LLNL-CONF-468780

# Consistent Approximation of Local Flow Behavior for 2D Vector Fields using Edge Maps

S. Jadhav, H. Bhatia, P. T. Bremer, J. Levine, L. G. Nonato, V. Pascucci

February 2, 2011

TopoInVis
Zuerich, Switzerland
April 4, 2011 through April 6, 2011

## Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

# Consistent Approximation of Local Flow Behavior for 2D Vector Fields using Edge Maps

Shreeraj Jadhav, Harsh Bhatia, Peer-Timo Bremer, Joshua A. Levine, Luis Gustavo Nonato, and Valerio Pascucci

**Abstract** Vector fields, represented as vector values sampled on the vertices of a triangulation, are commonly used to model physical phenomena. To analyze and understand vector fields, practitioners use derived properties such as the paths of massless particles advected by the flow, called streamlines. However, currently available numerical methods for computing streamlines do not guarantee preservation of fundamental invariants such as the fact that streamlines cannot cross. The resulting inconsistencies can cause errors in the analysis, e.g. invalid topological skeletons, and thus lead to misinterpretations of the data. We propose an alternate representation for triangulated vector fields that exchanges vector values with an encoding of the transversal flow behavior of each triangle. We call this representation *edge maps*. This work focuses on the mathematical properties of edge maps; a companion paper discusses some of their applications [1]. Edge maps allow for a multi-resolution approximation of flow by merging adjacent streamlines into an interval based mapping. Consistency is enforced at any resolution if the merged sets maintain an order-preserving property. At the coarsest resolution, we define a notion of equivalency between edge maps, and show that there exist 23 equivalence classes describing all possible behaviors of piecewise linear flow within a triangle.

## 1 Introduction

Typically, vector fields are stored as a set of sample vectors at discrete locations. Vector values at unsampled points are defined by interpolating some subset of the known sample values. In this work, we consider two-dimensional domains represented as triangular meshes with samples at all vertices, and vector values on the interior of each triangle are computed by piecewise linear interpolation.

Many of the commonly used techniques for studying properties of the vector field require integration techniques that are prone to inconsistent results. Analysis based on such inconsistent results may lead to incorrect conclusions about the data. For example, vector field visualization techniques integrate the paths of massless particles (streamlines) in the flow [25] or advect a texture using line integral convolution (LIC) [2].

Shreeraj Jadhav, Harsh Bhatia, Joshua A. Levine and Valerio Pascucci
SCI Institute, University of Utah, USA, e-mail: {jadhav,hbhatia,jlevine,pascucci}@sci.utah.edu

Peer-Timo Bremer
Lawrence Livermore National Lab, USA, e-mail: bremer5@llnl.gov

Luis Gustavo Nonato
Universidade de São Paulo, Brazil, e-mail: gnonato@icmc.usp.br

Techniques like computation of the topological skeleton of a vector field [9, 10], require integrating separatrices, which are streamlines that asymptotically bound regions where the flow behaves differently [11]. Since these integrations may lead to compound numerical errors, the computed streamlines may intersect, violating some of their fundamental properties such as being pairwise disjoint. Detecting these computational artifacts to allow further analysis to proceed normally remains a significant challenge.

### Contributions

To address this challenge in situations where consistent analysis is required, we propose a new representation of flow through each triangle. This representation, called *edge maps*, approximates the flow by mapping pairs of boundary points that are on the same streamline to each other. Once computed, edge maps replace the numerical integration of streamlines with a lookup, which enforces that computed streamlines never cross. With implementation in mind, we can merge adjacent pairs of mapped points to provide a multi-resolution approximation of the edge map, enabling us to build more compact representations of the flow. At any resolution of merging, the edge maps still guarantee consistent streamlines. At the finest scale of resolution, streamlines remain accurate on triangle boundaries, while at the coarsest scale (the maximum possible amount of merging), the edge map represents one class of possible flow behavior. We enumerate these possibilities and show there are 23 equivalence classes that can exist for triangles in a piecewise linearly interpolated vector field. While the discussion within this work is primarily theoretical, an implementation of edge maps and their applications have been described in [1].

## 2 Related Work

One step towards producing consistent streamlines is to use higher accuracy integration techniques. Some recent results that improve on the traditional Runge-Kutta based techniques include the local exact method (LEM) of Kipfer et al. [13] that follows the lead of Nielson and Jung [16]. LEM solves an ODE for simplices on unstructured grids representing the position of the particle as a function of time, starting at a given position. While the solution is often more expensive than numeric integration, given an entry point of a particle to a triangle, LEM gives its exact path within the triangle. However, the exit point is calculated numerically as an intersection with the triangle edges which is prone to numeric errors. Hence, consistency of streamlines still cannot be guaranteed. Despite this, it is the most accurate technique available since it does not incur integration error. Although our work focuses on the mathematical properties of edge maps, we mention these results since a system using edge maps could first rely on such a computation for construction. Using LEM we can get a more accurate construction of edge maps.

Consistency becomes particularly desirable when computing structural properties of vector fields. Helman and Hesselink [10] compute a vector field's topological skeleton by segmenting the domain of the field using streamlines traced from each saddle of the field along its eigenvector directions. The nodes of the skeleton are critical points of the vector field and its arcs are the *separatrices* connecting them. Subsequently, the skeleton extraction has been extended to include periodic orbits [28]. Three dimensional variants of the topological skeleton have also been proposed [9, 12, 24, 27]. The reader should refer to [8, 14, 23] for more detailed surveys. However, it is well known that computing the topological skeleton can be numerically unstable due to errors inherent in the integration of separatrices and inconsistencies among neighboring triangles [4, 7, 16, 21].

A number of techniques have been proposed to extract the topological skeleton in a stable and efficient manner [3, 17, 22, 28]. Recent work of Reininghaus and Hotz [18] construct a combinatorial vector based on Forman's discrete Morse theory [6]. Using combinatorial fields allows the extraction of a consistent topological structure. However, combinatorial vector fields are limited by their high complexity, leading to later improvements to the algorithm [19]. While provably consistent, it is unclear how close the topological structure of the combinatorial field is to that of the original, piecewise linear, field. By comparison, this work proposes a multi-resolution technique that is both consistent and provides control over the level of approximation of the field.

# 3 Edge Maps

Let $\mathbf{V}\colon \mathscr{M} \to \mathbb{R}^2$ be a 2-dimensional vector field defined on a manifold $\mathscr{M}$. $\mathbf{V}$ is represented as a set of vector values sampled on the vertices of a triangulation of $\mathscr{M}$. Specifically, each vertex $p_i$ has the vector value $\mathbf{V}(p_i)$ associated with it. The vector values on the interior of each triangle $T$ with vertices $\{p_i, p_j, p_k\}$ in the triangulation are interpolated linearly using $\mathbf{V}(p_i)$, $\mathbf{V}(p_j)$, and $\mathbf{V}(p_k)$ (see Figure 1).

Given a vector field $\mathbf{V}$, we can define the *flow* $\phi(x,t)$ of $\mathbf{V}$. Treating $\mathbf{V}$ as a velocity field, the flow intuitively describes the parametric path that a massless particle travels according to the instantaneous velocity defined by $\mathbf{V}$. We define $\phi(x,t)$ as the solution of the differential equation:

$$\frac{d\phi(x,t)}{dt} = \mathbf{V}(x)$$

with the initial condition $\phi(x,0) = x_0$. Fixing a point $x$ and varying $t$, we call the collection of points produced by $\phi$ a *streamline*.

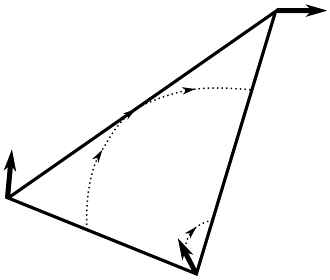


**Fig. 1** A triangle is represented as three vectors, which impart a flow through the interior.

## 3.1 Preliminaries

For the remainder of this work, we will assume the vector values for $\mathbf{V}$ on the interior of a triangle are defined by linearly interpolating the three vectors at the vertices of a triangle, $T$. We make three simplifying assumptions: (1) the vectors at all the vertices of the triangle are non-zero, (2) the vectors at any two vertices sharing an edge are not antiparallel, and (3) the vectors at two vertices on an edge $e$ are not both parallel to $e$. Any such configuration is unstable, and can be avoided by a slight perturbation. These small perturbations ensure a point to point mapping between the boundary of the triangle, and hence makes edge maps bijective. As we will see in the following sections, these assumptions will significantly reduce the complexity of many of the arguments as the locations of critical points become well defined.

Under the aforementioned assumptions, we can begin studying the properties of linearly varying vector fields. We first note the following two important properties that follow from our assumptions.

*Property 1.* Within $T$, $\mathbf{V}$ defines at most one critical point.

*Proof.* Critical points are defined as points $x$ where $\mathbf{V}(x) = 0$. Consider the component scalar fields $V_1$, $V_2$ where $\mathbf{V}(x) = (V_1(x), V_2(x))$. As the triangle is linearly varying, the graphs of $V_1$ and $V_2$ are planes lifted from the triangle. Critical points are the set of points where both these two planes intersect the constant plane of height zero. As we know from planar geometry, two planes either intersect in another plane or a line, and consequently these three planes intersect in a plane, a line, or a point.

If these three planes intersect in a plane, this defines an infinite number of zeros, which can only happen when assumption (1) is violated. If these planes intersect in a line, the only way that line passes through $T$ is when either (1) is violated (because two vertices are zero) or (2) is violated for at least two edges.

In all other cases, either a single critical point is defined (within the interior of $T$) or the field defines critical values outside of $T$. □

*Property 2.* On the line $\ell$ obtained by extending, from both sides, any edge $e$ of a triangle $T$, there exists at the most one point where the vector field is tangential to $\ell$.

*Proof.* Without loss of generality, rotate $\ell$ to be the $x$-axis and decompose $\mathbf{V}$ into its $x$ and $y$ components, i.e. $\mathbf{V}(\cdot) = (V_x(\cdot), V_y(\cdot))$. Following the linearity of the vector field $\mathbf{V}$, $V_y$ is also linear. By assumptions (1) and (3), $V_y = 0$ can only happen once, meaning that $\mathbf{V}$ is tangential to $\ell$ at only one point. We further note that on either side of this zero crossing, $V_y$ has a different sign, and hence the flow changes directions from flowing upwards to downwards, or vice versa. □

## 3.2 Defining Edge Maps

For a triangle $T$, let $\partial T$ and $\mathring{T}$ be the boundary and interior of $T$ respectively. We first examine the behavior of the flow at any point on $\partial T$. The flow behavior can be classified into four types (as illustrated in Figure 2) depending upon the location of that particle under flow $\phi(x,t)$ for positive and negative time t. Let $\varepsilon > 0$. Given a point $p \in \partial T$ such that $\phi(x,0) = p$, we define the following:

**Definition 1 (Inflow Point).** If for all $t \in ]0,\varepsilon]$, $\phi(x,t) \in \mathring{T}$ and $\phi(x,-t) \notin T$, then $p$ is an *inflow point*.

**Definition 2 (Outflow Point).** If for all $t \in ]0,\varepsilon]$, $\phi(x,t) \notin T$ and $\phi(x,-t) \in \mathring{T}$, then $p$ is an *outflow point*.

**Definition 3 (External Transition Point (ETP)).** If for all $t \in ]0,\varepsilon]$, $\phi(x,\pm t) \notin T$, then $p$ is an *external transition point (ETP)*.

**Definition 4 (Internal Transition Point (ITP)).** If for all $t \in ]0,\varepsilon]$, $\phi(x,\pm t) \in \mathring{T}$, then $p$ is an *internal transition point (ITP)*.

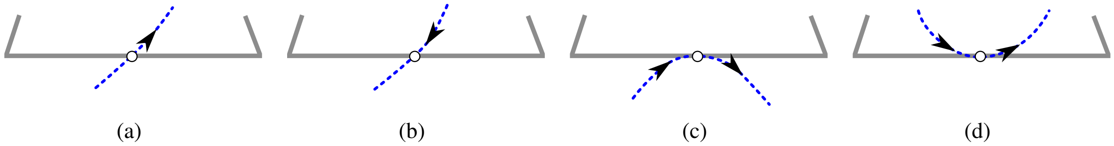


**Fig. 2** Based on the flow at a point on the boundary of a triangle, the point can be classified as one of the following: (a) Inflow Point, (b) Outflow Point, (c) External Transition Point (ETP), and (d) Internal Transition Point (ITP).

The concept of transition points is similar to that of boundary switch points mentioned in [5, 15, 26]. The above four definitions account for all possible behaviors of flow as it touches $\partial T$, since our assumptions only allow the possibility where flow travels along the edge instantaneously. Our ultimate goal is to model the flow through $\mathring{T}$. To start, we first pair points as they travel along the same streamline.

**Definition 5 (Origin-Destination Pair (o-d pair)).** Let $a,b \in \mathbb{R}$ and $a \leq b$ such that $\phi(x,[a,b]) \subset T$ where $p = \phi(x,a)$ and $q = \phi(x,b)$. We call $(p,q)$ an origin-destination pair if the time interval $[a,b]$ is maximal where $p,q \in \partial T$ and $\phi(x,]a,b[) \subset \mathring{T}$. We call $p$ an *origin point* and $q$ a *destination point*.

Here, maximal interval means that the time range $[a,b]$ produces the largest possible streamline contained within $\mathring{T}$, bounded by two points $p$ and $q$ on $\partial T$. Inflow, outflow, and transition points all play different roles in o-d pairs. The simplest case is for inflow and outflow points. Since the streamline of an inflow point $p$ flows to the $\mathring{T}$, $p$ will be paired up with the point $q$ (either an outflow point or ITP) where this streamline first touches the boundary after $p$.

An ITP flows to $\mathring{T}$ in both positive and negative time, thus creating at most two such maximal intervals for which a streamline is completely contained in $\mathring{T}$. Hence, an ITP may participate in two o-d pairs. In one pair it is an origin point while in the other it is a destination point. If the streamline is an orbit touching $\partial T$ only at the ITP, we pair the ITP with itself, resulting in a single o-d pair. In this case, the streamline of the orbit has many time intervals $[a,b]$ where $\phi(x,]a,b[) \subset \mathring{T}$, all of which are of equal length. On the contrary, an ETP always forms an o-d pair with itself, since the streamline does not flow to $\mathring{T}$. Hence its maximal time range will be when $a = b$.

Thus, all origin points are either inflow points or transition points; and all destination points are either outflow points or transition points. Note that if $x$ is chosen such that $\phi(x,t)$ is an orbit contained entirely in $\mathring{T}$, then such a streamline will never converge to $\partial T$ and hence any $p$ and $q$ on it will not form an o-d pair. Moreover, there are some points that do not converge to $\partial T$, because of the presence of a critical point in $T$. Such points can be classified as *unmapped inflow points* (associated with a sink), *unmapped outflow points* (associated with a source), or *sepx points* (intersections of the saddle separatrices with $\partial T$), and they are not included in the definition of o-d pairs.

Since we can determine the existence of an o-d pair for every point on $\partial T$ we can define a mapping to describe the transversal behavior of flow through points on $\partial T$. Let $P \subset \partial T$ be the set of all origin points, and $Q \subset \partial T$ be the set of all destination points.

**Definition 6 (Edge Map).** An *edge map* of $T$ is defined as a map $\xi : P \to Q$, such that $\xi(p) = q$, if $(p,q)$ is an o-d pair.

We will also call $q$ the *image* of $p$ under $\xi$.

We claim that $\xi$ is a bijection between $P$ and $Q$, and thus its inverse is also well defined. We call $\xi$ as the *forward edge map*, since it maps an origin point to its destination point under forward flow. Its inverse $\xi^{-1}$ maps a destination point to its origin point, and hence is called the *backward edge map*.

Using the forward and backward edge maps, integration of streamlines can be replaced with a map lookup that traverses the triangle. Using only this lookup, a streamline travelling through a triangulated vector field is approximated as a sequence of points on the boundaries of triangles through the field. This approximation discards the flow behavior of the interior of triangles, but as we shall see, maintains enough information to maintain consistency of streamlines.

### 3.3 Approximating Edge Maps

In this paper, we intend to define a feasible representation for storing and using an *approximated edge map* $\xi^*$ as a data structure to represent $\xi$. We achieve this by grouping the point to point exact mapping $\xi$ into connected intervals which preserve its ordering, forming what we call the *links*.

**Definition 7 (Link).** Let $O$ be a connected subset of $P$, and $D$ be a connected subset of $Q$, where $D = \xi(O)$. Let $\partial O$ be the end points (boundary) of $O$. We call $\zeta : O \to D$ a *link* if $\zeta$ is continuous, order-preserving, and $\zeta(x) = \xi(x)$ for each $x \in \partial O$. We call the set $O$ an *origin interval* and the set $D$ a *destination interval*.

A link $\zeta$ allows for a second level of approximation of $\xi$ on some subsets of the domain of $\xi$. Let $\mathring{O}$ be the interior of $O$. For each point $p \in \mathring{O}$, we allow the mapped points $\zeta(p)$ to shift from its original $\xi(p)$, except at the boundaries of $O$.

Although, $\zeta$ is an approximation, we at least require that it preserves the ordering of streamlines. The property of *order-preservation* handles this notion by disallowing links which have streamlines that cross. To enforce order-preservation, first note that since $O$ is a connected subset of $\partial T$, it can be parameterized as a single interval. Similarly, since $\zeta$ is continuous, we will only build links where $\xi(O)$ is a single interval, again parameterizable. As long as we preserve the ordering within this parameterization, $\zeta$ is a valid link. Thus, the property of order-preservation leads to the following theorem.

**Theorem 1.** *Order-preserving links always produce consistent (pairwise disjoint) streamlines.*

Since each $\zeta$ may only be defined on a subset of $P$, to completely approximate $\xi$ we need a set of links. This motivates the following definition.

**Definition 8 (Approximated Edge Map).** An *approximated edge map* $\xi^*$ of $\xi$ is a collection of $n$ links $\zeta_i : O_i \rightarrow D_i$, such that the $\{O_i\}$ and $\{D_i\}$ form partitioning of $P$ and $Q$ respectively.

We can see that an edge map can be subdivided into links in many ways to form such a partitioning. Since the boundaries of each link are "snapped" to require $\zeta(x) = \xi(x)$, using more links enforces more accuracy. Once the partitioning of $P$ is defined, one way of approximating $\xi$ as $\xi^*$ is creating links $\zeta_i$ which are linearly mapped between $O_i$ and $D_i$, as proposed in [1]. Such a $\zeta$ satisfies the properties of order-preservation, and provides a simpler map lookup computation.

Earlier, we pointed out that certain points on $\partial T$ can be classified as unmapped inflow, unmapped outflow or sepx points, and do not participate in any o-d pair. It follows from the definition of the link that these points cannot be included in either of the origin or destination intervals, and hence do not get included in any link. Connected subsets of such points are called *unmapped intervals*.

## 3.4 Base Edge Maps

Links can be created by *merging* the origin and destination sets of adjacent o-d pairs such that the origin and destination intervals remain connected sets and the orientation of the link is preserved. It turns out that this merging process can be expanded only so much, until specific points where the continuity of the flow breaks. At this maximal amount of merging, we have a minimal set of links in the approximated edge map.

**Definition 9 (Base Edge Map).** A *base edge map* of $T$ is an approximated edge map $\xi^*$ such that the number of links is minimal. Links of a base map are called *base links*.

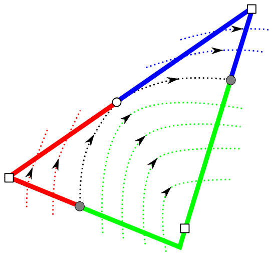


**Fig. 3** The *base edge map* for the triangle in Fig. 1 has 3 links (shown in different colors). It is generated by splitting the boundary at ITP's (white circles), their images (grey dots), and ETP's (white squares).

This merging of o-d pairs into a single link cannot be done across transition points, sepx points, and ITP images. This follows since orientation of a link cannot be preserved across transition points while sepx points and ITP images break the continuity of the map. Thus, the intervals in a base edge map are bounded by the transition points, sepx points, and the image points of ITPs. A base edge map can be readily constructed by identifying these points and splitting the boundary of a triangle at these points into intervals. The intervals can be paired up into links using connectivity information. Figure 3 shows the creation of base edge maps for a regular triangle (with no critical point in the interior).

# 4 Classification of Base Edge Maps

There are many ways to approximate an edge map $\xi$ as a set of links. Using the concept of a base edge map, many different edge maps can be reduced to the same base edge map. Thus, a base edge map can be thought of as representative of an infinite number of edge maps, each representing the same flow behavior. This motivates a study of all possible base edge maps, which leads to a notion of *equivalence classes* of the edge maps that represent all possible types of linearly varying flow. In what follows, all proofs are included in the appendix.

## *4.1 Equivalence of Maps*

While the notion of reducing any edge map to its base edge map is well defined, we need to define why two base edge maps are considered equivalent in order to declare them as belonging to the same class. Before discussing equivalence, we shall enumerate all the features of a map that we would like to include when comparing two maps.

- *Pairing* of intervals $(O_i, D_i)$ as links.
- *Orientation* of links $(O_i \to D_i)$.
- *Cyclic order* of all intervals (including unmapped) on $\partial T$.

We define a way to construct a mixed graph from any edge map $\xi$ approximated by $\xi^*$ and the set of links $\{\zeta_1, \zeta_2, \ldots, \zeta_n\}$. Our goal is to capture all the above features into the graph. A **mixed graph** $\mathscr{G}(V, E, A)$ can be constructed such that:

- Nodes: $V = \{I_k\}$ where $I_k$ is either an origin interval $O_i$, a destination interval $D_i$, an unmapped interval $U_j$, or a sink or source $S_p$ in $\mathring{T}$.
- Undirected Edges: $E = \{\{I_1, I_2\}\}$ where $I_1$ and $I_2$ are adjacent on $\partial T$.
- Directed Edges: $A = \{(I_1, I_2)\}$ where either $\zeta_i : I_1 \to I_2$ (corresponding to a link); $I_1 = U_j$ and $I_2 = S_p$ (an unmapped inflow interval); or $I_1 = S_p$ and $I_2 = U_j$ (an unmapped outflow interval).

An example of such a graph is shown in Figure 4. Though unmapped intervals do not participate in the edge map, they have a structural bearing on the map and consequently on the mixed graph. They separate links that would otherwise appear contiguous and could be potentially merged. In the case of a saddle, while the sepx points are also unmapped, they need not have an explicit place in the graph as their existence can be inferred from its neighboring intervals (and consequently the existence of a saddle in $\mathring{T}$).

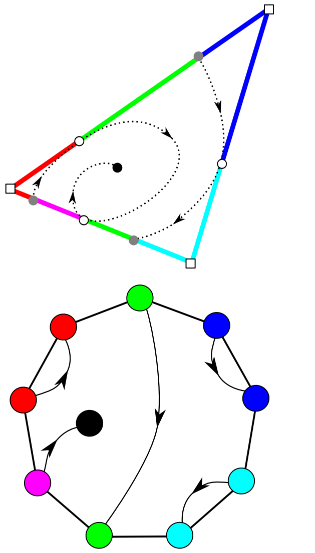


**Fig. 4** Construction of mixed graph from a map. Points of the magenta interval do not form o-d pairs since they flow directly to $S_p$.

**Definition 10 (Equivalence).** Two edge maps are considered equivalent if the mixed graphs of their base edge maps are *isomorphic*.

Under this definition of equivalence, we present a group theoretic approach for counting all possible equivalence classes of base edge maps. We use the Groups, Algorithms, and Programming (GAP) system [20] to do the following computation. There are three important steps: *generation* of initial sample space; *restricting* this space to those possible in linearly varying flow; and identifying *equivalences*.

### 4.2 Initial Sample Space

Our analysis for all possible equivalence classes first considers all possibilities for links and unmapped intervals. We start by disallowing an interval from spanning beyond an edge. This simplifying assumption helps enumerate the sample space, and later we shall remove these splits caused by vertices, allowing links to span across them. From the definition of base links and given the property that all links of a base edge map are bounded by transition points, sepx points, and ITP images, any base edge map has a bounded number of links and unmapped intervals.

Let $a,b,c$ symbolically represent the edges of $T$ and $d$ be either the sink or source in $\mathring{T}$, if it exists. For an interval $I$, its *location*, $loc(I)$, can be any of the four values in $L = \{a,b,c,d\}$. In the following lemma we show that in $\mathscr{G}$, no two directed edges having their origin nodes in the same location can have their destination nodes in the same location. Said more formally:

**Lemma 1 (Edge Level Links).** *Let $\mathscr{G}(V,E,A)$ be a mixed graph of an edge map where no node $I_i$ spans more than one edge of $\partial T$ and let $loc(I_i)$ be its location. Consider two directed edges $(I_1,I_2),(I_3,I_4) \in A$. If $loc(I_1) = loc(I_3)$ and $loc(I_2) = loc(I_4)$, then $(I_1,I_2),(I_3,I_4)$ can be merged.*

From Lemma 1 we see that there are a maximum of 13 directed edges in the graph of any edge map with edge level links, as each pair in $S = L \times L$ can only appear once and the pair $(d,d)$ cannot appear because $d$ is either a source or a sink, but not both. Moreover, no pair $(d,\cdot)$ can co-exist with a pair $(\cdot,d)$, again because $d$ is either a source or a sink. Thus, each configuration for any set of directed edges $A$ can be represented as certain elements of $\mathscr{P}(S)$, the power set of $S$. Consequently, the number of possible configurations for directed edges is $2^{13}$.

Under rotations of $T$, certain directed edges are equivalent. For example, $\{(a,b),(b,c)\}$ is equivalent to $\{(b,c),(c,a)\}$. Similarly, $\{(b,b),(c,c)\}$ is equivalent to $\{(c,c),(a,a)\}$ and $\{(a,a),(b,b)\}$. We next eliminate such equivalences under rotation by using the action of a permutation group that imparts rotations $(a \to b \to c \to a)$ and $(a \to c \to b \to a)$.

These permutations allow us to enumerate the possibilities for the set $A$; however, they still lack any cyclic ordering of nodes to form the undirected edges $E$ in $\mathscr{G}$. We generate all the mixed graphs under the constraint that there is a maximum of one TP on the interior of an edge (a consequence of Property 2) by grouping all intervals sharing the same edge label. We do this by first generating permutations of all origin/unmapped inflow nodes and destination/unmapped outflow nodes separately on every edge. All possible orderings of nodes on that edge are then the Cartesian product of these two sets. This ultimately constructs the set of all possible mixed graphs (our initial sample space) which numbers more than 100 million.

### 4.3 Restricting Cases

Our initial enumeration of all graphs is quite large, but contains many invalid graphs. Using the following lemmas based on the linear nature of $\mathbf{V}$, we filter this initial set. In particular, the following must hold:

**Lemma 2 (ITP-vertex).** *An ITP cannot exist on a vertex of $T$.*

Next, Lemma 3 states an important fact that can be used to qualify a critical point based on the existence of an ETP on the interior of an edge of the triangle.

**Lemma 3 (ETP-Saddle).** *Let $q \in \partial T - \{p_i, p_j, p_k\}$ be an ETP. If there exists a critical point $S_p \in \mathring{T}$, then $S_p$ is a saddle.*

Based on these lemmas, we enforce the following rules on all mixed graphs:

1. Directed edges of $\mathscr{G}$ cannot intersect. (Since streamlines never intersecting in piecewise linear flow.)
2. ITP cannot exist on vertices. (Lemma 2)
3. ETPs cannot exist on edges when there is source or sink (Lemma 3 and Property 1).

We can label some undirected edges of a mixed graph as either ITPs or ETPs using the combinatorial structure of the graph. For an undirected edge $\{I_1, I_2\}$ in $\mathscr{G}$, if $I_1$ and $I_2$ switch between inflow to outflow, then $\{I_1, I_2\}$ is a *combinatorial TP*. Moreover, if $I_1$ and $I_2$ do not have a directed edge between them in $A$, then the TP is an *combinatorial ITP*. Images of an ITP are directly implied by its directed edges. Similarly, if $I_1$ and $I_2$ have a directed edge between them, then the TP is a *combinatorial ETP*. We remark that in a very few number of cases, combinatorial ETPs actually correspond to an interior flow that wraps to itself, creating an orbit in the triangle. These cases are not distinguished by the flow maps, but can only appear if there is no source or sink within $\mathring{T}$.

Though the above analysis does not label all the undirected edges, these are sufficient to enforce the above mentioned rules to restrict the sample space of mixed graphs to under one thousand. After labeling combinatorial ITPs and ETPs, the second and third rule can be enforced directly. We enforce the first rule using a stack method of detecting intersections between links. In a valid map, any link that has its intervals on $\partial T$ divides the triangle into two halves such that no other link has intervals in both the halves, thus forming a last in, first out sequence on $\partial T$ equivalent to pairing parentheses in a mathematical expression.

## 4.4 Identifying Equivalences

For the final equivalence computation in GAP, we merge contiguous directed edges so that all mixed graphs become base edge maps. We next use the dihedral group for all graph symmetries as well as the permutation group for all labelings of nodes. The dihedral group is useful for comparing the mixed graphs under all rigid transformations. The permutation group also includes inversion of the flow direction of directed edges to compare similarities in base edge maps under global inversion of flow. Using GAP, we enumerate all graphs that fall in the same orbit under its actions, each set of which represents an equivalence class for the graphs. This process produces 43 map classes as an output.

Since our mixed graph definition does not explicitly encode saddles or orbits, we then manually invalidate more cases based on some additional lemmas, many of which would be challenging to encode within GAP, but are simple to remove by inspection. We rely on two properties for the behavior of ITPs:

**Lemma 4 (ITP-Saddle).** *If there is a saddle in $\mathring{T}$, then an ITP cannot be present on $\partial T$.*

Additionally, Lemma 5 limits the location of a critical point $S_p$ within the interior of the triangle when an ITP exists. In particular, $S_p$ must be enclosed by the streamline extending from an ITP.

**Lemma 5 (ITP-CP Enclosure).** *Let $x \in \partial T$ be an ITP whose images $x_b, x_f \in \partial T$ and $S_p$ be a critical point in $\mathring{T}$. $S_p$ is an element of the area bounded by the streamline $\phi(x,t)$ and $\partial T$ where $t \in [a,b]$ s.t. $\phi(x,a) = x_b$ and $\phi(x,b) = x_f$.*

Thus, the following rules are enforced on the remaining cases to get the final equivalence classes:

1. The graphs should not violate Lemma 4.
2. The graphs should not violate Lemma 5.
3. There can be a maximum of one critical point. If there is a sink or a source in the map, there cannot a sepx point (Property 1, since a sepx point means there must additionally be a saddle).

4. There have to be exactly four sepx points or none at all (By definition of a linear saddle).

To enforce the above rules, we need to identify *combinatorial sepx* points in the graphs. Since, we have merged all contiguous directed edges, the graphs represent the base edge maps. Thus, any unlabeled undirected edges have to correspond to sepx points, since all ETP, ITP and ITP images have already been labeled. Knowing the labeling of each undirected edge allows us to hand invalidate 20 of the 43 cases produced by GAP. 23 classes persist after the invalidations, visualized in Figure 5. We have generated vector values for triangles to realize each of these cases as well, confirming the following result:

**Theorem 2.** *Under Definition 10 for equivalence, there are 23 equivalence classes of edge maps for triangles in a linearly varying vector field.*

## 5 Discussion

Edge maps are a natural way to encode one facet of the information a triangulated vector field presents. The study of the equivalence classes of edge maps helps us understand different structures of flow possible on a triangle. Every class represents a large number of possible realizations of the flow. While some of the edge map classes represent a more generic flow in terms of a point-to-point map, some of them may reflect degeneracies in the flow. For example, class 4 (first column, third row in Figure 5) represents the flow imparted by an orbit in a triangle. While it is widely known that such a field is unstable under small perturbations, its corresponding edge map class also represents this instability.

The main benefit of edge maps is that they explicitly store the origins and destinations of flow through individual triangles. This gives a direct control on how the streamlines are computed, enabling us to enforce consistency. While they store how streamlines travel across triangles, they discard any notion of what is happening within the triangle. However, they can still preserve the global structure of the flow in the sense that starting at a single point and traversing through a series of maps will give an accurate representation of the destination even if the path it takes is only approximated. If a higher resolution of the shape of a streamline is desired, it may be achieved by using numerical integration as a progressive computation. This is generally not required if the mesh is sufficiently fine.

In their current form, the maps drop all notion of time in their construction. One could model a “unit” of time as a jump across a single triangle. It might be more meaningful to add how long it takes to get from origin to destination across the triangle. The time it takes to travel through a link could then be added into the edge map. This modification would allow a more accurate representation of flow.

We believe that edge maps have the potential to complement existing techniques for processing vector fields. Since they maintain consistency while providing a level of control over accuracy, one could conceivably design algorithms for visualization and topological analysis that can leverage these properties. Some example applications we have already considered include error analysis of streamline integration techniques [1].

The structure of an edge map depends on the flow as well as the geometry of the underlying mesh. The current study of edge maps is based on linear interpolation. As future work, there are two apparent ways to generalize this study. One is by extending it to a generalized interpolation scheme; and another way is to extend it to higher dimensional simplices and other types of meshes.

**Acknowledgements** This work is supported in part by the National Science Foundation awards IIS-1045032, OCI-0904631, OCI-0906379 and CCF-0702817. This work was also performed under the auspices of the U.S. Department of Energy by the University of Utah under contracts DE-SC0001922, DE-AC52-07NA27344, and DE-FC02-06ER25781, and Lawrence Livermore National Laboratory (LLNL) under contract DE-AC52-07NA27344. Attila Gyulassy and Philippe P. Pebay provided many useful comments and discussions. LLNL-CONF-468780.

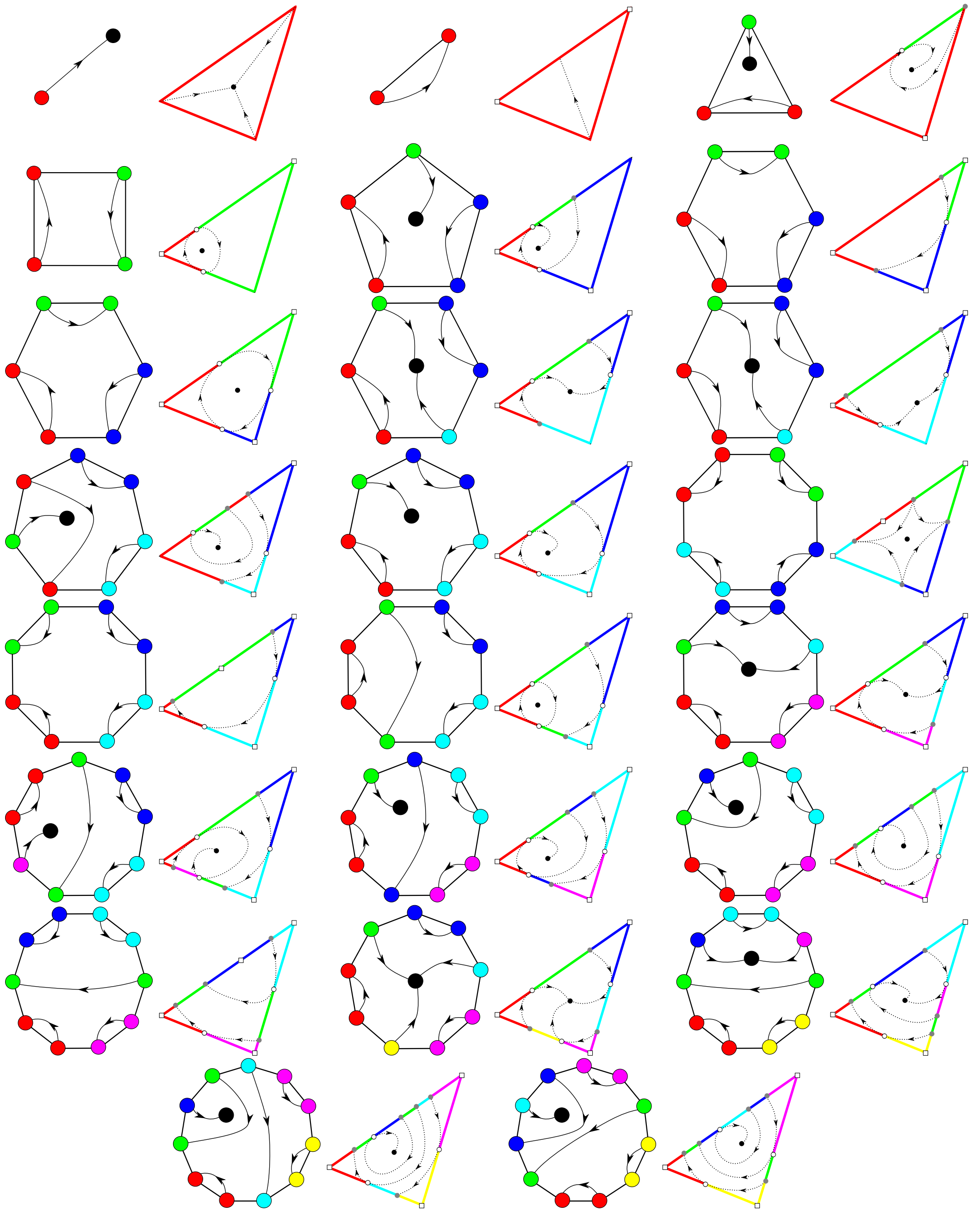


**Fig. 5** The 23 equivalent classes of mixed graphs, along with one possible rendition of edge map for piecewise linear flow for each class, in the order of increasing number of links.

## Appendix

### Proofs from Section 4.2

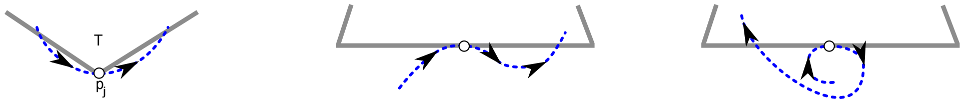
*Proof (Lemma 1).* In all the cases that follow, if $I_1$ and $I_3$ lie on an edge of $T$ then we assume that there exists a subset $I_5 \in \partial T$ such that $I_5$ lies between $I_1$ and $I_3$. Similarly, there exists $I_6 \in \partial T$ between $I_2$ and $I_4$. In such a scenario, $I_5$ is a set of inflow points and $I_6$ is a set of outflow points, otherwise there will be more than one TP on the same edge. We know from Property 2 that this is not possible. Let $a, b, c$ be the edges of $T$ and $d$ the critical point in $\mathring{T}$. We enumerate all possibilities of locations for the four intervals:

1. $loc(I_1)$ and $loc(I_3)$ is any edge, $e$; and $loc(I_2)$ and $loc(I_4)$ is $d$.
2. $loc(I_1)$ and $loc(I_3)$ is $d$; and $loc(I_2)$ and $loc(I_4)$ is any edge, $e$.
   (This is symmetric to case 1 in reverse flow.)
3. $loc(I_1)$ and $loc(I_3)$ is any edge, $e_1$; and $loc(I_2)$ and $loc(I_4)$ is another edge, $e_2$.
4. $loc(I_1)$ and $loc(I_3)$ is any edge, $e$; and $loc(I_2)$ and $loc(I_4)$ is the same edge $e$.
5. $loc(I_1)$, $loc(I_3)$, $loc(I_2)$, and $loc(I_4)$ is $d$.
   (This case is not possible since the one critical point is either source or sink but not both.)

From the discussion in Section 3.4 we know that we can merge adjacent o-d pairs except across transition points, sepx points, and ITP images. For all the above cases, streamlines on $I_5$ and $I_6$ are bounded by $(I_1, I_2)$ and $(I_3, I_4)$ irrespective of their locations. Therefore, since there cannot be a TP on $I_5$ or $I_6$, we can also rule out ITP images. Similarly, sepx points are impossible since all four sepx points will have to lie on $I_5$ and $I_6$ which would imply a TP on them. Thus, under the given constraint, $(I_1, I_2)$ and $(I_3, I_4)$ can be merged. □

### Proofs from Section 4.3

*Proof (Lemma 2).* Let vertex $p_j$ of $T$ be an ITP. From the definition of ITP, the curve $\phi(x, \pm t) \in \mathring{T}$ for all $t \in ]0, \varepsilon]$, such that $\phi(x, 0) = p_j$. The streamline $\phi(x, \pm t)$ is a $C^1$ continuous curve, while the two edges of $T$ meeting at $p_j$ define a corner. By $C^1$ continuity, $\phi(x, \pm t)$ has a well defined tangent everywhere, including at $p_j$, but since the interior angle of $T$ at $p_j$ is less than $\pi$, this is not possible for $\phi(x, \pm t)$ being contained in $\mathring{T}$. Thus, the streamline $\phi(x, \pm t)$ must exit $T$ to maintain its continuity (as shown in Figure 6(a)). This contradicts the definition of an ITP, implying an ITP is not possible on a vertex. □



(a) (b) (c)

**Fig. 6** (a) In Lemma 2, a streamline passing through a vertex $(p_j)$ of triangle $T$ must exit and re-enter $T$. (b) The first case in Lemma 3, multiple switches in flow across line of the edge. (c) The second case in Lemma 3, more than one critical point.

*Proof (Lemma 3).* Let an ETP $x_e$ exist on an edge $a$, of $T$. Thus, $\phi(x_e, \pm t) \notin T$ for $t \in ]0, \varepsilon]$. Let $S_p \in \mathring{T}$ be the critical point. From Property 1 we know that there is exactly one interior critical point in the vector field. Therefore, if $S_p$ is a sink, all streamlines eventually flow to it; if $S_p$ is a source, all streamlines emerge from it; and if $S_p$ is an orbit, all streamlines are closed loops. Thus, if $S_p$ is a sink or a source, $\phi(x_e, t)$ will have to flow to, or emerge from it. This cannot happen without crossing the line $\ell$ of edge $a$. There are two ways in which $\phi(x_e, t)$ can cross $\ell$. First, it may make multiple switches of direction (Figure 6(b)) over $\ell$, violating Property 2. Second, one of the images (backward or forward) gets trapped in the enclosure of $\phi(x_e, t)$ and flows to a critical point inside (Figure 6(c)), violating the fact that there is exactly one defined critical point. If $S_p$ is an orbit, the closed loop has to enclose $S_p$ else it would define another orbit. Thus again $\phi(x_e, t)$ is forced to cross $\ell$, violating Property 2. Thus $S_p$ has to be a saddle. □

### Proofs from Section 4.4

*Proof (Lemma 4).* Let an ITP $x_i$ exist on an edge $a$ of $T$, Thus, $\phi(x_i, \pm t) \in \mathring{T}$ for $t \in ]0, \varepsilon]$. Let $S_p \in \mathring{T}$ be the critical point. Assume, to build a contradiction, that $S_p$ is a saddle. This assumption implies that all streamlines including $\phi(x_i,t)$ are parallel to the separatrices of the saddle as $t \to \pm\infty$. We show that $\phi(x_i,t)$ crosses the line of an edge of $T$ multiple times to stay parallel to the separatrices in those limits of $t$, contradicting Property 2. This follows since at least one separatrix intersects the line of every edge of $T$ forcing multiple intersections of $\phi(x_i,t)$ with the line of the edge to which $x_i$ belongs. Hence, $S_p$ cannot be a saddle. The cases for saddle sectors are shown in Figures 7(a) and 7(b). □

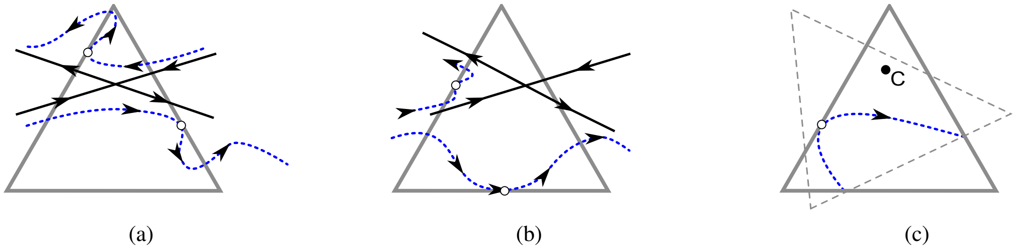


**Fig. 7** (a) and (b) Lemma 4: Cases of saddle sectors intersecting $\partial T$ invalidating the existence of an ITP on $\partial T$. (c) Construction of a new triangle as in Lemma 5 to place both the forward and backward images of an ITP are on the same edge.

*Proof (Lemma 5).* Let an ITP $x_i$ exist on an edge $a$, of $T$ and $S_p$ be a critical point in $\mathring{T}$. Assume, to build a contradiction, that $S_p$ exists outside the enclosure of $\phi(x_i,t)$ in $\mathring{T}$. We show that this is not possible. Since $x_i$ is an ITP, $S_p$ is not a saddle according to Lemma 4. Let $a, b, c$ be the three edges of $T$.Now we enumerate all possible combinations of edges on which $x_i$, $x_f$, and $x_b$ can lie:

1. edge$(x_i)$ = edge$(x_b)$ = edge$(x_f)$.
The only way to realize this case is an orbit such that $x_i = x_f = x_b$. Such an orbit immediately encloses the critical point.
2. edge$(x_i)$ = edge$(x_b)$ or edge$(x_i)$ = edge$(x_f)$.
These two cases are symmetrical under reversal of flow. If either $x_f$ or $x_b$ lie on the same edge as $x_i$, by Property 2, the other image gets enclosed into the $\phi(x_i,t)$ and has to flow to a critical point.
3. edge$(x_i) \neq$ edge$(x_b)$ = edge$(x_f)$.
In this case, both $x_b$ and $x_f$ lie on the same edge indicating a switch from inflow to outflow. This implies that there exists at least one TP between them including the two points. If one of $x_f$ and $x_b$ is a TP, it has to be an ITP since part of its streamline is already in $\mathring{T}$. Due to Property 2, the other image of this ITP has to flow into a critical point in the enclosure of $\phi(x_i,t)$. If a point in the interior is a TP, this TP cannot be an ETP since we have already established that $S_p$ is not a saddle and an ETP would violate Lemma 3. Now one of the images of this ITP can flow back to the same edge, but the other has to flow to $S_p$. Thus $S_p$ is in the enclosure of $\phi(x_i,t)$.
4. edge$(x_i) \neq$ edge$(x_b) \neq$ edge$(x_f)$.
In this case, we can always construct a new triangle by connecting $x_f$ and $x_b$ as an edge and arbitrarily choosing a third point such that $S_p$ still lies in the interior of the new triangle as shown in figure 7(c). An argument identical to case 3 can now be applied. □

## References

1. H. Bhatia, S. Jadhav, P.-T. Bremer, G. Chen, J. A. Levine, L. G. Nonato, and V. Pascucci. Edge maps: Representing flow with bounded error. In *IEEE Pacific Visualization Symposium*, 2011. Accepted.
2. B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *SIGGRAPH*, pages 263–270, 1993.
3. G. Chen, K. Mischaikow, R. S. Laramee, P. Pilarczyk, and E. Zhang. Vector field editing and periodic orbit extraction using morse decomposition. *IEEE Trans. Vis. Comput. Graph.*, 13(4):769–785, 2007.
4. G. Chen, K. Mischaikow, R. S. Laramee, and E. Zhang. Efficient morse decompositions of vector fields. *IEEE Trans. Vis. Comput. Graph.*, 14(4):848–862, 2008.
5. W. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In *Proc. of IEEE Visualization '99*, pages 349–354, 1999.
6. R. Forman. A user's guide to discrete morse theory. In *Proc. of the 2001 Internat. Conf. on Formal Power Series and Algebraic Combinatorics, A special volume of Advances in Applied Mathematics*, page 48, 2001.
7. C. Garth, H. Krishnan, X. Tricoche, T. Tricoche, and K. I. Joy. Generation of accurate integral surfaces in time-dependent vector fields. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1404–1411, 2008.
8. C. Garth and X. Tricoche. Topology- and feature-based flow visualization: Methods and applications. In *SIAM Conference on Geometric Design and Computing*, 2005.
9. A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *IEEE Visualization*, pages 33–41, 1991.
10. J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *IEEE Computer*, 22(8):27–36, 1989.
11. M. W. Hirsch, S. Smale, and R. L. Devaney. *Differential Equations, Dynamical Systems, and An Introduction To Chaos*. Elsevier Academic Press, 2$^{\text{nd}}$ edition, 2004.
12. K. M. Janine, J. Bennett, G. Scheuermann, B. Hamann, and K. I. Joy. Topological segmentation in three-dimensional vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 10:198–205, 2004.
13. P. Kipfer, F. Reck, and G. Greiner. G.: Local exact particle tracing on unstructured grids. *Computer Graphics Forum*, 22:133–142, 2003.
14. R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology Based Flow Visualization: The State of the Art. In *Topology-Based Methods in Visualization (Proceedings of Topo-in-Vis 2005)*, Mathematics and Visualization, pages 1–19. Springer, 2007.
15. S. K. Lodha, J. C. Renteria, and K. M. Roskin. Topology preserving compression of 2d vector fields. In *Proceedings of the conference on Visualization '00*, VIS '00, pages 343–350, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.
16. G. M. Nielson and I.-H. Jung. Tools for computing tangent curves for linearly varying vector fields over tetrahedral domains. *IEEE Trans. Vis. Comput. Graph.*, 5(4):360–372, 1999.
17. K. Polthier and E. Preuß. Identifying vector fields singularities using a discrete hodge decomposition. In *Mathematical Visualization III*, pages 112–134. Ed: H.C. Hege, K. Polthier, 2003.
18. J. Reininghaus and I. Hotz. Combinatorial 2d vector field topology extraction and simplification. In *Topological Methods in Data Analysis and Visualization. Theory, Algorithms, and Applications. (TopoInVis'09)*, 2009.
19. J. Reininghaus, C. Löwen, and I. Hotz. Fast combinatorial vector field topology. in press, 2010.
20. M. Schönert et al. *GAP – Groups, Algorithms, and Programming*. Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, fifth edition, 1995.
21. G. Scheuermann, T. Bobach, H. Hagen, K. Mahrous, B. Hamann, K. I. Joy, and W. Kollmann. A tetrahedra-based stream surface algorithm. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 151–158, Washington, DC, USA, 2001. IEEE Computer Society.
22. G. Scheuermann, H. Krüger, M. Menzel, and A. P. Rockwood. Visualizing nonlinear vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109–116, 1998.
23. G. Scheuermann and X. Tricoche. Topological methods in flow visualization. In C. Hansen and C. Johnson, editors, *In Visualization Handbook*, pages 341–356. Elsevier, 2004.
24. H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. On the applicability of topological methods for complex flow data. In H. Hauser, H. Hagen, and H. Theisel, editors, *Topology-based Methods in Visualization*, Mathematics and Visualization, pages 105–120. Springer, 2007.
25. G. Turk and D. Banks. Image-guided streamline placement. In *SIGGRAPH*, pages 453–460, 1996.
26. T. Weinkauf, H. Theisel, H. c. Hege, and H. p. Seidel. Boundary switch connectors for topological visualization of complex 3d vector fields. In *In Proc. VisSym 04*, pages 183–192, 2004.
27. T. Weinkauf, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel. Extracting higher order critical points and topological simplification of 3d vector fields. In *Proc. of IEEE Visualization '05*, pages 559–566, 2005.
28. T. Wischgoll and G. Scheuermann. Detection and Visualization of Closed Streamlines in Planar Fields. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165–172, 2001.