



# Local bilinear computation of Jacobi sets

Daniel Klötzl<sup>1</sup> · Tim Krake<sup>1</sup> · Youjia Zhou<sup>2</sup> · Ingrid Hotz<sup>3</sup> · Bei Wang<sup>2</sup> · Daniel Weiskopf<sup>1</sup>

Accepted: 26 May 2022 / Published online: 30 June 2022  
© The Author(s) 2022

## Abstract

We propose a novel method for the computation of Jacobi sets in 2D domains. The Jacobi set is a topological descriptor based on Morse theory that captures gradient alignments among multiple scalar fields, which is useful for multi-field visualization. Previous Jacobi set computations use piecewise linear approximations on triangulations that result in discretization artifacts like zig-zag patterns. In this paper, we utilize a local bilinear method to obtain a more precise approximation of Jacobi sets by preserving the topology and improving the geometry. Consequently, zig-zag patterns on edges are avoided, resulting in a smoother Jacobi set representation. Our experiments show a better convergence with increasing resolution compared to the piecewise linear method. We utilize this advantage with an efficient local subdivision scheme. Finally, our approach is evaluated qualitatively and quantitatively in comparison with previous methods for different mesh resolutions and across a number of synthetic and real-world examples.

**Keywords** Jacobi set · Topological data analysis · Multi-fields · Visualization techniques

## 1 Introduction

There exist numerous methods for the analysis and visualization of multi-fields. One particular approach is the comparison of scalar fields via topological descriptors. The understanding, interpretation, and computation of topological descriptors remain a challenging task [30]. One of these tools is the Jacobi set. The Jacobi set is based on Morse theory and captures gradient alignments among multiple scalar fields. Currently, Jacobi set computation is based on piece-

wise linear approximations on triangulations [8]. Piecewise linear interpolation typically produces non-smooth and inaccurate results. In the case of edge-based Jacobi set extraction, it leads to zig-zag patterns and discretization artifacts as illustrated in the top row of Fig. 1. These zig-zag patterns are even preserved for higher resolutions since the traditional method only extracts the edges of the underlying triangulation and does not utilize the cell interior. To address these problems, previous works explore the simplifications of Jacobi sets by either simplifying the scalar functions (indirect methods) or the extracted Jacobi sets (direct methods). While these approaches may improve the Jacobi set representation, they all rely on extracting edges that form the Jacobi set using a piecewise linear approach.

In this paper, our main goal is to extend the piecewise linear computation to a bilinear one to arrive at a more accurate geometry of the extracted Jacobi set. To achieve this, we introduce an equivalent formulation of the piecewise linear method and then extend it to our higher-order approach. Our contributions can be summarized as follows:

- We provide an equivalent formulation for the computation of Jacobi sets and embed the piecewise linear method into this framework.
- Due to the reformulation, we introduce a novel Jacobi set extraction that depends on bilinear interpolation,

✉ Daniel Klötzl  
daniel.kloetzl@visus.uni-stuttgart.de

Tim Krake  
tim.krake@visus.uni-stuttgart.de

Youjia Zhou  
zhou325@sci.utah.edu

Ingrid Hotz  
ingrid.hotz@liu.se

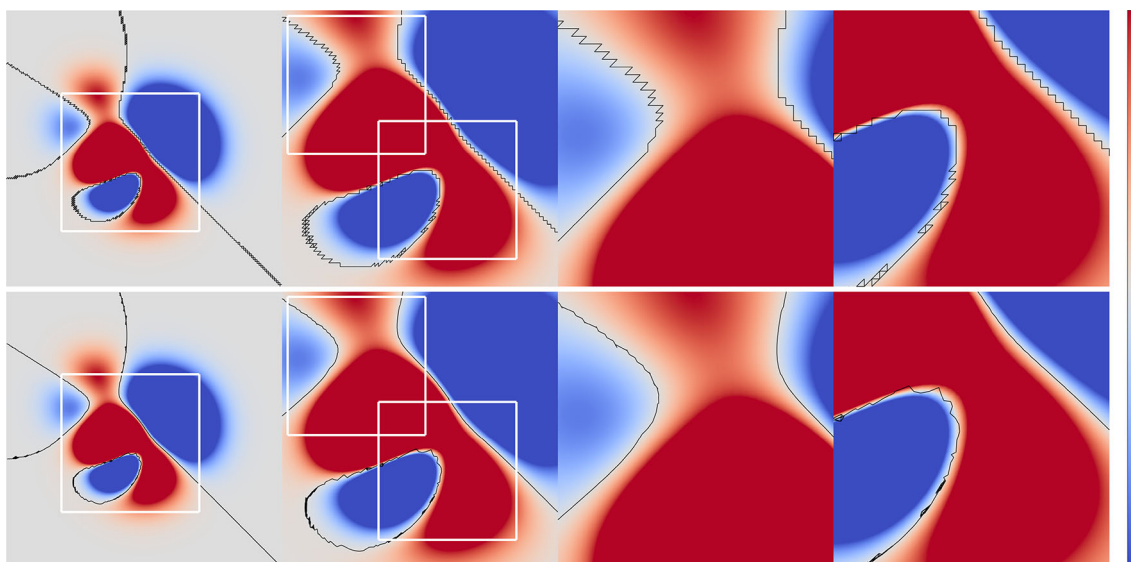
Bei Wang  
beiwang@sci.utah.edu

Daniel Weiskopf  
daniel.weiskopf@visus.uni-stuttgart.de

<sup>1</sup> University of Stuttgart (VISUS), Stuttgart, Germany

<sup>2</sup> University of Utah (SCI), Salt Lake City, United States

<sup>3</sup> Linköping University (ITN), Linköping, Sweden



**Fig. 1** Comparison of the traditional piecewise linear method (top) and our local bilinear method (bottom) for a fixed resolution and different zoom levels on a rectangular grid. The Jacobi set is marked in black and the color-coding corresponds to the ground-truth gradient alignment field (of two analytic functions). The color bar is ordered from positive

(top) to negative (bottom), where white indicates the zero. It can be observed that the piecewise linear method results in zig-zag patterns and discretization artifacts, whereas our method leads to a smoother representation of the Jacobi set

leading to a smoother, non-edge-based Jacobi set representation.

- We present an efficient local subdivision method that further improves the extracted Jacobi set representation.

We compare our method to the piecewise linear method for three different datasets: an analytic dataset, a dataset with a von Kármán vortex street, and the Hurricane Isabel dataset.

The paper is structured as follows: After the introduction, we review related work in Sect. 2. To lay the foundations, Sect. 3 provides the theoretical background of Jacobi sets as well as the piecewise linear method to calculate them. In Sect. 4, we introduce the relevant concepts for our local bilinear approach. Subsequently, we derive our approach in three steps and present an efficient local subdivision scheme. Next, Sect. 5 provides a thorough evaluation of our method. Finally, the conclusion with future research directions is presented in Sect. 6.

## 2 Related work

The context of this paper is the analysis and visualization of multi-fields via the comparison of scalar fields. In the following, we will summarize previous works that address this topic by extracting joint topological structures [17,30], some of which are closely related to the notion of Jacobi sets

Simultaneous analysis of isosurfaces of several scalar fields using a variation density function has been proposed

by Nagaraj et al. [23]. Tools studying multivariate data topologically go one step further. Such methods are, in particular, the piecewise linear computation of Jacobi set [8], the Reeb space [11] and its variants [6], Pareto sets [19], and multivariate persistent homology [5]. The Reeb space compresses the components of the level sets of a multivariate mapping and obtains a summary representation of their relationships. Instead of exploring Jacobi sets in the domain, Chattopadhyay et al. [7] studied multi-field topology by considering projections of the Jacobi set into the Reeb space. Thus, these two concepts are shown to be related as the image of the Jacobi sets under the mapping corresponds to certain singularities in the Reeb space [6].

Another related concept is that of Pareto sets [19]. Here, a set of scalar fields is analyzed with respect to consensus or disagreement among ascending and descending manifolds, critical points, and their connectivity. A follow-up paper [18] investigates the relation to Jacobi sets and derives subset and equivalence relations between Jacobi sets and Pareto sets. In addition, the mapper construction [28] and the joint contour nets [6] can be considered as discrete variants of the Reeb space.

All these previous works either introduce a different topological descriptor or establish connections to the concept of Jacobi sets. In contrast, our approach aims to improve the geometrical representation of Jacobi sets with a higher-order interpolation scheme.

As part of topological data analysis tools, the Jacobi set can be used to compare multiple scalar fields [10], track critical

points [4,8], define silhouettes of objects [12], and extract ridges in image analysis [26]. It has also been employed in various scientific applications [1,2,8,20,22]. However, in practice, the Jacobi set contains degeneracies and discretization artifacts that can be reduced by simplification of the Jacobi set. Bremer et al. [4] perform persistent simplification of the scalar field to remove small loops of the Jacobi set that lie entirely within successive time steps. Luo et al. [21] compute the Jacobi set by approximating the gradients of a pair of scalar fields. Nagaraj and Natarajan [24] simplify the Jacobi set by explicitly reducing the number of components. Bhatia et al. [3] identify and remove unimportant portions of the Jacobi set by determining how the pair of scalar fields must be modified locally. Our proposed work utilizes bilinear interpolation to obtain a smoother and more accurate geometric representation of the Jacobi set (compared to the piecewise linear approach), while preserving its topology. Therefore, our work can be seen as a completely different approach. To the best of our knowledge, the computation of Jacobi sets under higher-order interpolation schemes has not been explored and understood, and our investigation using bilinear interpolation is the first step in this direction.

Another related concept is the general search for parallel vectors in two fields, e.g., for the computation of ridgelines or the detection of vortex cores in the flow. An introduction to the parallel vectors operator has been given by Peikert and Roth [27], formulated as a search for roots of the cross-product of the two vector fields. As a computational solution, they suggest an extraction via zero-level isolines for two-dimensional vector fields, e.g., by using marching squares. Further improvements for higher-dimensional fields were presented by Theisel et al. [29] and van Gelder et al. [14], who express the problem as a streamline integration problem and make use of tracing algorithms.

While our approach also works with cross-products and an extraction via zero-level isolines, we focus on an edge-based identification of gradient alignments. To be more precise, our Jacobi set extraction starts with scalar fields, computes edge-based gradients in a local bilinear fashion, and identifies the gradient alignments. This means that we do not extract a global gradient field, which is necessary for the parallel vectors operator. Furthermore, due to the preservation of the topology of the piecewise linear method [8], our approach guarantees several topological properties; in particular, it satisfies the Even Degree Lemma [8], which is at the core of our connection method.

### 3 Background

In this section, we review the underlying theory of Jacobi sets and outline the piecewise linear computation of Jacobi sets by Edelsbrunner and Harer [8].

### 3.1 Jacobi sets and Morse theory

The Jacobi set is defined for a pair of Morse functions on a smooth  $d$ -manifold  $\mathbb{M}$ . For the purpose of this paper, we restrict our attention to a subset  $\mathbb{M} \subset \mathbb{R}^2$  of the Euclidean space  $\mathbb{R}^2$ . Given a smooth function  $f : \mathbb{M} \rightarrow \mathbb{R}$ , the gradient  $\nabla f(\mathbf{x})$  at the point  $\mathbf{x} \in \mathbb{M}$  is well-defined by its partial derivatives. The point  $\mathbf{x}$  is a *critical point* if the gradient vanishes; otherwise it is a *regular point*. A critical point  $\mathbf{x}$  is *non-degenerate* if the Hessian (i.e., the matrix of second-order derivatives) is invertible. The function  $f$  is a *Morse function* if every critical point is non-degenerate and the critical points have distinct function values [9, page 128].

Given two Morse functions  $f, g : \mathbb{M} \rightarrow \mathbb{R}$ , the Jacobi set  $\mathbb{J}(f, g)$  is defined as the set of points where the gradients of both functions are linearly dependent [8], that is

$$\begin{aligned} \mathbb{J}(f, g) := & \{\mathbf{x} \in \mathbb{M} \mid \nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0 \\ & \text{or } \lambda \nabla f(\mathbf{x}) + \nabla g(\mathbf{x}) = 0, \lambda \in \mathbb{R}\} \\ = & \{\mathbf{x} \in \mathbb{M} \mid \mathbf{x} \text{ is a critical point of } f(\mathbf{x}) + \lambda g(\mathbf{x}) \\ & \text{or of } \lambda f(\mathbf{x}) + g(\mathbf{x}), \lambda \in \mathbb{R}\}. \end{aligned}$$

In other words, points in the Jacobi set are referred to as the *simultaneous critical points* of  $f$  and  $g$  [8]. For any value  $t \in \mathbb{R}$ , we consider the restriction of  $f$  to the level sets  $g^{-1}(t)$  of  $g$ ,  $f_t : g^{-1}(t) \rightarrow \mathbb{R}$ .  $\mathbb{J}(f, g)$  can be alternatively formulated as the closure of the set of critical points of  $f_t$ , that is

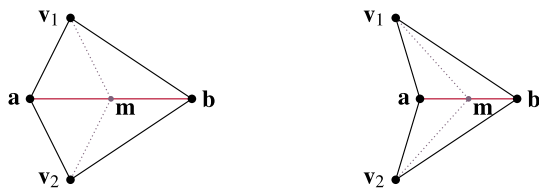
$$\mathbb{J}(f, g) = \text{cl}\{\mathbf{x} \in \mathbb{M} \mid \mathbf{x} \text{ critical point of } f_t, \text{ for some } t \in \mathbb{R}\}.$$

The Jacobi set is symmetric, i.e.,  $\mathbb{J}(f, g) = \mathbb{J}(g, f)$ . Furthermore, it is a smoothly embedded 1-manifold in  $\mathbb{M}$  [8].

### 3.2 Piecewise linear computation of Jacobi sets

We now describe the traditional piecewise linear computation of Jacobi sets following the work of Edelsbrunner and Harer [8]. Let  $K$  be a triangulation of  $\mathbb{M}$  and let  $\phi, \psi : \text{Vert}(K) \rightarrow \mathbb{R}$  be functions defined on the vertices of  $K$ ,  $\text{Vert}(K)$ . They consider the functions  $f$  and  $g$  to be the piecewise linear extensions of  $\phi$  and  $\psi$ , respectively,  $f, g : |K| \rightarrow \mathbb{R}$ . Then, they model both of these functions as the limits of a series of smooth functions,  $\lim_{n \rightarrow \infty} f_n = f$  and  $\lim_{n \rightarrow \infty} g_n = g$ . The Jacobi set is constructed as the limit,  $\mathbb{J}(f, g) := \lim_{n \rightarrow \infty} \mathbb{J}(f_n, g_n)$ .

The computation of  $\mathbb{J}(f, g)$  is performed by tracing the critical points of the 1-parameter family of functions  $h_\lambda = f + \lambda g$  along the edges of  $K$ . To be more precise, the algorithm determines for each edge  $\mathbf{ab}$  the value  $\lambda$ , such that  $h_\lambda(\mathbf{a}) = h_\lambda(\mathbf{b})$ . This leads to  $\lambda = \lambda_{\mathbf{ab}} = (f_{\mathbf{b}} - f_{\mathbf{a}}) / (g_{\mathbf{a}} - g_{\mathbf{b}})$ .



(a) Convex quadrilateral cell      (b) Concave quadrilateral cell

**Fig. 2** Critical edge test for 2-dimensional complexes. The reverse transformation line from a quadratic  $\{av_1bv_2\}$ -cell between  $v_1$  and  $v_2$  is marked by the dotted lines

To determine whether such an edge  $\mathbf{ab}$  is critical (i.e., belonging to the Jacobi set), the algorithm examines the *link* of  $\mathbf{ab}$ . Recall that the *star* of a simplex  $\tau \in K$  is the collection of simplices that contains  $\tau$ ; and the *link* includes simplices in the closure of the star, i.e.,  $\text{St}(\tau) = \{\sigma \in K \mid \tau \leq \sigma\}$  and  $\text{Lk}(\tau) = \{\nu \in \text{Cl}(\text{St}(\tau)) \mid \nu \cap \tau = \emptyset\}$ . The lower star of a vertex  $\mathbf{a}$  consists of all simplices in the star that have  $\mathbf{a}$  as their highest vertex, while the lower link is the portion of the link that bounds the lower star. The lower link of  $\mathbf{ab}$  is the subcomplex of vertices  $v_i$  whose  $h_{\lambda_{\mathbf{ab}}}$  function values are smaller than the ones from  $\mathbf{a}$  and  $\mathbf{b}$ .

Let  $l$  be the restriction of  $h_{\lambda_{\mathbf{ab}}}$  to the link of  $\mathbf{ab}$ . The computation relies on the following lemma.

**Critical Edge Lemma** An edge  $\mathbf{ab}$  belongs to  $\mathbb{J}$  if and only if vertex  $\mathbf{a}$  is a critical point of  $l$ . Moreover, the multiplicity of  $\mathbf{ab}$  in  $\mathbb{J}$  is the multiplicity of  $\mathbf{a}$  as critical point of  $l$ .

An edge  $\mathbf{ab}$  with the vertices  $v_1$  and  $v_2$  in its link  $\text{Lk}(\mathbf{ab})$  (see Fig. 2) is a Jacobi edge or critical if and only if [25]

$$h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) > h_{\lambda_{\mathbf{ab}}}(v_1) \quad \text{and} \quad h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) > h_{\lambda_{\mathbf{ab}}}(v_2), \quad (1)$$

$$\text{or} \quad h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) < h_{\lambda_{\mathbf{ab}}}(v_1) \quad \text{and} \quad h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) < h_{\lambda_{\mathbf{ab}}}(v_2). \quad (2)$$

We differentiate between the criticality of the edges: if Eq. 1 holds, the edge is maximal; if Eq. 2 holds, the edge is minimal. These inequalities may lead to degenerate cases, which are resolved by using the Simulation of Simplicity [13]. We refer to Natarajan [25] for details.

Recall that the degree of a vertex is the number of edges that contain the vertex. After computing a set of critical edges, the next step of the algorithm is to construct the 1-manifold  $\mathbb{J}$ . Based on the Even Degree Lemma by Edelsbrunner and Harer [8], stated below, the algorithm concatenates the edges in pairs, thus forming the 1-manifold  $\mathbb{J}$ . Our proposed approach builds upon the above edge-based Jacobi set extraction under bilinear interpolation.

**Even Degree Lemma** The degree of a vertex in  $\mathbb{J}$  is even.

## 4 Our approach

In this section, we first outline the motivation for our approach by establishing a new perspective on the piecewise linear method. This lays the foundation for our proposed local bilinear method that preserves the topology of Jacobi set from [8] and extracts the Jacobi set locally with a bilinear interpolation scheme. Thus, the restriction to an edge-based Jacobi set representation is avoided and the geometry is enhanced. Finally, an appropriate local subdivision method for an efficient Jacobi set extraction is presented.

### 4.1 Motivation and derivation

The piecewise linear computation of Jacobi sets by Edelsbrunner and Harer [8] has several limitations. One of these is that only the edges of the underlying triangulation can be identified as components of the Jacobi set. This leads to zig-zag patterns, poor convergence for increasing mesh resolution, and an imprecise Jacobi set representation. To avoid these issues, the natural way is to use a higher-order interpolation scheme, which is the core of our approach.

To apply this, we propose to use an equivalent formulation of Jacobi sets, that is

$$\mathbb{J}(f, g) = \{\mathbf{x} \in \mathbb{M} \mid \nabla f(\mathbf{x}) \times \nabla g(\mathbf{x}) = 0\}. \quad (3)$$

In our setting, where  $\mathbb{M}$  is a smooth 2-manifold embedded in  $\mathbb{R}^3$ , the cross-product of  $\nabla f(\mathbf{x})$  and  $\nabla g(\mathbf{x})$  can be expressed with the help of the unit vectors  $\mathbf{e}_i$  and the determinant

$$\nabla f(\mathbf{x}) \times \nabla g(\mathbf{x}) = \det \begin{pmatrix} \mathbf{e}_1 & \partial_1 f(\mathbf{x}) & \partial_1 g(\mathbf{x}) \\ \mathbf{e}_2 & \partial_2 f(\mathbf{x}) & \partial_2 g(\mathbf{x}) \\ \mathbf{e}_3 & \partial_3 f(\mathbf{x}) & \partial_3 g(\mathbf{x}) \end{pmatrix}.$$

Assuming that  $\mathbb{M}$  lies in the  $xy$ -plane, the cross-product is

$$\nabla f(\mathbf{x}) \times \nabla g(\mathbf{x}) = (\partial_x f(\mathbf{x}) \partial_y g(\mathbf{x}) - \partial_y f(\mathbf{x}) \partial_x g(\mathbf{x})) \cdot \mathbf{e}_3.$$

We define  $\kappa_{\mathbf{x}}$ , called *gradient alignment value*, as

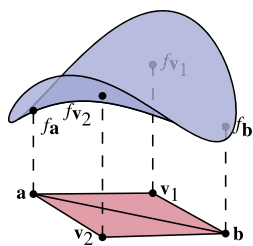
$$\kappa_{\mathbf{x}}(f, g) := \partial_x f(\mathbf{x}) \partial_y g(\mathbf{x}) - \partial_y f(\mathbf{x}) \partial_x g(\mathbf{x}). \quad (4)$$

The gradient alignment value  $\kappa_{\mathbf{x}}$  characterizes the linear independence of the gradients at the point  $\mathbf{x} \in \mathbb{M}$ . This leads to the scalar field  $\kappa_{\mathbf{x}} : \mathbb{M} \rightarrow \mathbb{R}$  as a function of  $\mathbf{x}$ , which we call the *gradient alignment field*. According to the Jacobi set definition, the point  $\mathbf{x}$  belongs to the Jacobi set  $\mathbb{J}(f, g)$  if and only if  $\kappa_{\mathbf{x}}(f, g) = 0$ .

Our next step is to examine how the (smooth) gradient alignment field  $\kappa_{\mathbf{x}}(f, g)$  behaves for both linearly and bilinearly interpolated functions  $f$  and  $g$ , the corresponding gradient alignment fields are denoted  $\kappa_{\mathbf{x}}^{\text{li}}$  and  $\kappa_{\mathbf{x}}^{\text{bi}}$ , respectively.



**Fig. 3** Illustration of the scalar field  $f$  over two connected triangles  $\mathbf{abv}_1$  and  $\mathbf{av}_2\mathbf{b}$



We assume  $f$  to be linear in each of the triangles  $\{\mathbf{abv}_1\}$  and  $\{\mathbf{av}_2\mathbf{b}\}$  (for the notation, see Fig. 3). For the triangle  $\{\mathbf{abv}_1\}$ , we use the formula  $f_{\mathbf{abv}_1}(x, y) := a_{00}^f + a_{10}^f x + a_{01}^f y$ , which leads to the following system of linear equations:

$$\begin{pmatrix} 1 & x_a & y_a \\ 1 & x_b & y_b \\ 1 & x_{v_1} & y_{v_1} \end{pmatrix} \cdot \begin{pmatrix} a_{00}^f \\ a_{10}^f \\ a_{01}^f \end{pmatrix} = \begin{pmatrix} f_a \\ f_b \\ f_{v_1} \end{pmatrix}, \tag{5}$$

where  $f_a = f(x_a, y_a)$ ,  $f_b = f(x_b, y_b)$ ,  $f_{v_1} = f(x_{v_1}, y_{v_1})$ . The analytical solution leads to the following gradient of  $f_{\mathbf{abv}_1}$ :

$$\begin{aligned} \nabla f_{\mathbf{abv}_1}(x, y) &= \begin{pmatrix} a_{10}^f \\ a_{01}^f \end{pmatrix} \\ &= \frac{1}{A_{\mathbf{abv}_1}} \begin{pmatrix} (f_b - f_a)y_{v_1} + (f_a - f_{v_1})y_b + (f_{v_1} - f_b)y_a \\ (f_a - f_b)x_{v_1} + (f_{v_1} - f_a)x_b + (f_b - f_{v_1})x_a \end{pmatrix}, \end{aligned}$$

where  $A_{\mathbf{abv}_1} = x_a(y_b - y_{v_1}) + x_b(y_{v_1} - y_a) + x_{v_1}(y_a - y_b)$  describes the area of the parallelogram spanned by the edges  $\mathbf{v}_1\mathbf{a}$  and  $\mathbf{v}_1\mathbf{b}$ . An analogous calculation can be done for  $f_{\mathbf{av}_2\mathbf{b}}$ ,  $g_{\mathbf{abv}_1}$ , and  $g_{\mathbf{av}_2\mathbf{b}}$ . According to Equation 4, the gradient alignment value at the vertex  $\mathbf{v}_1$  is given by

$$\begin{aligned} \kappa_{\mathbf{v}_1}^{\text{li}}(f, g) &= \nabla f_{\mathbf{abv}_1}(\mathbf{v}_1) \times \nabla g_{\mathbf{abv}_1}(\mathbf{v}_1) \\ &= \frac{(f_b - f_a)g_{v_1} + (f_a - f_{v_1})g_b + (f_{v_1} - f_b)g_a}{A_{\mathbf{abv}_1}}. \end{aligned} \tag{6}$$

Analogously, for the triangle  $\{\mathbf{av}_2\mathbf{b}\}$ , we compute the gradient alignment value at the vertex  $\mathbf{v}_2$  via

$$\begin{aligned} \kappa_{\mathbf{v}_2}^{\text{li}}(f, g) &= (\nabla f_{\mathbf{av}_2\mathbf{b}}(\mathbf{v}_2) \times \nabla g_{\mathbf{av}_2\mathbf{b}}(\mathbf{v}_2)) \\ &= \frac{(f_a - f_b)g_{v_2} + (f_{v_2} - f_a)g_b + (f_b - f_{v_2})g_a}{A_{\mathbf{av}_2\mathbf{b}}}. \end{aligned} \tag{7}$$

These calculations show that the resulting values  $\kappa_{\mathbf{v}_1}^{\text{li}}(f, g)$  and  $\kappa_{\mathbf{v}_2}^{\text{li}}(f, g)$  stem from the gradient alignment field  $\kappa_x^{\text{li}}$ , that is constant on each triangle (because the gradients  $\nabla f$  and  $\nabla g$  are constant on each triangle). With these observations and the fact that the Jacobi set is equal to the zero-level set

of  $\kappa_x$ , it is possible to identify critical edges via a change of sign with respect to  $\kappa_{\mathbf{v}_1}^{\text{li}}(f, g)$  and  $\kappa_{\mathbf{v}_2}^{\text{li}}(f, g)$ .

Comparing these considerations with the approach of Edelsbrunner and Harer [8], we will notice that the two approaches are equivalent. To show this, we consider an edge  $\mathbf{ab}$  with  $\text{Lk}(\mathbf{ab}) = \{\mathbf{v}_1, \mathbf{v}_2\}$ . For the edge  $\mathbf{ab}$  to be critical, either Eq. 1 or Eq. 2 needs to be true. Let us assume that Eq. 1 holds. We can reformulate it in terms of  $\kappa_{\mathbf{v}_1}^{\text{li}}$  and  $\kappa_{\mathbf{v}_2}^{\text{li}}$ :

$$\begin{aligned} &h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) > h_{\lambda_{\mathbf{ab}}}(\mathbf{v}_1) \quad \text{and} \quad h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) > h_{\lambda_{\mathbf{ab}}}(\mathbf{v}_2) \\ \Leftrightarrow &0 > h_{\lambda_{\mathbf{ab}}}(\mathbf{v}_1) - h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) \quad \text{and} \quad 0 > h_{\lambda_{\mathbf{ab}}}(\mathbf{v}_2) - h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) \\ \Leftrightarrow &0 > \frac{(f_b - f_a)g_{v_1} + (f_a - f_{v_1})g_b + (f_{v_1} - f_b)g_a}{g_a - g_b} \\ \text{and} &0 < \frac{(f_a - f_b)g_{v_2} + (f_{v_2} - f_a)g_b + (f_b - f_{v_2})g_a}{g_a - g_b}. \\ \Leftrightarrow &0 > \frac{\kappa_{\mathbf{v}_1}^{\text{li}} \cdot A_{\mathbf{a,b,v}_1}}{g_a - g_b} \quad \text{and} \quad 0 < \frac{\kappa_{\mathbf{v}_2}^{\text{li}} \cdot A_{\mathbf{a,v}_2,\mathbf{b}}}{g_a - g_b}. \end{aligned}$$

An analogous calculation can be done for Eq. 2 with swapped signs for both conditions. As a result, the following relations, which explicitly show the connection between the approach of Edelsbrunner and Harer and the gradient alignment field, can be established:

$$\begin{aligned} \text{Eq1: } &h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) > h_{\lambda_{\mathbf{ab}}}(\mathbf{v}_1) \quad \text{and} \quad h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) > h_{\lambda_{\mathbf{ab}}}(\mathbf{v}_2) \\ \Leftrightarrow &\begin{cases} \text{if } g_a - g_b > 0: & 0 > \kappa_{\mathbf{v}_1}^{\text{li}} A_{\mathbf{abv}_1} \quad \text{and} \quad 0 < \kappa_{\mathbf{v}_2}^{\text{li}} A_{\mathbf{av}_2\mathbf{b}}, \\ \text{if } g_a - g_b < 0: & 0 < \kappa_{\mathbf{v}_1}^{\text{li}} A_{\mathbf{abv}_1} \quad \text{and} \quad 0 > \kappa_{\mathbf{v}_2}^{\text{li}} A_{\mathbf{av}_2\mathbf{b}}, \end{cases} \\ \text{Eq2: } &h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) < h_{\lambda_{\mathbf{ab}}}(\mathbf{v}_1) \quad \text{and} \quad h_{\lambda_{\mathbf{ab}}}(\mathbf{a}) < h_{\lambda_{\mathbf{ab}}}(\mathbf{v}_2) \\ \Leftrightarrow &\begin{cases} \text{if } g_a - g_b > 0: & 0 < \kappa_{\mathbf{v}_1}^{\text{li}} A_{\mathbf{abv}_1} \quad \text{and} \quad 0 > \kappa_{\mathbf{v}_2}^{\text{li}} A_{\mathbf{av}_2\mathbf{b}}, \\ \text{if } g_a - g_b < 0: & 0 > \kappa_{\mathbf{v}_1}^{\text{li}} A_{\mathbf{abv}_1} \quad \text{and} \quad 0 < \kappa_{\mathbf{v}_2}^{\text{li}} A_{\mathbf{av}_2\mathbf{b}}, \end{cases} \end{aligned}$$

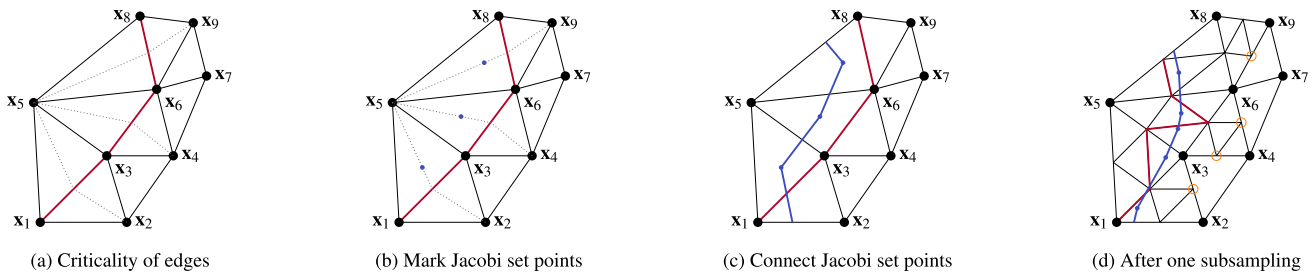
where the term  $g_a - g_b$  determines whether an edge  $\mathbf{ab}$  is maximal or minimal.

At this point, the consequences are twofold. Firstly, the gradient alignment values  $\kappa_{\mathbf{v}_1}^{\text{li}}(f, g)$  and  $\kappa_{\mathbf{v}_2}^{\text{li}}(f, g)$  characterize a new Jacobi set condition for the edge  $\mathbf{ab}$  via

$$\mathbf{ab} \in \mathbb{J}(f, g) \Leftrightarrow \text{sgn}(\kappa_{\mathbf{v}_1}^{\text{li}}(f, g)) \neq \text{sgn}(\kappa_{\mathbf{v}_2}^{\text{li}}(f, g)). \tag{8}$$

Secondly, we observe that the piecewise linear approach by Edelsbrunner and Harer is also equivalent to the condition in Eq. 8. In fact, the conditions only differ in a scaling factor: whereas the gradient alignment field for linearly interpolated functions  $f$  and  $g$  (on each triangle) encodes both the topology and geometry in the condition (due to the scaling  $A_{\mathbf{abv}_1}$  and  $A_{\mathbf{av}_2\mathbf{b}}$ ), the approach by Edelsbrunner and Harer solely relies on topological relations.

Now, the next step is to investigate how the gradient alignment field  $\kappa_x(f, g)$  behaves for bilinearly interpolated functions  $f$  and  $g$ . To this end, let  $f$  and  $g$  be given on the



**Fig. 4** Overview of our local bilinear method with one subdivision step: critical edges (red), Jacobi set points (blue dots), the connected Jacobi set via line segments (blue), and resulting T-nodes (orange). After the detection of critical edges via edge-based (linear) gradient alignment values **(a)**, the Jacobi set points are computed via (bilinear) gradient

alignment values at locations indicated by the dotted lines **(b)**. Then, the Jacobi set points are connected using the underlying connectivity of the critical edges **(c)**. The last illustration shows the result after one subdivision step, including the resulting T-nodes **(d)**

quadrilateral  $\{\mathbf{av}_1 \mathbf{bv}_2\}$ -cell (for the notation, see Fig. 3) by the coefficients  $b_{ij}^f$  and  $b_{ij}^g$ , respectively, and the formula

$$f(x, y) = b_{00}^f + b_{10}^f x + b_{01}^f y + b_{11}^f xy,$$

$$g(x, y) = b_{00}^g + b_{10}^g x + b_{01}^g y + b_{11}^g xy.$$

The coefficients can be analogously computed to Eq. 5 via a system of linear equations. However, in this case the gradients are not constant and are given by

$$\nabla f(x, y) = \begin{pmatrix} b_{10}^f + b_{11}^f y \\ b_{01}^f + b_{11}^f x \end{pmatrix}, \quad \nabla g(x, y) = \begin{pmatrix} b_{10}^g + b_{11}^g y \\ b_{01}^g + b_{11}^g x \end{pmatrix}.$$

Hence, the gradient alignment field  $\kappa_x$  is linear as the following general calculation shows:

$$\begin{aligned} \kappa_x^{bi}(f, g) &= \nabla f(x, y) \times \nabla g(x, y) \\ &= (b_{10}^f b_{01}^g - b_{01}^f b_{10}^g) + (b_{10}^f b_{11}^g - b_{11}^f b_{10}^g) x \\ &\quad + (b_{11}^f b_{01}^g - b_{01}^f b_{11}^g) y. \end{aligned} \tag{9}$$

This aspect enables the computation of the zero-level set in the quadrilateral, instead of identifying critical edges of the underlying triangulation. In fact, the method by Edelsbrunner and Harer can be extended using a bilinear interpolation scheme. To be more precise, the gradient alignment values  $\kappa_{v_1}^{li}$  and  $\kappa_{v_2}^{li}$  (see Eq. 6 and Eq. 7) are used for the topological identification of critical edges, whereas for the geometrical identification a bilinear interpolation scheme is used.

To this end, the gradient alignment values  $\kappa_{v_1}^{bi}$  and  $\kappa_{v_2}^{bi}$  in the case of bilinearly interpolated  $f$  and  $g$  are given by the following formulas (we skip the technical details):

$$\begin{aligned} \kappa_{v_1}^{bi} &= \frac{1}{D} \cdot (A_{abv_1} \cdot \kappa_{v_1}^{li} \cdot (x_{v_2} y_{v_1} + x_{v_1} y_{v_2} - x_{v_2} y_{v_2}) \\ &\quad + A_{av_2 v_1} \cdot \kappa_a^{li} \cdot (x_{v_1} y_b + x_b y_{v_1} - x_b y_b) + A_{av_2 b} \cdot \kappa_{v_2}^{li} \cdot (x_{v_1} y_{v_1}) \end{aligned}$$

$$+ A_{bv_1 v_2} \cdot \kappa_b^{li} \cdot (x_{v_1} y_a + x_a y_{v_1} - x_a y_a)), \tag{10}$$

$$\begin{aligned} \kappa_{v_2}^{bi} &= \frac{-1}{D} \cdot (A_{av_2 b} \cdot \kappa_{v_2}^{li} \cdot (x_{v_1} y_{v_1} - x_{v_1} y_{v_2} - x_{v_2} y_{v_1}) \\ &\quad + A_{av_2 v_1} \cdot \kappa_a^{li} \cdot (x_b y_{v_2} + x_{v_2} y_b - x_b y_b) + A_{abv_1} \cdot \kappa_{v_1}^{li} \cdot (x_{v_2} y_{v_2}) \\ &\quad + A_{bv_1 v_2} \cdot \kappa_b^{li} \cdot (x_a y_a - x_a y_{v_2} - x_{v_2} y_a)). \end{aligned} \tag{11}$$

The values  $\kappa_a^{li}$  and  $\kappa_b^{li}$  are given by

$$\begin{aligned} \kappa_a^{li}(f, g) &:= \frac{(f_{v_1} - f_{v_2})g_a + (f_{v_2} - f_a)g_{v_1} + (f_a - f_{v_1})g_{v_2}}{A_{av_2 v_1}} \\ \kappa_b^{li}(f, g) &:= \frac{(f_{v_2} - f_{v_1})g_b + (f_{v_1} - f_b)g_{v_2} + (f_b - f_{v_2})g_{v_1}}{A_{bv_1 v_2}} \end{aligned}$$

with the area for counter-clockwise sorted  $\mathbf{p}_1, \mathbf{p}_2$ , and  $\mathbf{p}_3$ :

$$\begin{aligned} A_{\mathbf{p}_1 \mathbf{p}_2 \mathbf{p}_3} &:= \det \begin{pmatrix} x_{\mathbf{p}_1} & x_{\mathbf{p}_2} & x_{\mathbf{p}_3} \\ y_{\mathbf{p}_1} & y_{\mathbf{p}_2} & y_{\mathbf{p}_3} \\ 1 & 1 & 1 \end{pmatrix} \\ &= x_{\mathbf{p}_1}(y_{\mathbf{p}_2} - y_{\mathbf{p}_3}) + x_{\mathbf{p}_2}(y_{\mathbf{p}_3} - y_{\mathbf{p}_1}) + x_{\mathbf{p}_3}(y_{\mathbf{p}_1} - y_{\mathbf{p}_2}). \end{aligned}$$

The denominator  $D$  is computed via

$$\begin{aligned} D &= ((x_a - x_{v_1}) \cdot (x_b - x_{v_2})) \cdot (y_a y_{v_1} + y_b y_{v_2}) \\ &\quad + ((x_a - x_b) \cdot (x_{v_2} - x_{v_1})) \cdot (y_a y_b + y_{v_1} y_{v_2}) \\ &\quad + ((x_a - x_{v_2}) \cdot (x_{v_1} - x_b)) \cdot (y_a y_{v_2} + y_b y_{v_1}). \end{aligned}$$

### 4.2 Algorithm

An overview of our local bilinear method is given in Fig. 4, and the algorithm can be divided into three steps:

1. Computation of linear gradient alignment values  $\kappa_x^{li}$  with the identification of critical edges.
2. Computation of Jacobi set points via the bilinear gradient alignment values  $\kappa_x^{bi}$ .
3. Generation of the Jacobi set representation  $\mathbb{J}(f, g)$ .

### 4.2.1 Identification of critical edges

In the previous subsection, we formulated a method (compare Eq. 8), that is topologically equivalent to the one of Edelsbrunner and Harer [8] in terms of the (linear) gradient alignment values  $\kappa_{v_1}^{li}$  and  $\kappa_{v_2}^{li}$ . We use these values to identify critical Jacobi set edges. This is the first part of our method and also the initial step in Algorithm 1. To be more precise, given edges  $E = (e_i)$ , we iterate over each edge  $e_i$  (line 2), identify  $v_1$  and  $v_2$  as the lower link of  $e_i$  (line 3), and compute the gradient alignment values  $\kappa_{v_1}^{li}$  and  $\kappa_{v_2}^{li}$  via Eq. 6 and Eq. 7 (line 4). Afterward, the critical edge test via the change of sign method is performed (line 5).

While the piecewise linear approach stops here (identifying critical edges of the underlying triangulation), the next step of our method is to determine a geometrically more precise Jacobi set representation via Jacobi set points.

### 4.2.2 Computation of Jacobi set points

In contrast to the piecewise linear computation by Edelsbrunner and Harer, we avoid marking critical edges of the underlying triangulation by introducing *Jacobi set points*. As illustrated in Fig. 4a, a Jacobi set point is conceptually assigned to a critical edge  $ab$  and lies on the line segment that connects the vertices  $v_1$  and  $v_2$  via  $m$  (compare Fig. 2).

---

#### Algorithm 1: Computation of Jacobi set points

---

**Input:** Scalar fields  $f$  and  $g$ , edges  $E = (e_i)$

**Output:**  $JE = (e_i, p_i)$ : List of critical edges  $e_i$  with corresponding Jacobi set points  $p_i$

```

1 begin
2   for  $e_i \in E$  do
3     Find  $v_1, v_2 \in \text{Lk}(e_i)$ 
4     Compute  $\kappa_{v_1}^{li}, \kappa_{v_2}^{li}$  // Eq. 6 and Eq. 7
5     if  $\text{sgn}(\kappa_{v_1}^{li}) \neq \text{sgn}(\kappa_{v_2}^{li})$  then
6       Compute  $\kappa_{v_1}^{bi}, \kappa_{v_2}^{bi}$  // Eq. 10 and Eq. 11
7       if  $\text{sgn}(\kappa_{v_1}^{bi}) \neq \text{sgn}(\kappa_{v_2}^{bi})$  then
8          $\lambda = \kappa_{v_1}^{bi} / (\kappa_{v_2}^{bi} - \kappa_{v_1}^{bi})$ 
9       else
10         $\lambda = \kappa_{v_1}^{li} / (\kappa_{v_2}^{li} - \kappa_{v_1}^{li})$ 
11         $m = a + (b - a) / 2$ 
12        if  $\lambda < 1/2$  then
13           $p_i = v_1 + 2\lambda(m - v_1)$ 
14        else
15           $p_i = m + (1 - 2\lambda)(m - v_2)$ 
16        add  $(e_i, p_i)$  to  $JE$ 

```

---

Given the fact that bilinearly interpolated functions  $f$  and  $g$  result in a linear gradient alignment field on the quadrilateral  $\{av_1bv_2\}$ -cell, there is only one zero on this line segment if the signs of the values  $\kappa_{v_1}$  and  $\kappa_{v_2}$  differ. In this context, the zero of the line segment is given by

$$0 = \kappa_{v_1} + \lambda \cdot (\kappa_{v_2} - \kappa_{v_1}) \Leftrightarrow \lambda = -\frac{\kappa_{v_1}}{\kappa_{v_2} - \kappa_{v_1}},$$

where  $\lambda$  is the weighting coefficient that characterizes the location of the zero via

$$p_i = \begin{cases} v_1 + 2\lambda \cdot (m - v_1) & \text{if } \lambda < 1/2, \\ m + (1 - 2\lambda) \cdot (m - v_2) & \text{if } \lambda \geq 1/2. \end{cases} \quad (12)$$

Thus, the main idea is that the collection of all Jacobi set points approximates the Jacobi set (see Fig. 4b and c). The computation of Jacobi set points is the second part of Algorithm 1 (starting with a critical edge  $e_i$  in line 6) and will be explained in the following in more detail.

To compute a Jacobi set point, we compute the bilinear gradient alignment values  $\kappa_{v_1}^{bi}$  and  $\kappa_{v_2}^{bi}$  (line 6). Then, we perform the change of sign method for  $\kappa_{v_1}^{bi}$  and  $\kappa_{v_2}^{bi}$  (line 7). In the case of different signs, the weighting coefficient  $\lambda$  characterizing the Jacobi set point is computed via bilinear gradient alignment values  $\kappa_{v_1}^{bi}$  and  $\kappa_{v_2}^{bi}$  (line 8). Otherwise, linear gradient alignment values  $\kappa_{v_1}^{li}$  and  $\kappa_{v_2}^{li}$  are used (line 10). Finally the exact location of the Jacobi set point is computed via Eq. 12 (line 11–15) and added to the list of pairs  $(e_i, p_i)$  of Jacobi set points  $p_i$  and critical edges  $e_i$  (line 16).

### 4.2.3 Connection of Jacobi set points

Given a collection of Jacobi set points computed by Algorithm 1, our goal is to connect them in a way such that they geometrically approximate the Jacobi set  $\mathbb{J}(f, g)$ . To achieve this, we utilize the preserved topological properties of the (linear) gradient alignment values (from Edelsbrunner and

---

#### Algorithm 2: Connection of Jacobi set points

---

**Input:**  $JE = (e_i, p_i)$ : List of critical edges  $e_i$  with corresponding Jacobi set points  $p_i$

**Output:**  $JS = \{l_k = (p_{k_i}, p_{k_j})\}$ : Jacobi set lines

```

1 begin
2   for  $e_i$  do
3     Find critical edges  $e_{i_1}, \dots, e_{i_d}$  that share a
       vertex with the edge  $e_i$ 
4     for  $e_{i_j}$  do
5       add line segment  $l = (p_i, p_j)$  to  $JS$ 

```

---

Harer, see Eq. 8) to apply the Even Degree Lemma. According to the lemma, the vertices  $\mathbf{a}$  and  $\mathbf{b}$  of a Jacobi edge  $\mathbf{ab}$  have an even degree. It implies that a Jacobi edge  $\mathbf{ab}$  is connected to an even number of other Jacobi edges. This is due to the fact that the degrees of  $\mathbf{a}$  and  $\mathbf{b}$  are even and since both degrees include the edge  $\mathbf{ab}$  as well, the number of Jacobi edges connected to  $\mathbf{ab}$  equals  $(\text{degr}(\mathbf{a}) - 1) + (\text{degr}(\mathbf{b}) - 1)$ . The important consequence of this is:

Each Jacobi set point is connected to an even number (greater than zero) of Jacobi set points and, in particular, there are no isolated Jacobi set points.

Based on this result, we can describe a method to connect Jacobi set points. An illustration of this method is presented in Fig. 5, and a detailed description is given by Algorithm 2, which will be explained in the following. Given the list of pairs  $(\mathbf{e}_i, \mathbf{p}_i)$  of Jacobi set points  $\mathbf{p}_i$  that are related to critical edges  $\mathbf{e}_i$ , we iterate over each critical edge  $\mathbf{e}_i$  (line 2) to find all critical edges  $\mathbf{e}_{i_j}$  that share a vertex with  $\mathbf{e}_i$  (line 3). These are used to construct line segments  $l = (\mathbf{p}_i, \mathbf{p}_j)$  with the respective Jacobi set points  $\mathbf{p}_i$  and  $\mathbf{p}_j$  (line 4-5). This method produces a collection of line segments that geometrically approximates the Jacobi set while preserving the topology.

Algorithm 2 could be considered as a “dualization” process where each critical edge is replaced by its corresponding Jacobi set point (as its “dual”); in fact, two Jacobi set points are connected by a line segment if their corresponding critical edges share the same vertex. This aspect is visualized in Fig. 5, where we use a color-coding and numbering to highlight the connectivity. In the top, the pairs of critical edges and Jacobi set points are represented (the dotted lines emphasize the pairs). The bottom image shows the Jacobi set representation with our connection method. The effect of our connection method is that the adjacency of critical edges (e.g., edge 1 is connected to the edges 0, 2, 3 and 5) is reflected by the connectivity of Jacobi set points (see the respective Jacobi sets points). In this way, the topology of the piecewise linear method is preserved. We provide a formal proof of the homotopy equivalence in Appendix.

### 4.3 Local subdivision method

One of the advantages of our local bilinear approach is the usage of a geometrically more precise extraction of the Jacobi set. This typically leads to better convergence properties compared to piecewise linear schemes. Therefore, we propose a local subdivision scheme that makes use of the already identified critical edges by only doing computations in these locations. Given a critical edge  $\mathbf{e}_i = \mathbf{ab}$  and the related Jacobi set point  $\mathbf{p}_i$  within a  $\{\mathbf{av}_1\mathbf{bv}_2\}$ -cell (see Fig. 6), we now describe a single subdivision step for the edge  $\mathbf{ab}$ :

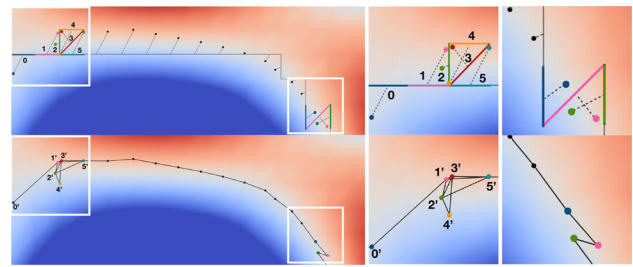
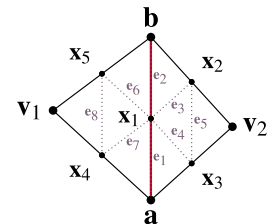


Fig. 5 Illustration of our connection method for the computed Jacobi set points. The top shows our computed Jacobi set points along with the piecewise linear method, and the bottom illustrates how our method connects the points. A protruding triangle (resulting in a loop) is shown on the left, whereas a straightforward topology is presented on the right

Fig. 6 Local refinement around a critical edge  $\mathbf{ab}$



- For the critical edge  $\mathbf{ab}$ , bisect each edge that belongs to the two adjacent triangles and connect the new vertices  $\mathbf{x}_1, \dots, \mathbf{x}_5$  such that 8 smaller triangles are obtained.
- Use an interpolant to compute the function values of  $f$  and  $g$  at the new vertices  $\mathbf{x}_1, \dots, \mathbf{x}_5$ .
- Our local bilinear method can now be applied to the resulting new edges  $\mathbf{e}_1, \dots, \mathbf{e}_8$ .

This procedure is done for each critical edge, which means a local efficient subdivision scheme is used. An illustration is presented in Fig. 4d. It can be observed that T-nodes occur, which violate the rules of a triangulation. However, for our purposes, we do not need a consistent global triangulation for further computations. The new critical edges and Jacobi set points will only be computed in the local neighborhood, and therefore, the T-nodes do not affect the repeated application of the local subdivision method.

## 5 Results

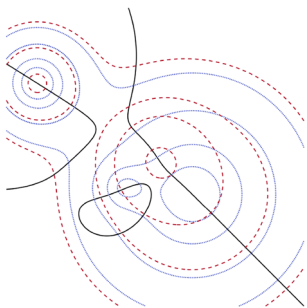
To evaluate our local bilinear method qualitatively and quantitatively, we apply it to three different datasets: an analytical example, a simulated von Kármán vortex street, and the multi-field Hurricane Isabel dataset. Additionally, it is compared to the piecewise linear method by Edelsbrunner and Harer [8]. The methods are performed using MATLAB (R2022a) on a MacBook Pro with an Intel Dual-Core i5 CPU @ 3.1 GHz and 8 GB of RAM. The computation times for each of the datasets are given in Table 1.

For the quantitative analysis, we introduce the *gradient alignment measure*  $\mu$  that measures the accuracy of the



**Table 1** Comparison of computation times (in seconds). For the piecewise linear method the time for identifying critical edges was measured. For the local bilinear method with and without subdivision (subd.) the computation times of Algorithm 1 and 2 were added

	Analytic	Kármán	Hurricane
Piecewise linear	0.0055	0.0384	0.2281
Local bilinear	0.0121	0.0728	0.5318
Local bilinear subd.	0.0219	0.1731	2.0936



**Fig. 7** Contour lines of the analytic functions  $f$  (red dashed lines) and  $g$  (blue dotted lines) with the resulting analytic Jacobi set  $\mathbb{J}(f, g)$  marked in black

extracted Jacobi set. Given a globally computed gradient alignment field  $\kappa_{\mathbf{x}}(f, g)$  as well as a Jacobi set representation  $JS = \{l_k = (\mathbf{p}_{k_i}, \mathbf{p}_{k_j})\}$  via a collection of line segments, we define the gradient alignment measure as

$$\begin{aligned} \mu(JS) &:= \frac{\int_{JS} \kappa_{\mathbf{x}}(f, g) \, dx}{\int_{JS} 1 \, dx} = \frac{\sum_k \int_{l_k} \kappa_{\mathbf{x}}(f, g) \, dx}{\sum_k \int_{l_k} 1 \, dx} \\ &\approx \frac{\sum_k \frac{1}{2} \left( |\kappa_{\mathbf{p}_{k_i}}(f, g)| + |\kappa_{\mathbf{p}_{k_j}}(f, g)| \right) \|\mathbf{p}_{k_j} - \mathbf{p}_{k_i}\|_2}{\sum_k \|\mathbf{p}_{k_j} - \mathbf{p}_{k_i}\|_2}. \end{aligned}$$

In short, the measure computes the deviation of the Jacobi set with regard to the globally computed gradient alignment field  $\kappa_{\mathbf{x}}(f, g)$ . The background of this measure is that the gradient alignment value is zero for  $\mathbf{x} \in \mathbb{J}(f, g)$ .

### 5.1 Analytic dataset

The first dataset consists of two artificially generated scalar fields. Both are constructed by the sum of three bivariate normal distributions on the unit square. To get an impression of the two functions, a visualization of the contour lines is presented in Fig. 7. The figure also shows the Jacobi set of these two functions (black line) given by an analytical formula. Therefore, we can compare the Jacobi set extracted by our local bilinear approach (or by the piecewise linear approach) with the true Jacobi set.

In Fig. 1, the Jacobi sets are computed by the piecewise linear approach (top row) and our local bilinear method (bottom row) for a fixed resolution and different zoom levels on a (triangulated) rectangular grid. First of all, both methods identify the fundamental topological structure correctly. For each zoom level, the piecewise linear approach produces a solution that highly suffers from the edge-based geometric representation of the Jacobi set (which stems from the underlying triangulation), leading to zig-zag patterns. Furthermore, the zoomed-in image (top right) reveals that the Jacobi set representation consists of several (“glued”) triangles of the underlying triangulation, resulting in misleading topological patterns. While our local bilinear method obtains the same topology, these triangles are merged geometrically, leading to a clearer geometric representation. In sum, our local bilinear method results in a smoother and more precise Jacobi set representation (e.g., see third column).

In the following, we also evaluate the proposed local subdivision method. It can be applied to both our local bilinear method and the piecewise linear method. The qualitative results are presented in Fig. 8, and the corresponding gradient alignment measure is depicted in Fig. 9.

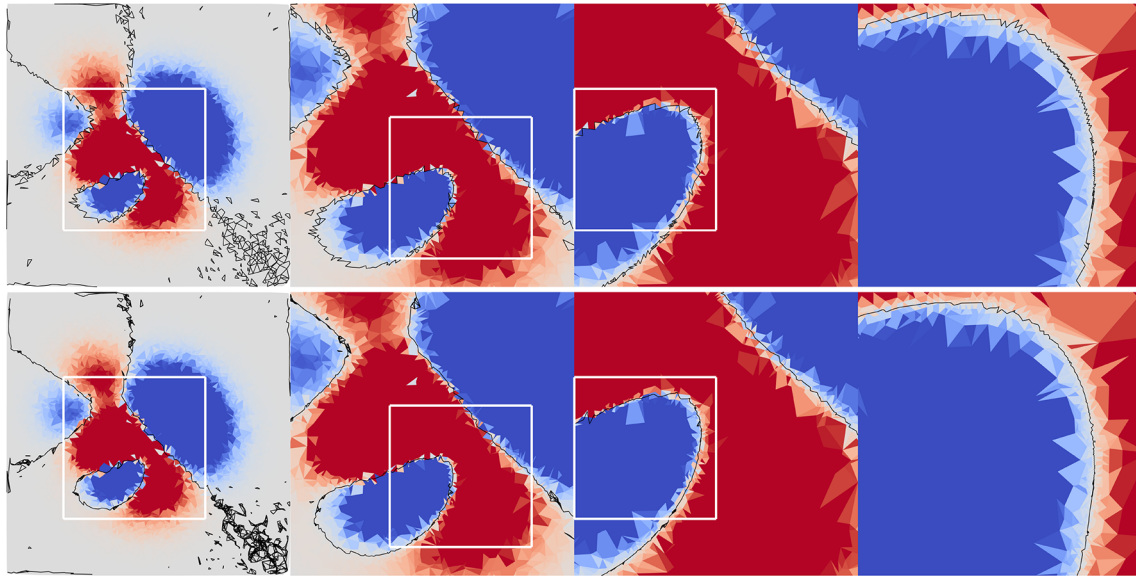
In general, both Jacobi set representations produce incorrectly detected regions and artifacts for the initial configuration, especially at the bottom right. This is due to the coarse irregular mesh and resulting degenerate areas. With the application of subdivision, both methods produce better results; however, our local bilinear method results in much smoother representations. This observation can also be noticed in the error plot in Fig. 9. The gradient alignment measure shows that our approach results in less error and has a faster convergence than the piecewise linear method.

### 5.2 Von Kármán vortex street dataset

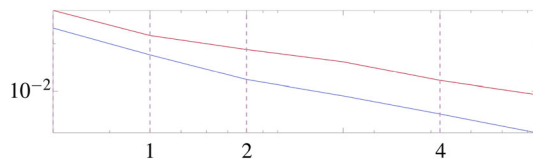
While the analytic dataset provides first insights into our approach, we now apply our method to a simulated *von Kármán vortex street* [15]. The dataset consists of 1501 time steps describing the velocity and has a resolution of  $640 \times 80$ . We use the magnitudes of the flow velocity for two consecutive time steps as the two scalar fields for the Jacobi set computation. For our examples, we start at time step 1500, where the vortex street is fully emerged.

A visualization of the von Kármán vortex street for this time step is depicted in Fig. 10 top. The figure also shows the Jacobi set computed by the piecewise linear approach (middle) and our approach (bottom). In general, we observe that the Jacobi set captures the topological structure of each Kármán vortex. However, similar to previous observations, the zig-zag patterns of the piecewise linear approach make it hard to distinguish between different topological structures.

For a zoomed-in view, the top rows of Fig. 11 show the framed area in Fig 10 in more detail. We observe that both



**Fig. 8** Comparison of 0, 1, 2, and 4 subdivision steps with adjusted zoom levels (left to right) for the piecewise linear (top) and local bilinear (bottom) method using the analytic dataset on a triangulated mesh

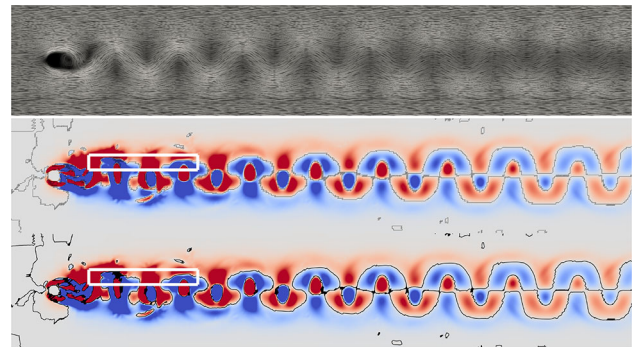


**Fig. 9** The gradient alignment measure for different subdivision steps (analogously to Fig. 8) for the piecewise linear (red) and our local bilinear (blue) method. The vertical dashed lines correspond to the snapshots in Fig. 8

methods fail to produce the correct topological structure at the top left (compare the gradient alignment field in the background). However, due to our connection method, our Jacobi set representation suggests that this region is problematic and may contain wrongly detected areas. In the piecewise linear case, the regular identification of Jacobi set patterns via the underlying triangulation makes it hard to distinguish between correctly and incorrectly detected areas. Besides of that, at the middle and top right areas, our local bilinear method captures the geometry better due to the smooth Jacobi set representation.

Another comparison of these two methods is done by using the local subdivision method (rows 2–4 in Fig. 11). We observe for both methods that the incorrect topological structures in the top left disappear after one subdivision step. Another important aspect is shown in the third row (with two subdivision steps). Topologically identified patterns, such as the isolated triangle at the bottom left, can disappear visually for our bilinear method due to geometrical contraction.

with  $80 \times 80$  data points. The globally computed gradient alignment values are not interpolated to emphasize the subdivision process



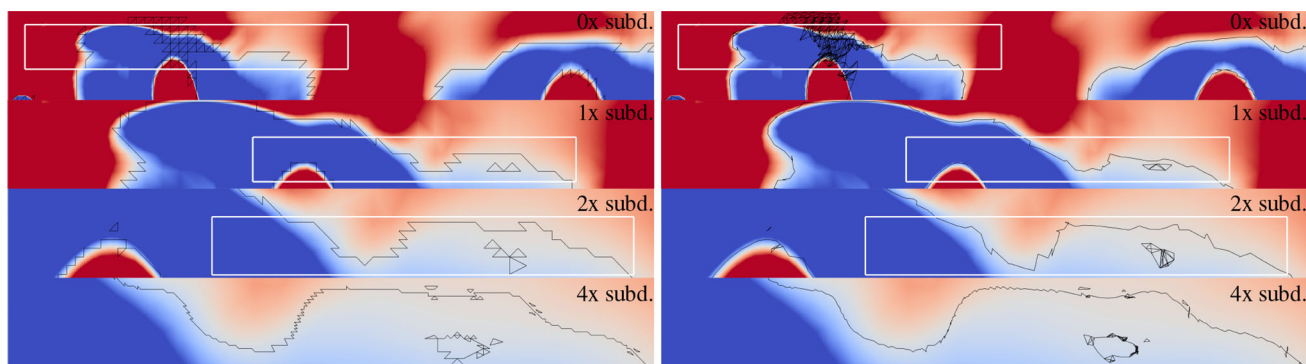
**Fig. 10** LIC visualization of the fully emerged von Kármán vortex street (top) as well as a comparison of the Jacobi set computed with the piecewise linear method (middle) and our method (bottom). The two scalar fields represent the flow velocity magnitude for two consecutive time steps. The boxed-in areas will be discussed in more detail in Fig. 11

This can never happen for the piecewise linear approach. Applying subdivision four times, we observe that the representation of topological structures is significantly improved. While the piecewise linear approach identifies the circular shape rather poorly (due to the underlying grid), the geometry of our method captures the circularity more clearly (in the limit, the shape is nearly an ellipse).

### 5.3 Hurricane Isabel dataset

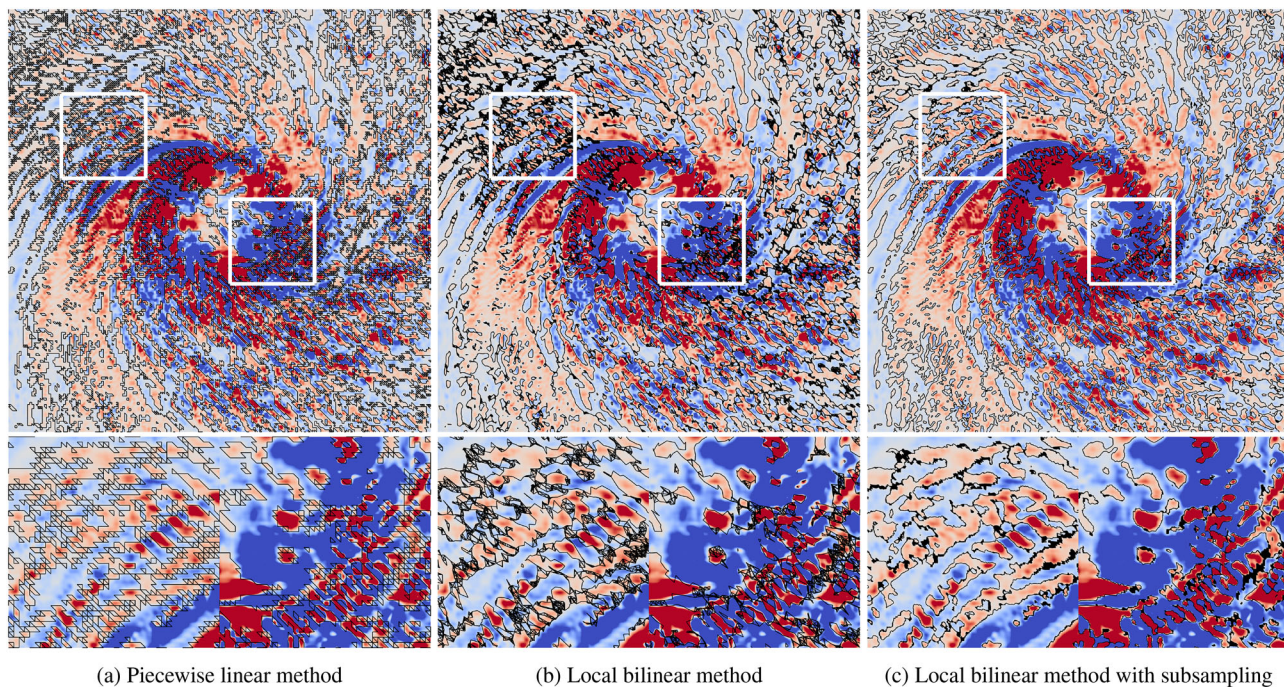
Jacobi sets are motivated as a topological descriptor of multiple scalar fields with different quantities. Here, we perform





**Fig. 11** For the piecewise linear (left) and local bilinear (right) computation of the Jacobi set, the top rows show the zoomed-in view of Fig. 10 (see the white box). As scalar functions, flow magnitudes of two con-

secutive time steps are used. Furthermore, the subdivision method is applied in the zoomed-in areas for both methods



**Fig. 12** Comparison of the piecewise linear (a) and local bilinear (b) computation of Jacobi sets for the Hurricane Isabel dataset. For the Jacobi set, a pressure and temperature scalar field is used. Furthermore, our local bilinear method with two subdivision steps is illustrated (c)

the Jacobi set extraction on the multi-field real-world dataset Hurricane Isabel<sup>1</sup>. The dataset consists of multiple 3D scalar fields and a 3D velocity vector field that have a resolution of  $500 \times 500 \times 100$  and 48 time steps.

We use the pressure and temperature scalar field at time step 30 and height 50 to compare the Jacobi set computation methods in Fig. 12. At this height, the hurricane has a large spatial expansion and reveals many characteristic structures.

Comparing the piecewise linear method (Fig. 12a) with our method (Fig. 12b), we observe that many areas are marked

in the top left region. Whereas in the piecewise linear case it is nearly impossible to track topological structures, our method (without subdivision) already has a clearer representation due to geometrical enhancements. However, it can be observed that the preservation of topology leads to a folding of line segments, which can clutter areas in the representation (see exemplary Fig. 5). Our subdivision method (Fig 12c) solves this issue by thinning out those areas. This leads to a much clearer representation.

To further demonstrate that our proposed local subdivision method improves the results, the other zoomed-in area at the center shows that the identified topological structures match more clearly the zero-level set (white areas) of the gradient

<sup>1</sup> Hurricane Isabel data produced by the Weather Research and Forecast (WRF) model, courtesy of NCAR and the U.S. National Science Foundation (NSF)

alignment field in the background. In fact, several connected topological patterns are separated correctly.

## 6 Conclusion

In this paper, we recapped the piecewise linear computation of Jacobi sets and embedded its theory into our new formulation. This view enables a new formulation that extends the piecewise linear computation to be based on a bilinear interpolation scheme. While preserving the topological structures, our bilinear interpolation results in a smoother and geometrically more accurate Jacobi set representation. In addition, our efficient local subdivision method tailored to our approach further improves the Jacobi set representation.

In the future, we could extend our local bilinear method to higher dimensions as well as more than two scalar functions. Additionally, other interpolation methods could be used that are based on finite elements or higher interpolation schemes such as biquadratic or bicubic interpolation.

Another idea is to evaluate our local bilinear method in terms of detecting new features. This not only includes the raw results but also the representation via connected line segments. In this regard, the appearance and the spatial spread of the connected line segments (representing a band-like structure) could contain more information about the underlying data, e.g., uncertainty or other features. In addition, since our method significantly facilitates the investigation of Jacobi sets, another research direction is to find new interpretations and applications of Jacobi sets across diverse research areas, such as image processing or computer vision.

**Acknowledgements** This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the grants DFG 270852890-GRK 2160/2 and DFG 251654672-TRR 161, the Swedish Research Council (VR) under the grant 2019-05487, the U.S. Department of Energy (DOE) under the grant DOE DE-SC0021015, and the National Science Foundation (NSF) through the grant NSF IIS-1910733.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

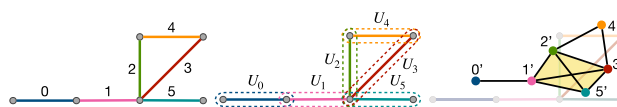


Fig. 13 Applying Nerve Lemma to a piece of the Jacobi set

## Appendix: Proof of homotopy equivalence

In Algorithm 2, each critical edge is replaced by its corresponding Jacobi set point, where two Jacobi set points are connected by a line segment if their corresponding critical edges share the same vertex. We claim that such a mapping preserves the topology due to the Nerve Lemma (see [16, Corollary 4G.3, page 459] and [9, Nerve Theorem, page 71]). In our context, as shown in Fig. 13 left, the input to Algorithm 2 is a set of critical edges  $e_i$ , which form a simplicial complex  $K$  together with their vertices. Recall the closure of a simplex contains the set of its faces. We form a cover  $\mathcal{U}$  of  $K$  by treating the closure of each critical edge  $e_i$  as a cover element  $U_i := \text{Cl}(e_i)$ , where  $K \subseteq \bigcup_{U_i \in \mathcal{U}} U_i$  (see Fig. 13 middle). Each cover element  $U_i$  is a convex and closed set, and the intersection of  $U_i$  and  $U_j$  is contractible. Therefore, the Nerve Lemma applies and the nerve of  $\mathcal{U}$ , which contains a 3-simplex (a yellow tetrahedron) as shown in Fig. 13 right, is homotopy equivalent to  $K$ . The output of Algorithm 2 is by construction precisely the 1-skeleton of the nerve of  $\mathcal{U}$ .

## References

- Artamonova, V., Alekseev, V.V., Makarenko, N.G.: Gradient measure and Jacobi sets for estimation of interrelationship between geophysical multifields. *J. Phys.: Conf. Ser.* **798**, 012–040 (2017). <https://doi.org/10.1088/1742-6596/798/1/012040>
- Barnett, T.P., Pierce, D.W., Schnur, R.: Detection of anthropogenic climate change in the world's oceans. *Science* **292**(5515), 270–274 (2001). <https://doi.org/10.1126/science.1058304>
- Bhatia, H., Wang, B., Norgard, G., Pascucci, V., Bremer, P.T.: Local, smooth, and consistent Jacobi set simplification. *Comput. Geom.* **48**(4), 311–332 (2015). <https://doi.org/10.1016/j.comgeo.2014.10.009>
- Bremer, P.T., Bringa, E.M., Duchaineau, M.A., Gyulassy, A.G., Laney, D., Mascarenhas, A., Pascucci, V.: Topological feature extraction and tracking. *J. Phys. Conf. Ser.* **78**(012), 007 (2007). <https://doi.org/10.1088/1742-6596/78/1/012007>
- Carlsson, G., Zomorodian, A.: The theory of multidimensional persistence. *Discrete Comput. Geom.* **42**(1), 71–93 (2009). <https://doi.org/10.1007/s00454-009-9176-0>
- Carr, H., Duke, D.: Joint contour nets. *IEEE Trans. Vis. Comput. Graph.* **20**(8), 1100–1113 (2014). <https://doi.org/10.1109/TVCG.2013.269>
- Chattopadhyay, A., Carr, H., Duke, D., Geng, Z.: Extracting Jacobi structures in Reeb spaces. In: *EuroVis—Short Papers* (2014). <https://doi.org/10.2312/eurovisshort.20141156>
- Edelsbrunner, H., Harer, J.: Jacobi sets of multiple Morse functions. In: *Found. Comput. Math.*, pp. 37–57 (2002). <https://doi.org/10.1017/CBO9781139106962.003>
- Edelsbrunner, H., Harer, J.: *Computational Topology: An Introduction*. American Math. Soc, Providence (2010)



10. Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Local and global comparison of continuous functions. In: IEEE Visualization, pp. 275–280 (2004). <https://doi.org/10.1109/VISUAL.2004.68>
11. Edelsbrunner, H., Harer, J., Patel, A.K.: Reeb spaces of piecewise linear mappings. In: Proc. 24th Ann. Sympos. Comput. Geom., pp. 242–250 (2008). <https://doi.org/10.1145/1377676.1377720>
12. Edelsbrunner, H., Morozov, D., Patel, A.: The stability of the apparent contour of an orientable 2-manifold. In: Topological Methods in Data Analysis and Visualization, pp. 27–41. Springer (2011). [https://doi.org/10.1007/978-3-642-15014-2\\_3](https://doi.org/10.1007/978-3-642-15014-2_3)
13. Edelsbrunner, H., Mücke, E.P.: Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. ACM Trans. Graph. **9**(1), 66–104 (1990). <https://doi.org/10.1145/77635.77639>
14. Gelder, A.V., Pang, A.: Using PVsolve to analyze and locate positions of parallel vectors. IEEE Trans. Vis. Comput. Graph. **15**(4), 682–695 (2009). <https://doi.org/10.1109/TVCG.2009.11>
15. Günther, T., Gross, M., Theisel, H.: Generic objective vortices for flow visualization. ACM Trans. Graph. **36**(4), 141 (2017). <https://doi.org/10.1145/3072959.3073684>
16. Hatcher, A.: Algebraic Topology. Cambridge Univ Press, Cambridge (2002)
17. He, X., Tao, Y., Wang, Q., Lin, H.: Multivariate spatial data visualization: a survey. J. Vis. **22**(5), 897–912 (2019). <https://doi.org/10.1007/s12650-019-00584-3>
18. Huettnerberger, L., Garth, C.: A comparison of Pareto sets and Jacobi sets. In: Topological and Statistical Methods for Complex Data, pp. 125–141. Springer (2015). [https://doi.org/10.1007/978-3-662-44900-4\\_8](https://doi.org/10.1007/978-3-662-44900-4_8)
19. Huettnerberger, L., Heine, C., Carr, H., Scheuermann, G., Garth, C.: Towards multifield scalar topology based on Pareto optimality. Comput. Graph. Forum **32**(3), 341–350 (2013). <https://doi.org/10.1111/cgf.12121>
20. Iuricich, F., Scaramuccia, S., Landi, C., De Floriani, L.: A discrete Morse-based approach to multivariate data analysis. In: SIGGRAPH Asia Symp. Vis., 5:1–8 (2016). <https://doi.org/10.1145/3002151.3002166>
21. Luo, C., Safa, I., Wang, Y.: Approximating gradients for meshes and point clouds via diffusion metric. Comput. Graph. Forum **28**(5), 1497–1508 (2009). <https://doi.org/10.1111/j.1467-8659.2009.01526.x>
22. Makela, K., Ophelders, T., Quigley, M., Munch, E., Chitwood, D.H., Downtin, A.: Automatic tree ring detection using Jacobi sets. Comput. Res. Rep. [arXiv:abs/2010.08691](https://arxiv.org/abs/2010.08691) (2020). <https://doi.org/10.48550/arXiv.2010.08691>
23. Nagaraj, S., Natarajan, V.: Relation-aware isosurface extraction in multifield data. IEEE Trans. Vis. Comput. Graph. **17**(2), 182–191 (2010). <https://doi.org/10.1109/TVCG.2010.64>
24. Nagaraj, S., Natarajan, V.: Simplification of Jacobi sets. In: Topological Data Analysis and Visualization, pp. 91–102. Springer (2011). [https://doi.org/10.1007/978-3-642-15014-2\\_8](https://doi.org/10.1007/978-3-642-15014-2_8)
25. Natarajan, V.: Topological analysis of scalar functions for scientific data visualization. Ph.D. thesis, Duke University (2004)
26. Norgard, G., Bremer, P.T.: Ridge-valley graphs: Combinatorial ridge detection using Jacobi sets. Comput. Aided Geom. Des. **30**(6), 597–608 (2013). <https://doi.org/10.1016/j.cagd.2012.03.015>
27. Peikert, R., Roth, M.: The “parallel vectors” operator - a vector field visualization primitive. In: IEEE Visualization, vol. 14, pp. 263–270 (1999). <https://doi.org/10.1109/VISUAL.1999.809896>
28. Singh, G., Mémoli, F., Carlsson, G.: Topological methods for the analysis of high dimensional data sets and 3D object recognition. In: Eurographics Symposium on Point-Based Graphics, pp. 91–100 (2007). <https://doi.org/10.2312/SPBG/SPBG07/091-100>
29. Theisel, H., Sahner, J., Weinkauff, T., Hege, H.C., Seidel, H.P.: Extraction of parallel vector surfaces in 3D time-dependent fields and application to vortex core line tracking. In: IEEE Visualization, pp. 631–638 (2005). <https://doi.org/10.1109/VISUAL.2005.1532851>
30. Yan, L., Masood, T.B., Sridharamurthy, R., Rasheeda, F., Natarajan, V., Hotz, I., Wang, B.: Scalar field comparison with topological descriptors: Properties and applications for scientific visualization. Comput. Graph. Forum **40**(3), 599–633 (2021). <https://doi.org/10.1111/cgf.14331>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Daniel Klötzl** Daniel Klötzl is a PhD student at the Visualization Research Center, University of Stuttgart, where he is a member of the International Research Training Group (IRTG) DROBIT. His research interests include topological methods in data analysis and visualization. In particular, his studies are directed toward time-dependent topology for feature tracking and the concept of Jacobi sets. He received both MSc and BSc in mathematics at the University of Tübingen.



**Tim Krake** Tim Krake is a PhD student at the Visualization Research Center, University of Stuttgart, and at the Hochschule der Medien, Stuttgart, where he is a member of the joint graduate school Digital Media. He is currently working on the integration of mathematical techniques (mainly based on spectral decomposition methods) to the field of computer graphics and visualization. In this context, his particular interest is on dynamic mode decomposition (DMD). He received both MSc and BSc in mathematics at the University of Tübingen.



**Youjia Zhou** Youjia Zhou is a PhD student at the School of Computing and the Scientific Computing and Imaging (SCI) Institute, University of Utah. Her research focuses on developing visual analytics systems for large and complex data, in particular, networks, high-dimensional point clouds, and vector/tensor fields, using topological data analysis techniques.





**Ingrid Hotz** Ingrid Hotz is currently a Professor in Scientific Visualization at the Linköping University in Sweden. Her research interests lie in data analysis and scientific visualization, ranging from basic research questions to effective solutions to visualization problems in applications.



**Bei Wang** Bei Wang is an Assistant Professor at the School of Computing and a faculty member at the SCI Institute, University of Utah. She received her PhD in Computer Science from Duke University. Her research interests include topological data analysis, data visualization, computational topology, computational geometry, machine learning, and data mining.



**Daniel Weiskopf** Daniel Weiskopf is a Professor at the Visualization Research Center (VISUS) of the University of Stuttgart, Germany. He received his Dr.rer.nat. degree (similar to PhD) in Physics from the University of Tübingen, Germany, in 2001, and the Habilitation degree in Computer Science from the University of Stuttgart in 2005. His research interests include visualization, visual analytics, eye tracking, human-computer interaction, computer graphics, and special and general relativity. Photo by Max Kovalenko.