manifolds glued together in complicated ways [EHP08]. Fig 1b and Fig 1d show two examples of the Reeb spaces corresponding to a stable and an unstable bivariate-fields, respectively. We indicate the dark (red) lines in the Reeb spaces, along which the regular sheets are glued, as the Jacobi structures and formally introduce them in the next section.

**Joint Contour Net.** The Joint Contour Net (JCN) [CD13, DCS*12] is a data structure that approximates the Reeb space of a Piecewise-Linear (PL) multi-field \( f = (f_1, f_2, \ldots, f_r) : I \subset \mathbb{R}^d \to \mathbb{R}^r \) in a \( d \)-dimensional interval (box) \( I \), and is denoted by \( \text{JCN}[f_1, f_2, \ldots, f_r] \). The idea of the JCN is based on the quantization of the level sets of the fields into discrete ones. A **quantized level set** of \( f_i \) at an isovalue \( h \in \mathbb{Z} \) is denoted by \( Qf_i^{-1}(h) \) and is defined as: \( Qf_i^{-1}(h) := \{ x \in M : \text{round}(f_i(x)) = h \} \). A connected component of the quantized level set in the mesh is called a quantized contour or a contour slab. The part of the contour slab in a single cell of the mesh is called a contour fragment. Now the first step of the JCN algorithm constructs all the contour fragments corresponding to a quantization of each component field. In the second step, the joint contour fragments are computed by computing the intersections of these contour fragments for the component fields in a cell. The third step is to construct an adjacency graph of these joint contour fragments, a node in the graph corresponds to a joint contour.
and show the correspondence between the Jacobi structure of a PL multi-field. Alternatively, the Jacobi set from the domain to the Reeb space. The Jacobi structure is the set of points in the Reeb space. Then we give an algorithm for computing the Jacobi structure for a PL multi-field. Our method for computing the Jacobi structure of a PL multi-field \( f = (f_1, f_2, \ldots, f_r) \) can be described in three steps:

1. Compute the JCN of the multi-field as an approximation of its Reeb space (as demonstrated by Figure 2(a)-(b)),
2. Construct a Multi-Dimensional Reeb Graph (MDRG) from the JCN (as demonstrated by Figure 2(c)),
3. Finally, detect the Jacobi structure in JCN as the critical nodes of the MDRG (as demonstrated by Figure 2(d)).

The first step - computation of the JCN - has already been outlined in the last section. Next we describe the construction of the MDRG structure which is the core of our method.

### 3. Our Method

In this section, first we define a Jacobi structure in the Reeb space. Then we give an algorithm for computing the Jacobi structure for a PL multi-field \( f := (f_1, f_2, \ldots, f_r) : I \subset \mathbb{R}^d \rightarrow \mathbb{R}^r \), using the JCN. In the process we also build a new Multi-Dimensional Reeb Graph data-structure.

#### Definition 3.1

The Jacobi Structure of a Reeb space corresponding to a multi-field is defined as the projection of its Jacobi set from the domain to the Reeb space. Alternatively, the Jacobi structure is the set of points in the Reeb space that parametrizes the singular fibers of the multi-field.

Figure 1a-1b and 1c-1d show the correspondence between the components of the Jacobi set and the Jacobi structure. Our method for computing the Jacobi structure of a PL multi-field \( (f_1, f_2, \ldots, f_r) \) can be described in three steps:

1. Compute the JCN of the multi-field as an approximation of its Reeb space (as demonstrated by Figure 2(a)-(b)),
2. Construct a Multi-Dimensional Reeb Graph (MDRG) from the JCN (as demonstrated by Figure 2(c)),
3. Finally, detect the Jacobi structure in JCN as the critical nodes of the MDRG (as demonstrated by Figure 2(d)).

The first step - computation of the JCN - has already been outlined in the last section. Next we describe the construction of the MDRG structure which is the core of our method.

#### The Multi-Dimensional Reeb Graph

The MDRG gives a new hierarchical (tree) structure for packing the joint contours of the component fields corresponding to a multi-field \( (f_1, f_2, \ldots, f_r) \). Each dimension (or hierarchy or level) of the MDRG incrementally adds a new component field to form the joint contours of that dimension. In the first dimension, it packs the contours of the component \( f_1 \). In the second dimension, it packs the joint contours of \( (f_1, f_2) \) (alternatively, the contours of \( f_2 \), restricted to the contours of \( f_1 \)). In the third hierarchy, it packs the joint contours of \( (f_1, f_2, f_3) \) (alternatively, the contours of \( f_3 \), restricted to the joint contours of \( (f_1, f_2) \)). And this is continued until the \( r \)-th dimension to obtain a \( r \)-dimensional MDRG of \( f \). Now to pack the joint contours at each dimension, the MDRG stores the corresponding Reeb graphs, recursively. That is, in the first dimension it stores the Reeb graph (or contour tree in case of a simple domain) of \( f_1 \). In the second dimension, it stores the Reeb graphs of \( f_2 \), restricted to the contours of \( f_1 \). In the third dimension, it stores the Reeb graphs of \( f_3 \), restricted to the joint contours of \( (f_1, f_2) \) and so on.

In Figure 2(c), we show an example of the MDRG, constructed from the JCN of a PL bivariate field, i.e. for \( r = 2 \). First we extract the Reeb graph (contour tree, in this example) from the JCN with respect to the field \( f_1 \) (the algorithm is discussed later). This Reeb graph is denoted by \( R^1 \) and is packed in the first dimension of the MDRG. Note that each node (vertex) of \( R^1 \) corresponds to a contour slab of \( f_1 \) and is a collection of nodes in the JCN with same \( f_1 \)-value. Let \( R^1 \) has \( n \) vertices \( v_1, v_2, \ldots, v_n \). Next corresponding to each vertex of \( R^1 \) again we construct a Reeb graph with respect to field \( f_2 \). These Reeb graphs are denoted as \( R^2_1, R^2_2, \ldots, R^2_n \) and are packed in the second-dimension of the MDRG (collectively, denoted as \( R^2 \) in Figure 2(c)). For number of fields \( r > 2 \) this procedure is continued, recursively, to obtain a \( r \)-dimensional MDRG.

#### Definition 3.2

The critical nodes of a MDRG corresponding to a multi-field \( (f_1, f_2, \ldots, f_r) \) are given by the critical nodes of the Reeb graphs in the \( r \)-th dimension of the MDRG. Note, a node of a Reeb graph is critical if either of its indegree or outdegree is not one. Thus from the construction, the critical nodes of the MDRG correspond to the singular...
fibers of the multi-fields and in the Reeb space they form the Jacobi structure. Thus we have the following lemma.

**Lemma 3.1 (Convergence)** In the limiting case, when the quantization level increases, the set of critical nodes of the MDRG computed from a JCN converges to the accurate Jacobi structure of the Reeb space.

**Algorithm 1**: `CreateReebGraph(G, f_i)`

*Input:* A subgraph `G` of JCN and a chosen field `f_i`.

*Output:* The Reeb Graph `RG` with respect to field `f_i`.

1. Create Union-Find Structure `UF` for field `f_i`.
2. For each adjacent `g_1, g_2 ∈ G` with `f_i(g_1) = f_i(g_2)`, `UFAdd(g_1, g_2)`.
3. For each component `C_i` in `UF` do:
   - Create a node `n_{C_i}` in `RG`.
   - Map graph node-id(s) and field-values from `G` to `n_{C_i}`.
4. Order nodes `n_{C_1}, ..., n_{C_n}` according to `f_i` field values.
5. For each edge `e_{1,2}` in `G` do:
   - If `f_i(e_1) ≠ f_i(e_2)` then:
     - Add edge `e(n_{C_1}, n_{C_i})` in `RG` if not already present.
   - End if
6. End for
7. Return `RG`.

**Algorithm.** The algorithm for constructing the MDRG from an input JCN outputs a tree data-structure, with each level storing the Reeb graphs of that dimension. We start with the JCN and compute the Reeb Graph for property `f_1` by performing union-find (UF) processing over the nodes of the JCN. This breaks the JCN into subgraphs corresponding to slabs in the Reeb Graph of property `f_1`. The MDRG for each subgraph is then computed recursively, and stored in the node of the parent Reeb Graph to which its slab corresponds. In the process, the slabs get separated out into smaller and smaller components. The Reeb graph computation for a subgraph `G` of JCN and a chosen field `f_i` is shown in Algorithm 1. The complexity of the CreateReebGraph on a graph with `n` nodes is `O(n + p log N)` which is the complexity of a sequence of `p` UF operations (here, `p ≤ n`) [Tar75].

![Figure 3](image1.png)  
Figure 3: The Jacobi structures (shown in red color in JCN) of the bivariate fields: (a) (Parab, Height), (b) (Parab, Sphere).

![Figure 4](image2.png)  
Figure 4: (a) JCN of the nuclear scission data (b) Jacobi Structure ('red' nodes) (c) Selected components of the Jacobi Structure (d) Corresponding critical feature in the domain (e) Topological change at the Jacobi component—Nuclear Scission.

### Table 1: Performance results for some simulated datasets

<table>
<thead>
<tr>
<th>Field</th>
<th>Time (MDRG)</th>
<th>Time (JCN)</th>
<th>Slab Widths</th>
<th>Data-Dims</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Circle, Line)</td>
<td>53.24s</td>
<td>0.06s</td>
<td>35</td>
<td>(1, 1)</td>
<td>(40, 40, 40)</td>
</tr>
<tr>
<td>(Parab, Height)</td>
<td>1308s</td>
<td>0.12s</td>
<td>(1, 1)</td>
<td>(40, 40, 40)</td>
<td>(40, 40, 40)</td>
</tr>
<tr>
<td>(Parab, Sphere)</td>
<td>131.50s</td>
<td>0.22s</td>
<td>(1, 1)</td>
<td>(40, 40, 40)</td>
<td>(40, 40, 40)</td>
</tr>
</tbody>
</table>

**4. Implementation**

We implement our algorithm of computing the Jacobi structures using the Visualization Toolkit (VTK) [vtk] and an Open Graph Drawing Framework (OGDF) [ogd]. To justify the correctness of the algorithm, the algorithm is tested on simulated data generated from known functions - Circle: `x^2 + y^2`, Line: `y`, Sphere: `x^2 + y^2 + z^2`, Parab: `x^2 + y^2 - z`, Height: `z` and Cubic: `y^3 - xy + z^2`. Circle and Line data are in `[-5, 5]^2` and other data are considered in `[-5, 5]^3`. Figure 3 shows the extracted Jacobi structures (red nodes) of the bivariate fields: (Parab, Height) and (Parab, Sphere). The Jacobi structures in the figures are clearly showing the exact locations of the topological changes in the corresponding JCN. The size of each JCN node corresponds to the size of the joint contour slab in the geometric domain.

We also apply our algorithm on a real density data of protons and neutrons in the nucleus (see [DCS*12] for the data details) and detect the ‘exact’ location of the nuclear scission point - a topological feature where one nucleus fragments into two. This is shown in Figure 4(a)-(e).

**Performance results.** Table 1 shows performance results of the JCN and the MDRG implementations. All timings were performed on a 3.06 GHz 6-Core Intel Xeon with 64GB memory, running OSX 10.8.5, and using VTK 5.10.1.

### 5. Conclusion

In this article, we introduce a new Jacobi structure in the Reeb space which is important for detecting the exact location of topological changes in the multi-field data. We have also proposed a new data-structure and an algorithm for computing the Jacobi structures corresponding to general PL multi-fields.

© The Eurographics Association 2014.
Acknowledgments
The first author of the paper thanks Daisuke Sakurai and Shigeo Takahashi for their feedback during constructing the examples of the Reeb spaces in the Background section.

References


© The Eurographics Association 2014.