ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.33, No-1-2, 2025

Simplifying Jacobi Sets' Topology and Geometry by Selective Smoothing of Bivariate 2D Scalar Fields

Felix Raith
Leipzig University & Dept. of Environmental Informatics, Helmholtz Center for Environmental Research
Leipzig, Germany
raith@informatik.uni-leipzig.de

Gerik Scheuermann
Institute for Computer Science & ScaDS.AI
Leipzig University
Leipzig, Germany
scheuermann@informatik.uni-leipzig.de

Christian Heine
Institute for Computer Science
Leipzig University
Leipzig, Germany
heine@informatik.uni-leipzig.de

ABSTRACT

The topological analysis of multivariate fields is vital when investigating the relationship between functions. Jacobi sets, the set of all points at which the gradients of the functions are linearly dependent, are an essential tool for such analyses, as they extend the notion of critical points from scalar fields to multivariate fields. However, the Jacobi sets can become very complex, in particular, due to numerical errors and noise. These problems occur in practice, such as in eddy detection on sea surfaces. Although several methods for simplifying Jacobi sets exist in the literature, they mainly reduce Jacobi sets visually without adjusting the function values, which is essential for further data processing. This paper introduces a novel algorithm that changes the values of functions in a 2D bivariate scalar field, resulting in simplified Jacobi sets. For this, we use a neighborhood graph to identify the Jacobi sets to simplify, visualize the complexity of the Jacobi set for real-world examples, and compare the results with prior work. The new approach preserves features better and simplifies the geometry of the Jacobi sets by reducing zigzag patterns.

Keywords

Jacobi set, topological data analysis, bivariate data, topological simplification.

1 INTRODUCTION

Topological analysis methods are widely used, for example, in structural mechanics, to investigate the relationships between physical quantities such as pressure and temperature. Critical points, in particular, have been established as essential tools when analyzing scalar data sets. However, the representation of complex dependencies between individual scalar data sets is often limited.

Jacobi sets have emerged as a central approach for analyzing bivariate correlations [EH04]. Jacobi sets extend the concept of critical points from scalar to multivariate data. The applicability is multi-layered, ranging from the extraction of fiber surfaces [SN22] to the use as a fiber surface control polygon [RNSH25], automatic tree ring detection [MOQ*20], and more [NB13, AAM17, ISLDF16]. An intuitive interpretation of Jacobi sets is important here. However, challenges arise when numerical errors or noise increase the complexity of Jacobi sets and make it difficult for experts to analyze them (see Figure 7a). One approach to reducing Jacobi sets' complexity is simplifying them [SN09]. However, this simplification only refers to the graphical representation of the Jacobi sets without changing the function values, which are essential for further analysis. Smoothing filters change the function values, but they do so globally, often losing important structural information [RSH24]. Raith et al. [RSH24] have developed a method that simplifies the Jacobi sets by collapsing cells from connected components of the Jacobi sets. This approach shows misinterpretation in the assignment of degenerate cells in symmetric data. Bhatia et al. [BWN*15] propose a theoretical approach to simplify Jacobi sets by reducing the complexity of loops and zigzag patterns.

Based on Raith et al. [RSH24] and the theoretical approach of Bhatia et al. [BWN*15], we propose a new method to simplify Jacobi sets of piecewise linear bivariate 2D scalar fields, which adapts the function values. This new method simplifies the function's topology by removing connected components of the Jacobi set and the geometry of Jacobi sets by reducing zigzag patterns, thus reducing their complexity. A neighborhood graph contains additional information (see Figure 5) to identify the connected components of the Jacobi set and zigzag patterns to be removed. These are then removed using an algorithm inspired by loop subdivision.

The main contribution of this work includes:

- A new algorithm to simplify the Jacobi set topology by removing connected components of Jacobi sets.
- A new algorithm to simplify the Jacobi set geometry by removing zigzag patterns.
- An interactive visualization of the neighborhood graph to analyze and visualize Jacobi set complexity.

We compare the resulting Jacobi sets visually and by using the metrics of Raith et al. [RSH24] with the original data and the Collapse Algorithm from Raith et al.

2 RELATED WORK

Heine et al. [HLH*16] give an overview of topology-based methods in visualization, while He et al. [HTWL19] focus on multivariate data visualization.

In scalar field topology, essential features of scalar fields are represented by structures such as contour trees [CSA03, CSVDP10, VKvOB*97], Reeb graphs [Ree46], or the Morse-Smale complex [EHZ03]. These structures enable topological simplification, such as the removal of topological features in contour trees [CSVDP04], the removal of pairs of critical points in the Morse-Smale complex [BHEP04] or the simplification of critical points on point clouds [LSW09]. Edelsbrunner et al. [ELZ02] introduced persistent homology to simplify point clouds. Cohen-Steiner et al. [CSEH05] extended this to persistence diagrams.

However, it is challenging to apply these methods directly to multifield data. Singh et al. [SMC*07] addressed this by partial clustering on high-dimensional data. Carr and Duke [CD13] developed the Joint Contour Net to generalize contour trees. Chattopadhyay et al. [CCDG14] and Strodthoff and Jüttler [SJ15] studied Reeb graphs for multifield data. Tierny and Carr [TC16] compute the Reeb space of a bivariate piecewise linear scalar function. Chattopadhyay et al. [CCD*16] further simplified multivariate data using joint contour networks and Reeb skeletons.

The focus was less on Jacobi set simplification. An early work on this is noise reduction in Jacobi sets for time-varying data by Bremer et al. [BBD*07]. Suthambhara and Natarajan [SN09] applied Reeb graphs to simplify Jacobi sets, while Bhatia et al. [BWN*15] introduced a theoretical concept for topological simplification of Jacobi sets, including removing zigzag patterns by canceling critical point pairs. Meduri et al. [MSN24] simplify Jacobi sets for time-varying bivariate 2D fields, focusing on clutter-free visualization of feature traces. Raith et al. [RSH24] offered an approach by adjusting function values and collapsing cells into degenerate forms. Handling these degenerated cells is complex, and collapsing them is very challenging.

An alternative approach is non-topological smoothing-based methods. Tong et al. [TLHD03] split fields into parts, smoothed each separately and combined them again. However, the effectiveness is highly filter-dependent and alters the global function values.

Klötzel et al. [KKZ*22a] take a different approach, in which Jacobi sets are calculated by local bilinear interpolation. This generalization reduces zigzag patterns and improves structural clarity, especially at high resolutions. However, it often creates numerous small loops in noisy data regions, which can lead to confusing visualizations due to overlapping line segments. The follow-up work by Klötzel et al. [KKZ*22b] improved this approach, but the problems still exist, albeit moderately.

The new method simplifies Jacobi sets by removing loops through local adjusting of function values, in contrast to Raith et al.'s collapsing cells. It also reduces their complexity by geometrically identifying zigzag patterns and removing them through local adjusting of function values, in contrast to Bhatia et al.'s topological removal of critical point pairs.

3 BACKGROUND

A *scalar field* is a function $f : \mathbb{D} \to \mathbb{C}$, smoothly mapping from a *domain* $\mathbb{D}$, which is a compact d-manifold, to a *range* $\mathbb{C} \subset \mathbb{R}$ (also called *codomain*). In this paper, we only consider domains that are subsets of $\mathbb{R}^2$. A *critical point* $\mathbf{p}$ of f is a point $\mathbf{p} \in \mathbb{D}$, where the gradient, i.e., the vector of partial derivatives of f, vanishes: $\nabla f(\mathbf{p}) = \mathbf{0}$. A *bivariate scalar field* can be considered as two scalar fields, f, and g, mapped from the same domain.

A *k-simplex* σ is the convex hull of $k+1$ affinely independent points $P_\sigma = \{\mathbf{p}_0, \ldots, \mathbf{p}_k\}$. 2-simplices are triangles, 1-simplices are line segments, and 0-simplices are points. A simplex τ is a *face* of a simplex σ if $P_\tau \subseteq P_\sigma$. A *simplicial complex* is a set of simplices that are closed under the face relation, and any two simplices' intersection is either empty or a common face. A *piecewise linear* function assigns a function value to each 0-simplex and extends this to the other simplices using barycentric interpolation.

3.1 Jacobi sets

Edelsbrunner and Harer [EH04] introduced the concept of the *Jacobi set* $J(f,g)$ for a pair of functions f and g defined on a domain $\mathbb{D}$. The Jacobi set is the collection of points $\mathbf{x} \in \mathbb{D}$ where the gradients $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are linearly dependent:

$$J(f,g) := \{\mathbf{x} \in \mathbb{D} \mid \exists \lambda \in \mathbb{R} : \nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0 \text{ or } \lambda \nabla f(\mathbf{x}) + \nabla g(\mathbf{x}) = 0\} \quad (1)$$

Notably, the critical points of f and g are trivially part of the Jacobi set. Edelsbrunner and Harer [EH04] demonstrated that if $\mathbb{D}$ is a 2-manifold, $J(f,g)$ typically consists of a collection of smooth, pairwise disjoint curves

without self-intersections. The connected components of a Jacobi set are referred to as *Jacobi set curves*.

An equivalent formulation by Edelsbrunner et al. [EHNP04] is that the Jacobi set for two 2D functions comprises points for which the rank of the *Jacobian* matrix (composed from the gradients of f and g) is less than 2. They also introduced a more general criterion: The Jacobi set consists of points for which the local κ-measure is 0. Here, κ is defined as the magnitude of the wedge product of the function gradients.

An alternative method for calculating Jacobi sets without using the absolute value was proposed by Raith et al. [RSH24]. For piecewise linear data, the function restricted to the interior of each triangle σ can be represented as:

$$\mathbf{f}_{|\sigma}(\mathbf{x}) = \mathbf{A}_\sigma \mathbf{x} + \mathbf{b}_\sigma. \tag{2}$$

Using the Taylor expansion of $\mathbf{f}_{|\sigma}$ around $\mathbf{x}_0 = 0$, it follows that $\mathbf{A}_\sigma = \nabla \mathbf{f}_{|\sigma}(\mathbf{x})$. Therefore, the determinant of the Jacobian over a triangle is constant but typically discontinuous across adjacent triangles. A Jacobi set in this context consists of edges between cells with differing signs of the determinant.

We use this formulation because it provides a geometric intuition to simplify the Jacobi set. Specifically, $\mathbf{f}_{|\sigma}$ can be interpreted as a linear transformation that maps the triangle from the domain to the codomain. If the determinant of $\mathbf{A}_\sigma$ (the Jacobian) is negative, the image of the triangle is mirrored, reversing the orientation of its vertices. We define the sign of the Jacobian determinant as the *orientation* of the triangle. According to this orientation, we denote all cells with the same orientation and are enclosed by Jacobi set curves as *Jacobi set components*. Note that the object's boundary is treated like a Jacobi set curve to determine the Jacobi set components.

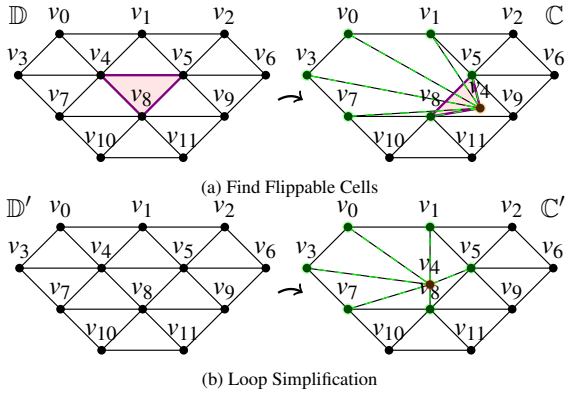
3.2 Loop Subdivision

Loop [Loo87] introduced *loop subdivision* as a central method of computer graphics for refining and smoothing triangular grids $G = (V, C)$. V is a set of vertices, and C is a set of triangular cells. The method consists of 3 parts: **(1)** Each triangle is divided into four smaller triangles. **(2)** New positions are calculated for each vertex $\mathbf{v}_i \in V$ in the grid for existing and new vertices. The position of the new vertices $\mathbf{v}'_i$ is determined by a weighted averaging of the set of all neighbor points of $\mathcal{N}(\mathbf{v}_i)$:

$$\mathbf{v}'_i = \alpha \mathbf{v}_i + \beta \sum_{\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)} \mathbf{v}_j \tag{3}$$

α the weight factor for the original point $\mathbf{v}_i$. β a weighting factor for the neighbor points $\mathcal{N}(\mathbf{v}_i)$, **(3)** The process is repeated until the desired refinement and smoothing are achieved.

We only use part **(2)** of the loop subdivision on the existing vertices in the codomain. As a result, the function values are adjusted and locally glazed.



(a) Find Flippable Cells

(b) Loop Simplification

Figure 1: Schematic overview of the Loop Method - Topology Simplification. (a) shows the Jacobi set component identified for removal in purple in domain $\mathbb{D}$ and the containing cell in red. The corresponding codomain $\mathbb{C}$ shows the mapped cell in red, the moving point in orange, and the edges of the neighboring cells dashed in green. (b) shows the grid after applying the loop simplification. In codomain $\mathbb{C}'$, the neighbors have pulled the cell's point towards them, changing its orientation. This is recognizable in domain $\mathbb{D}'$ and codomain $\mathbb{C}'$ by the absence of a purple Jacobi set.

4 METHOD

The idea is to simplify the Jacobi sets in two parts by removing Jacobi set components in the *Topology Simplification* (Figure 1) and smooth the Jacobi set by removing the zigzag pattern in the *Geometry Simplification* (Figure 2). We were inspired by loop subdivision and called this new approach the *Loop Method* accordingly. The Loop Method does not introduce new points and adjusts the values of the points in the codomain, which corresponds to the function values. Described is this method in structured form in Algorithm 1. We begin with an overview and then describe each component in detail, ensuring a systematic understanding of the process.

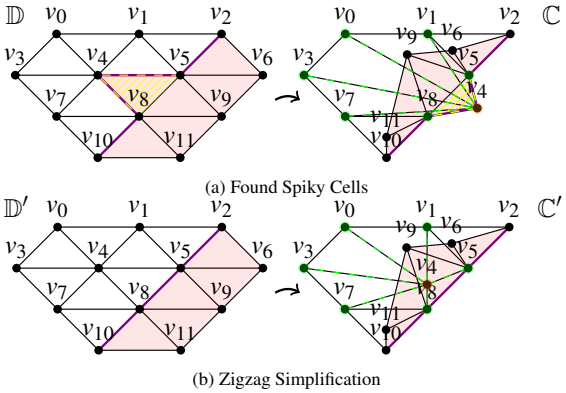
Algorithm 1: Algorithm Overview

Data : 2D Bivariate Scalar Field F, Threshold t
Result : Updated 2D Bivariate Scalar Field F

1 $NG := ComputeNeighborhoodGraph(F)$ } Topology Simp.
2 $FC := FindFlippableCells(NG, t)$ } Topology Simp.
3 $F := LoopSimplification(FC, F)$ } Topology Simp.
4 $NG := UpdateNeighborhoodGraph(F)$ } Geometry Simp.
5 $SC := FindSpikyCells(NG)$ } Geometry Simp.
6 $F := ZigZagSimplification(SC, F)$ } Geometry Simp.

4.1 Overview

The starting point is a given piecewise linear, bivariate 2D scalar field whose Jacobi set is calculated using the method of Section 3.1. For these, the Jacobi set components are determined, and a neighborhood graph is created that contains additional features to identify the



(a) Found Spiky Cells

(b) Zigzag Simplification

Figure 2: Schematic overview of the Loop Method - Geometry Simplification. (a) shows the edge of the Jacobi set (purple) in domain $\mathbb{D}$ and a spiky cell striped in yellow. Cells with a different orientation are red in domain $\mathbb{D}$ and codomain $\mathbb{C}$. In codomain $\mathbb{C}$, the spiky cell is also striped yellow, the moving point is orange, and the neighbors are dashed in green. (b) shows the data set after applying the zigzag simplification. The Jacobi set without the spiky cell appears in purple. Codomain $\mathbb{C}'$ shows how the spiky cell changes its orientation by moving the cell vertex.

components to be simplified (see Figure 1a). In the next step, the orientation of the associated cells of these components is changed in the *Loop Simplification* (cf. Figure 1b), and the neighborhood graph is updated accordingly. The cells of the Jacobi set components are then identified in this graph, which correspond to a spiky cell (see Figure 2a). Note that a *spiky cell* is a cell that lies on two Jacobi set edges. In the *Zigzag Simplification*, the orientation of these cells is adjusted if possible (see Figure 2b). The detailed algorithms for the individual steps are explained in more detail below. The focus is initially on the structure of the neighborhood graph and the specific extensions. This approach effectively simplifies Jacobi sets by reducing their complexity based on various structural properties of Jacobi sets.

4.2 Neighborhood Graph

The neighborhood graph is defined as $NG = (N, E)$, with a set of nodes N where each node $k \in N$ corresponds to a Jacobi set component and a set of edges E where an edge $(k,l) \in E$ exists if two nodes share at least one joint Jacobi set edge. Each node stores additional features, such as the orientation of a Jacobi set component, to analyze the complexity of individual Jacobi set components and visualize these, as in Figure 3. We use the graph to select Jacobi set components to simplify these.

ComputeNeighborhoodGraph(F) calculated this graph by calculating Jacobi sets for a given bivariate scalar field and identifying the Jacobi set components accordingly. To evaluate the complexity of each Jacobi set component, we calculated the features described and shown

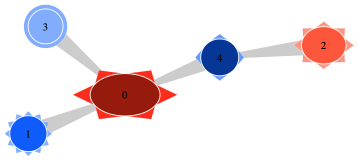


Figure 3: The complexity of the Jacobi set components shows the neighborhood graph, where the features appear as follows: Node size indicates the number of cells contained, color represents the orientation, outer circle saturation visualizes the domain area (A), and inner circle saturation shows the codomain area ($A^{\mathbb{C}}$). The shape of an ellipse represents the ratio of perimeter to area in the domain, and the number of spikes symbolizes the spikiness. The edges show the neighborhood, and the thickness shows the joint Jacobi set edges ratio to the component's total number of Jacobi set edges.

in Figure 3. Here, *spikiness* - measures the proportion of spiky cells. Note *UpdateNeighborhoodGraph(F)* updates the graph about possible changes in the field.

The primary purpose of these features is to identify Jacobi set components for simplification. *FindFlippableCells(NG, t)* identifies components that are to be removed. To do this, the hypervolume $HV = A \cdot A^{\mathbb{C}}$, a combination of the areas in domain and codomain, is calculated. If components are below a threshold t, their cells are summarized in a list and returned. In addition, *FindSpikyCells(NG)* identifies spiky cells of all nodes in the neighborhood graph and returns them to a list.

4.3 Loop Simplification

Algorithm 2: Loop Simplification

Data : 2D Bivariate Scalar Field F, Flippable Cells FC
Result : Updated 2D Bivariate Scalar Field F

```
while FC ≠ ∅ do
    foreach cell c ∈ FC do
        ec := CountEdgesToFlippableCells(FC, c)
        if ec ≠ 3 then
            cv := MoveCellVertexToNeighbors(F, c)
            F := UpdateField(F, cv)
            if CellIsFlipped(c) or
              CellAreaRemainsSame(F, cv) then
                FC := FC \ {c}
                FC := FC \ SameOrientation(F, c)
            FC := FC ∪
              FlippedCellPointNeighbors(F, cv)
```

Based on the cell list created in *FindFlippableCells(NG,t)*, the function values are adjusted as described in Algorithm 2. A greedy algorithm is used to identify the edge cells to shrink the regions to be simplified starting from the edge. We only want to change the grid minimally. Therefore, we determine the vertex of a cell that could change its orientation with

Data set	Length of Jacobi sets			# of Jacobi set components			JSC reduced by		Total Time in ms	
	Original Data	Coll. Algo.	Loop Method	Original Data	Coll. Algo.	Loop Method	Coll. Algo.	Loop Method	Coll. Algo.	Loop Method
Cylinder Flow	92.46	44.06	**38.29**	679	17	**14**	97.50%	**97.94%**	**211.07**	333.07
Red Sea	1 251.41	645.09	**622.46**	7 362	780	**667**	89.41%	**90.94%**	**519.83**	549.10
Hurricane Isabel	915 064	393 343	**380 280**	43 838	2 657	**1 884**	93.94%	**95.70%**	**5 461.20**	7 546.91
Tensile Bar A	1 382.95	615.60	**609.81**	817	**19**	35	**97.67%**	95.72%	**58.74**	70.73
Tensile Bar B	1 476.09	767.15	**674.87**	800	50	**38**	93.75%	**95.25%**	60.81	**55.81**
Tensile Bar C	1 503.17	675.68	**616.66**	883	**18**	22	**97.96%**	97.51%	**54.38**	57.23
Tensile Bar D	963.47	465.58	**458.36**	519	40	**32**	92.29%	**93.83%**	**85.82**	170.09
Tensile Bar E	1 004.65	**535.38**	555.37	490	38	**35**	92.24%	**92.86%**	**52.47**	89.22
Tensile Bar F	791.89	478.77	**472.82**	329	**24**	27	**92.71%**	91.79%	**36.44**	67.63
Tensile Bar G	815.44	514.54	**503.39**	326	36	**28**	88.96%	**91.41%**	**38.17**	68.68
Tensile Bar H	782.77	**447.89**	455.48	322	**28**	48	**91.30%**	85.09%	**48.42**	94.02

Table 1: Comparison of the Jacobi set simplification by the Loop Method and the Collapse Algorithm from Raith et al. [RSH24] with the original data. The measures from Raith et al. [RSH24] are used for all data sets, with the best values in bold. JSC is short for Jacobi set components.

minimal effort. To do this, we test which points of the cell are not surrounded by its neighbor points in the codomain and select the point from these points whose distance to the most distant neighbor is the greatest (cf. Figure 1a). The point position in the codomain is recalculated according to Equation 3 and thus adjusts the function values (cf. Figure 1). This process is visualized in Figure 1. Then, it is checked whether an adjustment could change the cell orientation. If this is the case, this cell and all its neighbors with the same orientation are removed from the FC list. If the orientation of the neighbor cells also changes, they are added to the FC list. If a cell's orientation cannot be changed, it is also removed from the FC list. This algorithm allows for removing Jacobi set components with minimally invasive adjustments to the function values and simplifying the Jacobi set's topology.

4.4 Zigzag Simplification

Algorithm 3: Zigzag Simplification

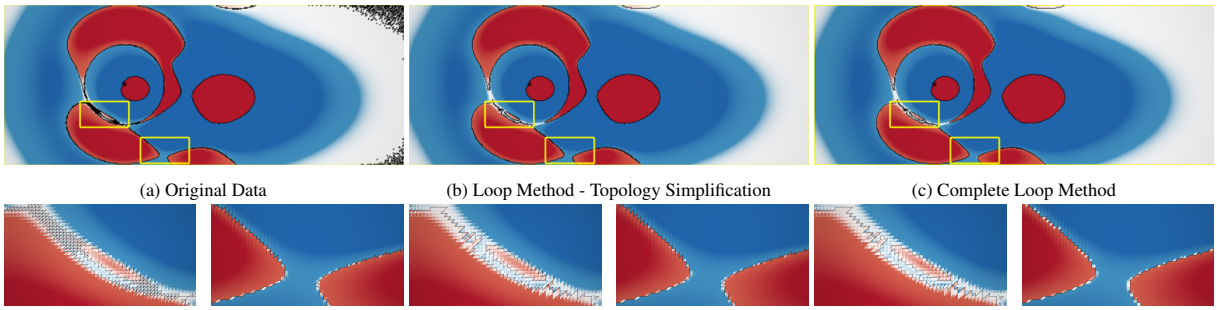
Data : 2D Bivariate Scalar Field F, Spiky Cells SC
Result : Updated 2D Bivariate Scalar Field F

while $SC \neq \emptyset$ **do**
  foreach *cell* $c \in SC$ **do**
    $cv := MoveCellVertexToNeighbors(F, c)$
    if $FlippedCellPointNeighbors(F, cv) = \emptyset$ **then**
      $SC := SC \setminus \{c\}$
    else
      $F := UpdateField(F, cv)$
      if $CellIsFlipped(c)$ ***or*** $CellAreaRemainsSame(F, cv)$ **then**
        $SC := SC \setminus \{c\}$

The orientation is adjusted to remove zigzag patterns based on the list of spiky cells identified in *FindSpikyCells(NG)*. Algorithm 3 first identifies the vertex for each cell from the SC list that could change its orientation with minimal changes in the function values. This vertex is selected similarly to loop simplification by checking the neighbors in the codomain. We use Equation 3 to calculate the new position of the vertex in the codomain. The algorithm then checks whether the neighbor cells would also change their orientation. In this case, the function values in the bivariate field F are not adjusted. The function values are adjusted if the neighbor cells do not change their orientation (cf. Figure 2b). In both cases, the cell is removed from the SC list. The cell is also removed from the list SC if the area remains constant. This algorithm simplifies the geometry of the Jacobi set and, accordingly, its complexity.

5 RESULTS

To evaluate the performance of the new *Loop Method*, we compare it with the Collapse Algorithm from Raith et al. [RSH24] and the original data. We limit ourselves to the Collapse Algorithm because the approach of Bhatia et al. [BWN*15] is only a theoretical concept without implementation, and other algorithms in the literature do not simplify the Jacobi sets by adjusting the function values. We consider the Collapse Algorithm variant A, which achieved the best results in Raith et al. We use data sets representing common application examples to analyze bivariate 2D data. First, we compare the complete Loop Method with the pure Loop Method - topology simplification of Algorithm 1 and the original data and show the effect on the individual function values. We then examine the scalability of the Loop Method with large data sets. Finally, we consider the



(a) Original Data (b) Loop Method - Topology Simplification (c) Complete Loop Method

(d) Cutout 1 (e) Cutout 2 (f) Cutout 1 (g) Cutout 2 (h) Cutout 1 (i) Cutout 2

Figure 4: Comparison of the Jacobi sets of the Loop Method - Topology Simplification with those of the complete Loop Method and the original data for the cylinder flow (178 702 cells) with threshold $t = 0.0001$ from Raith et al. Cutout 1 (coords. (-1.12, -1.06) - (0.13, -0.4)) and cutout 2 (coords. (0.4, -1.97) - (1.65, -1.3)) show more details.

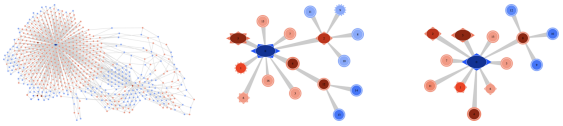
problematic symmetric tensile bar collections of Raith et al. The results of the comparison metrics of Raith et al., including a time analysis, are shown in Table 1. We use the same thresholds from Raith et al. to establish comparability with the Collapse Algorithm.

In all figures, the coloring in the domain represents the orientation of the cells in the codomain, and the saturation visualizes $A^{\mathbb{C}}$. The tests were performed on a MacBook Pro with Apple M1 Max and 64 GB RAM.

5.1 Data sets

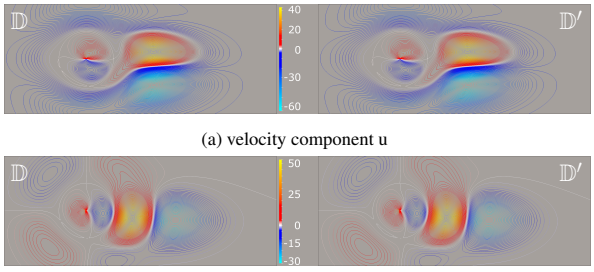
We evaluate the approaches using four well-established data sets from the literature, briefly introduced in this section: the Cylinder Flow, the Red Sea data set, the Hurricane Isabel data set, and the Tensile Bar collection. These data sets come from different application areas where bivariate data analysis is essential.

The Cylinder Flow was constructed by Jung et al. [JTZ93] and represents a synthetic velocity vector field defined on a regular 2D grid. It is publicly available via ETH Zürich [CGL]. The grid was triangulated, and the velocity components u and v were examined in the last time step. The Red Sea data set is known from the SciVis Contest 2020 [ZSYH14, ZKGH19, HLB*18]. We triangulate the grid and consider temperature and Okubo-Weiss criterion for the layer at height 0 with time step 1. We use the Hurricane Isabel data set known from the SciVis Contest 2004 [Uni, KKZ*22a]. For the analysis, we consider the layer at height 50 at time step 30 to analyze the two scalar variables, pressure, and temperature, on the triangulated grid. The tensile bar collection comprises well-studied data sets from structural mechanics [ZS18]. Similar to the literature, we analyze a single layer of the data sets with triangulated grids. The 3D tensor is mapped to a 2D tensor from which the principal invariants I are derived, i.e. the coefficients of the characteristic polynomial: $I_1 = \mathrm{tr}(T) = \lambda_1 + \lambda_2$ and $I_2 = \det(T) = \lambda_1 \cdot \lambda_2$, where $\mathrm{tr}(T)$ is the trace and $\det(T)$ is the determinant of the tensor T for the eigenvalues λ. A description of these tensors can be found in Holzapfel's textbook [Hol00].



(a) Original Data (b) Collapse Algorithm (c) Complete Loop Method

Figure 5: Neighborhood graphs for (a) the original data, (b) the Collapse Algorithm from Raith et al. [RSH24], and (c) the complete Loop Method for the cylinder flow.



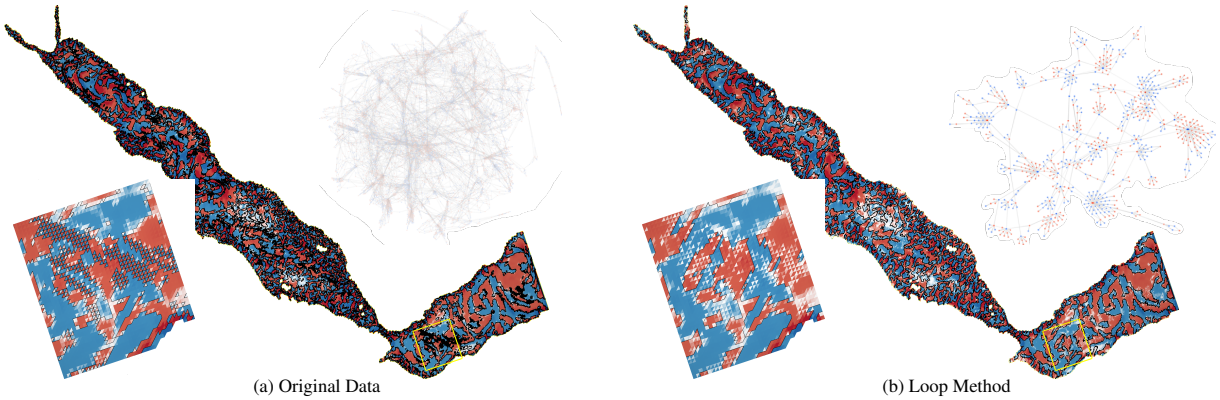
(a) velocity component u

(b) velocity component v

Figure 6: Shows the effects of the Loop Method (right) on the function values of the original data (left) using isolines. The color corresponds to the codomain values.

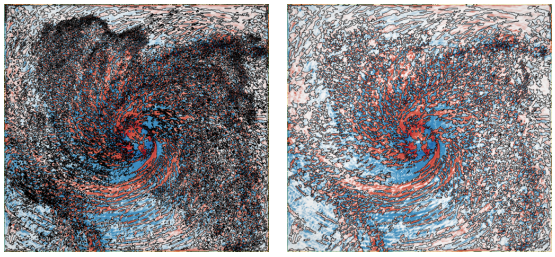
5.2 Comparison Topology and Geometry Simplification

Using the cylinder flow, we examine the effect of the topology simplification and the geometry simplification applied to it, corresponding to the complete Loop Method. We compare these results with the original data in Figure 4 and see in Figures 4d and 4f that Jacobi sets are simplified, and in Figures 4e and 4i, the zigzag patterns are also removed. For the comparison with the Loop Method - topology simplification, we additionally calculate the length of the Jacobi set (43.84) and the number of the Jacobi set components (14). Compared to the results from Table 1, the geometry simplification reduces the length of the Jacobi sets by a further 12.66%. Compared to the Collapse Algorithm, the Loop Method - topology simplification reduces the Jacobi sets more, and the complete Loop Method can even show a significantly reduced length of the Jacobi set.



(a) Original Data (b) Loop Method

Figure 7: Comparison of the Red Sea (498 002 cells) between the Jacobi sets of the original data (a) and the simplified Jacobi sets after applying the Loop Method for threshold $t = 0.0001$ (b) with the corresponding neighborhood graphs. The simplification effects in detail can be seen in cutout (coords. (24.5, 30.7) - (26.0, 32.4).



(a) Original Data (b) Loop Method

Figure 8: Comparison of the Jacobi sets (a) before and (b) after the simplification for Hurricane Isabel (498 002 cells) with the threshold $t = 200$ from Raith et al.

If we look at the neighborhood graphs of the Jacobi set components in Figure 5, we see that the additional properties of the components are easy to recognize. Some properties can already be recognized in the original data (Figure 5a), although this is made more difficult by the mass of the nodes. The comparison between the Collapse Algorithm (Figure 5b) and the Loop Method (Figure 5c) shows that the Jacobi sets can still be reduced. In particular, the spikiness decreases significantly, further simplifying the Jacobi set's complexity.

Finally, we look at the effects of simplifying Jacobi sets on the function values in Figure 6. The separate function values are visualized using isolines before applying the Loop Method (left) and after (right). Only the white isoline with the value 0 shows minimal changes, which shows that the function values are only slightly adjusted.

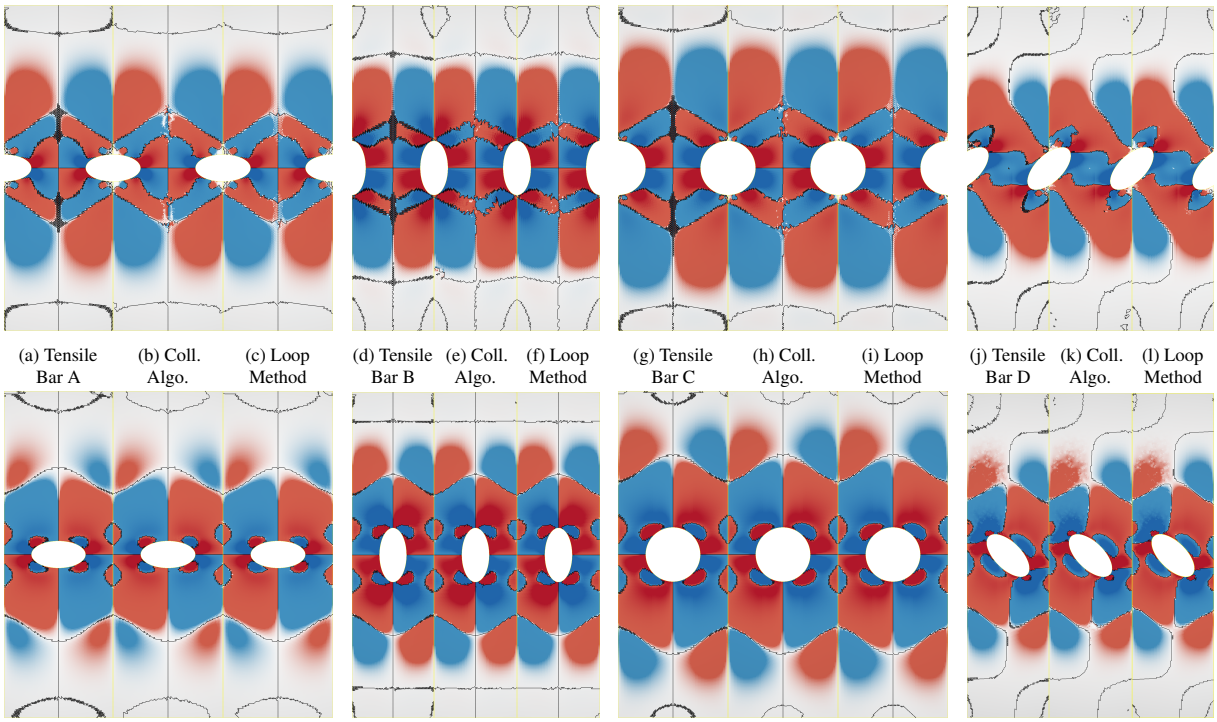
5.3 Evaluation of the Results

Figure 7 compares the Loop Method with the original data for more complex data sets like the Red Sea. In Figure 7a, it can be seen how numerous small Jacobi set components complicate the analysis. In Figure 7b, the Loop Method considerably reduces these. The results in Table 1 confirmed this simplification. The Hurricane Isabel data set achieves a similar result in Figure 8. The original data in Figure 8a show many Jacobi set components. Compared to the simplified Jacobi sets in Figure 8b, it is evident that the Jacobi sets and the small components are reduced. Table 1 also shows this effect, and for the Collapse Algorithm, an additional simplification of the Jacobi set components by 29.09% is achieved and reduces the length by 3.32%.

The tensile bar collection in Figure 9 also shows interesting results. These data sets are symmetric, which poses a particular challenge. Here, the Collapse Algorithm can simplify the Jacobi sets considerably, but the symmetry of the data is impaired (cf. Figure 9h). The loss of symmetry can be seen both in the Jacobi sets themselves and in the color representation of the data set. Let us compare the original data with the simplified Jacobi sets of the Loop Method. The Jacobi sets can be considerably simplified, as in Figures 9d and 9f. It shows that the symmetry is well preserved, as in Figure 9i. The Collapse Algorithm cannot map this symmetry correctly. Table 1 shows that the Collapse Algorithm and the Loop Method significantly reduce the Jacobi set components. The Loop Method reduces the length of the Jacobi sets the most, but the components are usually higher than those of the Collapse Algorithm. As a result, Jacobi set components are less reduced, but they are still 93% with increased quality.

5.4 Time Analysis

In Table 1, the total time for the Loop Method is less than $0.6s$ in 10 data sets. Only the time of Hurricane Isabel is above $7.5s$, although the Red Sea has the same cell number. This indicates that total time is not directly dependent on cell number but on the number of Jacobi set components in the original data. Table 1 also shows this dependency for the Collapse Algorithm. Comparing the total time of both algorithms shows that the Loop Method takes on average 50% more time than the Collapse Algorithm. One reason could be that the Loop Method consists of two parts, and the neighborhood graph has to be created or updated several times.



(a) Tensile Bar A (b) Coll. Algo. (c) Loop Method (d) Tensile Bar B (e) Coll. Algo. (f) Loop Method (g) Tensile Bar C (h) Coll. Algo. (i) Loop Method (j) Tensile Bar D (k) Coll. Algo. (l) Loop Method

(m) Tensile Bar E (n) Coll. Algo. (o) Loop Method (p) Tensile Bar F (q) Coll. Algo. (r) Loop Method (s) Tensile Bar G (t) Coll. Algo. (u) Loop Method (v) Tensile Bar H (w) Coll. Algo. (x) Loop Method

Figure 9: Comparison of the simplified Jacobi sets of the Loop Method with those of the Collapse Algorithm from [RSH24] and the original data for all tensile bars (22 360 - 36 336 cells). Thresholds for simplification from Raith et al.: $t = 0.001$ for tensile bar A-C, D, F, and H, $t = 0.01$ for Tensile Bar D, and $t = 0.0001$ for Tensile Bar G.

5.5 Discussion

Overall, the new Loop Method in Figure 4 features are well preserved, and in Figure 6, the adjustments are locally limited. This simplifies the Jacobi set components by an average of 93% and reduces the length of the Jacobi sets by an average of 50%. Compared to the Collapse Algorithm, the Loop Method can simplify the Jacobi sets according to the components by up to 29.09% and reduce the length of the Jacobi sets by up to 13.09%. Besides, it is shown that the Loop Method simplifies Jacobi sets for data sets with symmetry qualitatively better and preserves the symmetry in contrast to the Collapse Algorithm, but requires 50% more time on average. This is because the focus was, first and foremost, on the quality of the Jacobi sets to use the Loop Method as a preprocessing step. The potential for acceleration is there to achieve a similar time requirement.

6 CONCLUSION AND FUTURE WORK

This paper presented the new Loop Method for simplifying Jacobi sets for bivariate 2D scalar fields on unstructured 2D triangular grids. This method simplifies the Jacobi sets by locally adjusting the function values. First, simplify the topology by removing Jacobi set components before simplifying the geometry by removing zigzag patterns. These regions are identified using a neighborhood graph containing additional Jacobi set features, such as spikiness. We developed an interactive visualization of neighborhood graphs and used it in the visual analysis. The Loop Method reduces the Jacobi set components on average by 93% compared to the original data, and compared to the Collapse Algorithm, the Jacobi set components are simplified by up to 29.09% and reduces the spikiness, which reduces the length of the Jacobi sets by up to 13.09%. Besides, this new method significantly improves the quality of Jacobi sets in symmetric tensile bars.

Due to the higher quality of the Jacobi sets, the next step is a detailed evaluation with domain experts to check the robustness. A next step could also be to speed up the algorithm by splitting the Jacobi set components into spatially separated areas for parallel processing. Automatic threshold generation, such as an adaptive selection strategy based on the statistical properties of the data set, to identify the Jacobi set components to be simplified is also an important next step. Another step would be generalizing 3D data sets, especially for trivariate 3D scalar fields. Since the Jacobi sets also divide the space into different components in the 3D case. However, this generalization is not trivial and requires further investigation to evaluate its impact on the overall model. Finally, it could be a worthwhile goal to investigate and visualize the uncertainty of a simplified Jacobi set, especially for the codomain.

7 ACKNOWLEDGMENTS

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - SCHE 663/17-1. The authors thank Markus Stommel of the Leibniz Institute of Polymer Research Dresden, Germany for providing the tensile bar data sets and Bill Kuo, Wei Wang, Cindy Bruyere, Tim Scheitlin, and Don Middleton of the U.S. National Center for Atmospheric Research (NCAR), and the U.S. National Science Foundation (NSF) for providing the Weather Research and Forecasting (WRF) Model simulation data of Hurricane Isabel. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the program Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: SCADS24B.

8 REFERENCES

[AAM17] ARTAMONOVA I. V., ALEKSEEV V. V., MAKARENKO N. G.: Gradient measure and Jacobi sets for estimation of interrelationship between geophysical multifields. *Journal of Physics: Conference Series* (2017). doi:10.1088/1742-6596/798/1/012040.

[BBD*07] BREMER P., BRINGA E., DUCHAINEAU M., GYULASSY A., LANEY D., MASCARENHAS A., PASCUCCI V.: Topological feature extraction and tracking. *Journal of Physics: Conference Series* (2007). doi:10.1088/1742-6596/78/1/012007.

[BHEP04] BREMER P.-T., HAMANN B., EDELSBRUNNER H., PASCUCCI V.: A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics* (2004). doi:10.1109/TVCG.2004.3.

[BWN*15] BHATIA H., WANG B., NORGARD G., PASCUCCI V., BREMER P.-T.: Local, smooth, and consistent Jacobi set simplification. *Computational Geometry* (2015). doi:10.1016/j.comgeo.2014.10.009.

[CCD*16] CHATTOPADHYAY A., CARR H., DUKE D., GENG Z., SAEKI O.: Multivariate topology simplification. *Computational Geometry* (2016). doi:10.1016/j.comgeo.2016.05.006.

[CCDG14] CHATTOPADHYAY A., CARR H. A., DUKE D. J., GENG Z.: Extracting Jacobi Structures in Reeb Spaces. *EuroVis (Short Papers)* (2014). doi:10.2312/eurovisshort.20141156.

[CD13] CARR H., DUKE D.: Joint contour nets. *IEEE Transactions on Visualization and Computer Graphics* (2013). doi:10.1109/PacificVis.2013.6596141.

[CGL] CGL, ETH ZÜRICH: CGL @ ETHZ - Data. https://cgl.ethz.ch/research/visualization/data.php.

[CSA03] CARR H., SNOEYINK J., AXEN U.: Computing contour trees in all dimensions. *Computational Geometry* (2003). doi:10.1016/S0925-7721(02)00093-7.

[CSEH05] COHEN-STEINER D., EDELSBRUNNER H., HARER J.: Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry* (2005). doi:10.1007/s00454-006-1276-5.

[CSVDP04] CARR H., SNOEYINK J., VAN DE PANNE M.: Simplifying flexible isosurfaces using local geometric measures. *IEEE Visualization 2004* (2004). doi:10.1109/VISUAL.2004.96.

[CSVDP10] CARR H., SNOEYINK J., VAN DE PANNE M.: Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. *Computational Geometry* (2010). doi:10.1016/j.comgeo.2006.05.009.

[EH04] EDELSBRUNNER H., HARER J.: Jacobi Sets of Multiple Morse Functions. *Foundations of Computational Mathematics, Minneapolis 2002* (2004). doi:10.1017/CBO9781139106962.003.

[EHNP04] EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Local and global comparison of continuous functions. *IEEE Visualization 2004* (2004). doi:10.1109/VISUAL.2004.68.

[EHZ03] EDELSBRUNNER, HARER, ZOMORODIAN: Hierarchical morse-smale complexes for piecewise linear 2-manifolds. *Discrete & Computational Geometry* (2003). doi:10.1007/s00454-003-2926-5.

[ELZ02] EDELSBRUNNER, LETSCHER, ZOMORODIAN: Topological persistence and simplification. *Discrete & computational geometry* (2002). doi:10.1109/SFCS.2000.892133.

[HLB*18] HOTEIT I., LUO X., BOCQUET M., KOHL A., AIT-EL-FQUIH B.: Data assimilation in oceanography: current status and new directions. *New frontiers in operational oceanography* (2018). doi:10.1029/2018GL081387.

[HLH*16] HEINE C., LEITTE H., HLAWITSCHKA M., IURICICH F., DE FLORIANI L., SCHEUERMANN G., HAGEN H., GARTH C.: A survey of topology-based methods in visualization. *Computer Graphics Forum* (2016). doi:10.1111/cgf.12933.

[Hol00] HOLZAPFEL A. G.: *Nonlinear solid mechanics II*. John Wiley & Sons, Inc., 2000.

[HTWL19] HE X., TAO Y., WANG Q., LIN H.: Multivariate spatial data visualization: a survey.

Journal of visualization (2019). doi:10.1007/s12650-019-00584-3.

[ISLDF16] IURICICH F., SCARAMUCCIA S., LANDI C., DE FLORIANI L.: A discrete Morse-based approach to multivariate data analysis. In *SIGGRAPH ASIA 2016 Symposium on Visualization* (2016). doi:10.1145/3002151.3002166.

[JTZ93] JUNG C., TÉL T., ZIEMNIAK E.: Application of scattering chaos to particle transport in a hydrodynamical flow. *Chaos: An Interdisciplinary Journal of Nonlinear Science* (1993). doi:10.1063/1.165960.

[KKZ*22a] KLÖTZL D., KRAKE T., ZHOU Y., HOTZ I., WANG B., WEISKOPF D.: Local bilinear computation of Jacobi sets. *The Visual Computer* (2022). doi:10.1007/s00371-022-02557-4.

[KKZ*22b] KLÖTZL D., KRAKE T., ZHOU Y., STOBER J., SCHULTE K., HOTZ I., WANG B., WEISKOPF D.: Reduced connectivity for local bilinear Jacobi sets. *2022 Topological Data Analysis and Visualization (TopoInVis)* (2022). doi:10.1109/TopoInVis57755.2022.00011.

[Loo87] LOOP C.: *Smooth subdivision surfaces based on triangles*. PhD thesis, University of Utah, 1987.

[LSW09] LUO C., SAFA I., WANG Y.: Approximating gradients for meshes and point clouds via diffusion metric. *Computer Graphics Forum* (2009). doi:10.1111/j.1467-8659.2009.01526.x.

[MOQ*20] MAKELA K., OPHELDERS T., QUIGLEY M., MUNCH E., CHITWOOD D., DOWTIN A.: Automatic tree ring detection using Jacobi sets. *arXiv* (2020). doi:10.48550/arXiv.2010.08691.

[MSN24] MEDURI D., SHARMA M., NATARAJAN V.: Jacobi set simplification for tracking topological features in time-varying scalar fields. *The Visual Computer* (2024). doi:10.1007/s00371-024-03577-y.

[NB13] NORGARD G., BREMER P.-T.: Ridge–Valley graphs: Combinatorial ridge detection using Jacobi sets. *Computer Aided Geometric Design* (2013). doi:10.1016/j.cagd.2012.03.015.

[Ree46] REEB G.: Sur les points singuliers d'une forme de pfaff completement integrable ou d'une fonction numerique [on the singular points of a completely integrable pfaff form or of a numerical function]. *Comptes Rendus Acad. Sciences Paris* (1946).

[RNSH25] RAITH F., NSONGA B., SCHEUERMANN G., HEINE C.: Fast Fiber Surface and Fiber Line Extraction for Bivariate Scalar Fields using Dual Bounding Volume Hierarchy Traversal. In *2025 IEEE Pacific Visualization Symposium (PacificVis)* (2025), IEEE.

[RSH24] RAITH F., SCHEUERMANN G., HEINE C.: Topological Simplifcation of Jacobi Sets for Piecewise-Linear Bivariate 2D Scalar Fields with Adjustment of the Underlying Data. *Topological Data Analysis and Visualization (TopoInVis)* (2024). doi:10.1109/TopoInVis64104.2024.00007.

[SJ15] STRODTHOFF B., JÜTTLER B.: Layered Reeb graphs for three-dimensional manifolds in boundary representation. *Computers & Graphics* (2015). doi:10.1016/j.cag.2014.09.026.

[SMC*07] SINGH G., MÉMOLI F., CARLSSON G. E., ET AL.: Topological methods for the analysis of high dimensional data sets and 3D object recognition. *PBG@ Eurographics* (2007). doi:10.2312/SPBG/SPBG07/091-100.

[SN09] SUTHAMBHARA N., NATARAJAN V.: Simplification of Jacobi sets. *Topological Methods in Data Analysis and Visualization* (2009). doi:10.1007/978-3-642-15014-2_8.

[SN22] SHARMA M., NATARAJAN V.: Jacobi Set Driven Search for Flexible Fiber Surface Extraction. In *2022 Topological Data Analysis and Visualization (TopoInVis)* (2022). doi:10.1109/TopoInVis57755.2022.00012.

[TC16] TIERNY J., CARR H.: Jacobi fiber surfaces for bivariate Reeb space computation. *IEEE Transactions on Visualization and Computer Graphics* (2016). doi:10.1109/TVCG.2016.2599017.

[TLHD03] TONG Y., LOMBEYDA S., HIRANI A. N., DESBRUN M.: Discrete multiscale vector field decomposition. *ACM transactions on graphics (TOG)* (2003). doi:10.1145/882262.882290.

[Uni] UNIVERSITY CORPORATION FOR ATMOSPHERIC RESEARCH: Hurricane Isabel WRF Model Data Visualization. https://www.earthsystemgrid.org/dataset/isabeldata.html.

[VKvOB*97] VAN KREVELD M., VAN OOSTRUM R., BAJAJ C., PASCUCCI V., SCHIKORE D.: Contour trees and small seed sets for isosurface traversal. *Proceedings of the thirteenth annual symposium on Computational geometry* (1997). doi:10.1145/262839.269238.

[ZKGH19] ZHAN P., KROKOS G., GUO D., HOTEIT I.: Three-dimensional signature of the Red Sea eddies and eddy-induced transport. *Geophysical Research Letters* (2019). doi:10.1029/2018GL081387.

[ZS18] ZOBEL V., SCHEUERMANN G.: Extremal curves and surfaces in symmetric tensor fields. *The Visual Computer* (2018). doi:10.1007/s00371-017-1450-1.

[ZSYH14] ZHAN P., SUBRAMANIAN A. C., YAO F., HOTEIT I.: Eddies in the Red Sea: A statistical and dynamical study. *Journal of Geophysical Research: Oceans* (2014). doi:10.1002/2013JC009563.