Morse Set Classification and Hierarchical Refinement using Conley Index

Guoning Chen, Qingqing Deng, Andrzej Szymczak, Robert S. Laramee, *Member, IEEE,* and Eugene Zhang, *Member, IEEE,*

Abstract—Reliable analysis of vector fields is crucial for the rigorous interpretation of the flow data stemming from a wide range of engineering applications. *Morse decomposition* of a vector field has proven a useful topological representation that is more numerically stable than previous vector field skeletons. In this paper, we enhance the procedure of Morse decomposition and propose an automatic refinement scheme to construct the *Morse Connection Graph (MCG)* of a given vector field in a hierarchical fashion. Our framework allows a *Morse set* to be refined through a local update of the *flow combinatorialization*, which leads to a more detailed MCG. This refined MCG has consistent topology with the original MCG because the refinement is conducted locally. The computation is faster than the original τ-map approach because we reuse the previous tracing information and perform only local updates.
The classification of extracted Morse sets is a crucial step for the construction of MCG. In this work, we advocate the use of Conley index for the classification. Conley index is a more general characteristic than *Poincaré index* for the classification of flow dynamics. We present a framework to compute the Conley index of an isolating block in a flow. In addition, an efficient algorithm for computing an upper bound of the Conley index of any given Morse set is introduced to assist the automatic refinement process. Furthermore, an improved visualization technique for MCG is described which conveys the classification information of different Morse sets with the aid of the visualization of their Conley indices. Finally, we apply the proposed techniques to a number of synthetic and simulation data sets to demonstrate their utility.

Index Terms—Morse decomposition, vector field topology, upper bound of Conley index, topology refinement.

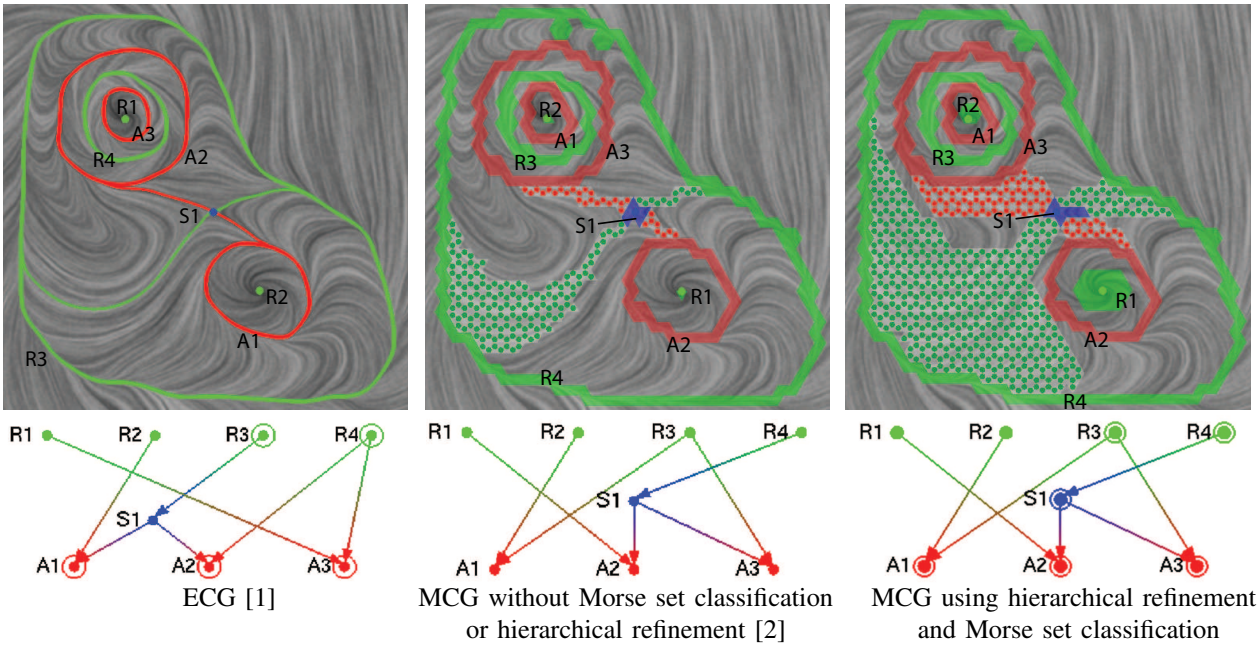
1 INTRODUCTION

NUMERICALLY stable topology of vector fields is required for the rigorous interpretation of the underlying dynamics of the flow data stemming from a wide variety of engineering applications such as Computational Fluid Dynamics (CFD), aerodynamics, tsunami modeling, and automobile and aircraft design. Conventional topology of vector fields consists of features possessing special trajectories which are either points (i.e. *fixed points*) or closed (i.e. *periodic orbits*). Chen et al. [2] has shown that this type of topology is sensitive to noise and error introduced during data acquisition and processing. To overcome this, they advocate *Morse decomposition* as a more reliable analysis of vector field topology. The result of a Morse decomposition of a vector field is an acyclic directed graph called a *Morse Connection Graph (MCG)*. The nodes of an MCG are *Morse sets* and the edges show the connectivity between them. Different from conventional trajectory-based topology of vector fields, the Morse sets in an MCG correspond to flow regions of recurrence which contain trajectory-based features such as fixed points and periodic orbits. This region-based representation accounts for error which leads to more numerically stable results than the trajectory-based topology of vector fields such as *vector field skeleton* [12] and *Entity Connection Graph (ECG)* [1]. Figure 1 compares these two different extraction methods for vector field topology.

In the pipeline of the computation of an MCG for a given vector field, the first step, *flow combinatorialization* is a key stage. The output of this process is a standard directed graph from which the MCG is derived. The nodes are the polygonal primitives of the space discretization (e.g. triangles), and the edges indicate the mapping relations between polygons. In order words, this graph encodes the flow dynamics. Since the MCG computation is based on this graph, its quality is crucial. Chen et al. introduce the idea of a $\tau-$map to compute a more accurate flow combinatorialization graph subject to the flow dynamics, from which fine MCGs can then be computed. More specifically, this method constructs the directed graph for the flow combinatorialization by keeping track of the image of each polygonal primitive over a constant time τ (reviewed in Section 3.3). This guarantees the capture of accurate dynamics of the flow but is computationally expensive. Furthermore, an ideal τ value is typically unknown for a given flow. This requires the user to carry out multiple computations with different τ values before a satisfying result is achieved. In addition, each computation is conducted in the whole flow domain. Consequently, it leads to a slow analysis process which limits the practicality of the Morse decomposition techniques.

The classification of extracted Morse sets is a crucial step for the construction of MCG. Chen et al. [2] make use

- *Guoning Chen, Qingqing Deng, and Eugene Zhang are with Oregon State University, Corvallis, OR 97331.*
  E-mail: {chengu,dengqi,zhange}@eecs.oregonstate.edu
- *Andrzej Szymczak is with Colorado School of Mines, Golden, CO 80401-1887.*
  E-mail: aszymcza@mines.edu
- *Robert S. Laramee is with Swansea University, Swansea, SA2 8PP, UK.*
  E-mail: r.s.laramee@swansea.ac.uk.



ECG [1]

MCG without Morse set classification or hierarchical refinement [2]

MCG using hierarchical refinement and Morse set classification

Fig. 1: The figure provides the ECG [1](left) and MCGs (middle and right) of an analytic vector field. In the results of ECG, sources are colored in green, sinks in red, and saddles in blue. The closed streamlines are periodic orbits with green curves for repelling orbits and red for attracting. In the results of MCGs, the colored regions are the Morse neighborhoods containing the corresponding Morse sets: green regions correspond to source-like Morse sets, red for sink-like Morse sets, and blue for saddle-like Morse sets. The color coding for the MCGs is similar to ECG. We point out that the MCG in the middle visualization lack the ability of distinguishing between a source-like Morse set from a periodic orbit-like source Morse set. The images to the right show the result of a hierarchical refinement (Section 4) of the Morse decomposition of this field as well as the MCG with Conley index information visualized (Section 3.2). The maximum allowed $\tau_{max} = 12$. In contrast to at least $11.15s$ computation time by manually adjusting τ (with $\tau = 7, 14$, and 28, respectively) our automatic refinement framework achieves the comparable result within 7.58 seconds. The color-dotted regions indicate the connections between Morse sets. The regions with green dots represent the connections between saddle-like Morse sets and source-like Morse sets, with red dots for the connections between saddle-like Morse sets and sink-like Morse sets, and with blud dots for the connections between saddle-like Morse sets.

of the flow directions at the boundary of each Morse set for classification. This method can characterize Morse sets into only three types: source-like, sink-like, and saddle-like. Compared to the ECG (Figure 1, left), this classification lacks the ability of distinguishing between a source/sink-like Morse set from a periodic orbit-like source Morse set, for instance, $R1$ and $R4$ are displayed similarly in the MCG in Figure 1 (middle).

To address the drawbacks of the slow analysis and insufficient classification of Morse sets of the previous technique, we enhance the Morse decomposition with a focus on the τ-map computation of vector fields and propose an efficient Morse decomposition framework based on a hierarchical refinement process. In addition, we advocate the use of *Conley index* for the classification of different Morse sets. An efficient computation of Conley index is introduced for the first time to assist the classification. More specifically, we modify the computation pipeline of MCG as follows. We first perform MCG computation using the *geometry-based method* [1] which is fast but coarse. Next, we enter an iterative process in which a Morse set in the current MCG is identified and refined through localized flow combinatorialization with the τ-map method of increasing τ values. This process repeats until none of the Morse sets can be further refined. See Figure 2 (a)-(d) for an example. The validity of our approach is justified by Graph Theory. Our framework is faster because the more expensive computations (higher τ values) are only performed in a small part of the mesh. Furthermore, particle tracing results from smaller τ values can be reused for the computation with higher τ values.

To compute the Conley index of a Morse set, we compute the three *Betti numbers* of a quotient space defined by the *isolating block* containing the Morse set (Section 3.2). This leads to an efficient algorithm due to our two-dimentional presumption. For simplicity and efficiency, instead of computing the actual Conley index of a given Morse set obtained using the flow combinatorialization graph, we compute its upper bound (Section 6). In all cases shown in this paper, the upper bound seems to be the same as the actual Conley index. When the flow combinatorialization graph is computed using the geometry-based method, the obtained upper bound is identical to the actual Conley index of a Morse set. The computed Conley index is then used to classify the Morse set. In addition, the Conley index is incorporated to our framework of hierarchical refinement, which yields an automatic refinement framework. In particular, the information of the Conley index of each Morse set is considered in a heuristic metric for the identification of Morse sets that require refinement (Section 7). Furthermore, a new visualization of MCG with the Conley index information of each Morse set is introduced

(Section 3.2). This allows the user to identify the different flow complexity within two Morse sets with the same basic type. For instance, $R1$ and $R4$ in Figure 1 (right) are both source-like Morse sets but with different flow complexity: $R1$ contains a fixed point while $R4$ a periodic orbit.

In the following, we first review the related work on vector field topology in Section 2. Section 3 provides the background on Morse decomposition. In Section 4, we introduce our pipeline of hierarchical Morse decomposition. Section 5 provides the detail of local flow combinatorialization. Section 6 provides the definition of an upper bound of the Conley index given a τ-map and describes an efficient algorithm to compute this bound. Section 7 then combines this upper bound with the area of a Morse set into an uniform metric to assist the automatic identification of Morse sets for refinement. The proposed framework has been applied to a number of synthetic and simulation datasets. The results are shown and discussed in Section 8 followed by a summary of the presented work in Section 9.

2 Related Work

This section reviews the related work from vector field visualization and mathematical topology.

2.1 Vector Field Topology

Since Helman and Hesselink [12] introduced the topological skeleton of vector fields to the visualization community, much work has been done to address the vector field topology extraction in 2D vector fields. In terms of fixed point extraction, Tricoche et al. [29] and Polthier and Preuß [20] present efficient algorithms to locate fixed points in a vector field. Scheuermann et al. extend the work on first-order fixed points to the higher-order fixed point analysis using *Clifford algebra* and present solutions to higher-order fixed point visualization [22]. The approaches of visualizing non-linear topology of a given vector field are presented in their work [21], [22]. Later, it is shown that more complicated recurrent flow patterns can be detected, such as periodic orbits. Wischgoll and Scheuermann are the first to present an algorithm for detecting periodic orbits in planar flows [31]. This technique has also been extended to 3D vector fields [32] and time-dependent flows [33]. Theisel et al. [26] present a mesh independent approach to compute periodic orbits. Recently, Chen et al. [1] present efficient algorithms to construct a more complete vector field topology, *Entity Connection Graph (ECG)* by including periodic orbits into the topological skeleton. They also introduce efficient algorithms to extract periodic orbits using Morse decomposition. Later, Chen et al. [2] study the instability of trajectory based vector field topology and in the first time propose the Morse decompositions for vector field topology computation which leads to more reliable interpretation of vector field topology. The presented work is closely related to Chen et al.'s work on Morse decomposition. What distinguishes this work from theirs is a hierarchical refinement framework for Morse decomposition using local computation which enables faster computation and consistent topology extraction. Furthermore, we provide an efficient algorithm to estimate Conley index for Morse sets. We point out that Conley index computation is not given detail in [2].

2.2 Morse-Smale Complex

Morse theory has been introduced by Edelsbrunner et al. [9], [8] for the analysis of scalar fields. In their work, the analysis of a scalar field is converted to the analysis of the gradient of the scalar field, which gives rise to a vector field. The Morse-Smale complex then decomposes the manifold into cells (usually quadrilateral) of uniform flow according to the gradient vector field. Note that this decomposition is equivalent to the segmentation of the flow domain using topological skeleton of vector fields [12] where the obtained cells are known as *basin*. Recent work on the Morse-Smale complex for the analysis of scalar fields can be found in [10], [11]. Our work on the Morse decompositions of vector fields is concerned with the extraction of the regions of flow recurrence containing fixed points and periodic orbits, as well as their connectivity information (Figure 1). The focus is the reliable identification of flow recurrence. In contrast to Morse-Smale complex which addresses scalar fields, Morse decomposition handles general vector fields.

2.3 Mulitscale Processing of Vector Fields

The presented work falls in the category of multiscale processing of vector fields. Multiscale processing of vector fields is an active research topic. There are two directions in multiscale processing: refinement and simplification. While this paper focuses on the refinement aspect, it is worth reviewing some simplification work on this topic. One of the earliest investigations on the subject of topology simplification in visualization was done by De Leeuw and Van Liere [3]. They make use of the distance to determine the pair of fixed points to be cancelled. In follow-up work, they perform topology simplification based on area metrics [5]. Boundary regions in the local neighborhood of sources and sinks are computed and topology is simplified based on flow regions with small areas. These techniques are applied to two important applications from vector field simulation [4]. Tricoche et al. [28] present a simplification method that also provides a piecewise analytic description for the simplified field. In this way, complementary visualizations such as texture-based methods [16] may be combined with the visualization result. They extend this method to time-dependent, 2D flows [30]. Tricoche et al. [29] also present a topology simplification method very similar to De Leeuw and Van Liere [5] however simplifications are achieved by actually modifying the vectors of the original, underlying data field. Theisel et al. [25] present an algorithm for compressing vector fields while preserving their topology. Later, they combine both topological simplification and topology preserving compression techniques [24]. The technique simplifies the topology of the underlying vector field based on assigning an importance to each critical point and separatrix. Features with less importance (below a certain weight threshold) are simplified. Compression is applied to the vector field with the simplified topology. Tong et al. [27] proposed multiscale decomposition of a vector field using Hodge-decomposition and

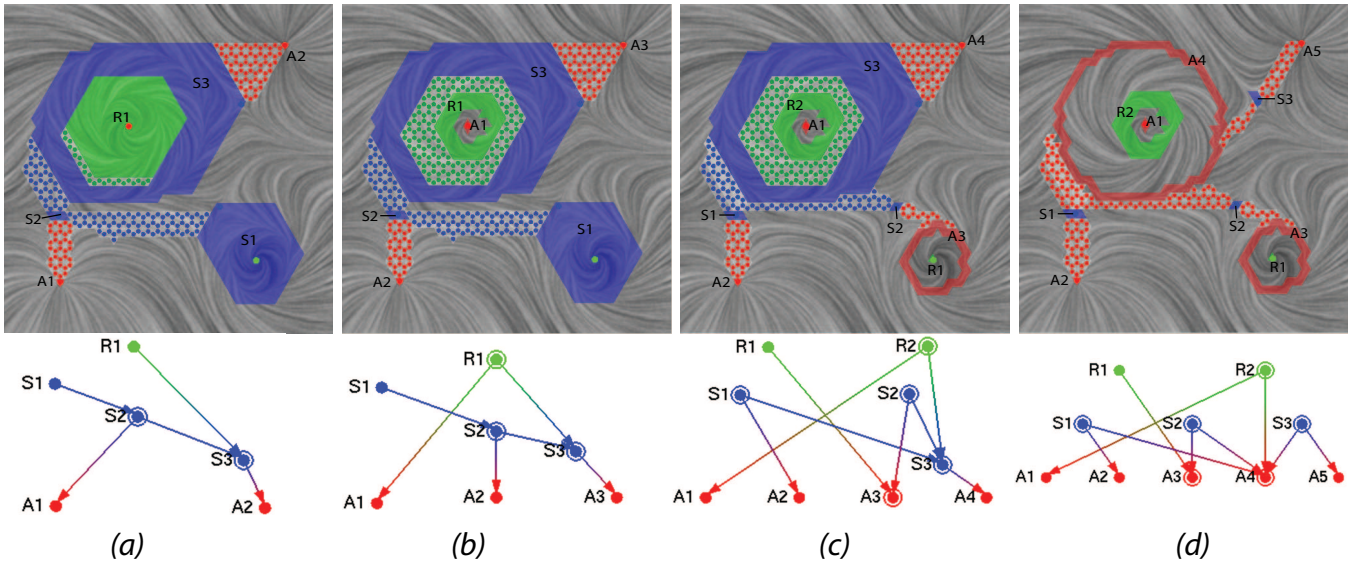


Fig. 2: This figure provides an example of locally refining a Morse decomposition of an analytic flow data over different Morse sets (b): R1, (c): S1, and (d): S3 with $\tau = 7.8$, 10, 10, respectively. (a) provides the Morse decomposition using a geometry-based method. Different colored regions correspond to different Morse sets. Note that the connection regions are also refined during the process. The corresponding MCGs are provided in the bottom row. In the MCGs, green points stand for the source Morse sets, red points for the sink Morse sets, and blue points for the saddle Morse sets. The information of the Conley index of each Morse set is also visualized (a)-(d). This provides the user the detailed classification of the extracted Morse sets (see the important Conley indices in Section 3.2). Morse set $S1$ in (a) has trivial Conley index $(0,0,0)$, but further decomposition reveals more features of interest (a saddle, a source, and a periodic orbit). Therefore, we include it in the constructed MCG (Section 7) for further refinement.

then smooth each-component independently before summing them. Zhang et al. [34] introduce a framework for fixed point pair cancellation based on Conley index theory for vector field editing. Chen et al. [1] extend this idea to include periodic orbits into this framework and present a more complete pair cancellation scenarios. For an overview of related work on vector field topology, see Laramee et al. [15].

3 Background

In this section, we review the important concepts related to Morse decomposition for the presented work.

3.1 Morse Decomposition and Morse Connection Graph (MCG)

Consider a manifold M and a subset $X \subset M$. A vector field can be expressed in terms of a differential equation $\dot{x} = V(x)$. The set of solutions to it gives rise to a *flow* on M; that is a continuous function $\varphi : \mathbf{R} \times M \to M$ satisfying $\varphi(0,x) = x$, for all $x \in M$, and

$$\varphi(t, \varphi(s,x)) = \varphi(t+s,x) \tag{1}$$

for all $x \in M$ and $t,s \in \mathbf{R}$. Given $x \in M$, its *trajectory* is

$$\varphi(\mathbf{R},x) := \cup_{t \in \mathbf{R}} \varphi(t,x). \tag{2}$$

$S \subset M$ is an *invariant set* if $\varphi(t,S) = S$ for all $t \in \mathbf{R}$. Observe that for every $x \in M$, its trajectory is an invariant set. Other simple examples of invariant sets include the following. A point $x \in M$ is a fixed point if $\varphi(t,x) = x$ for all $t \in \mathbf{R}$. More generally, x is a *periodic point* if there exists $T > 0$ such that $\varphi(T,x) = x$. The trajectory of a periodic point is called a periodic orbit.

A Morse decomposition of X for a flow φ_λ is a finite collection of disjoint compact invariant sets, called *Morse sets* [14]. The computation result of a Morse decomposition is an acyclic directed graph called *Morse connection graph*, or MCG. The Morse sets in MCG are denoted as

$$\mathbf{M}(X,\varphi) := \{M_\lambda(p) \mid p \in (\mathscr{P}_\lambda, \succ_\lambda)\},$$

where $\succ_\lambda$ is a strict partial order on the indexing set $\mathscr{P}_\lambda$ (i.e. the set of Morse sets with their Conley indices for the classification), such that for every $x \in X \setminus \cup_{p \in \mathscr{P}_\lambda} M_\lambda(p)$ there exist indices $p \succ_\lambda q$ such that

$$\omega(x) \subset M_\lambda(q) \quad \text{and} \quad \alpha(x) \subset M_\lambda(p).$$

In order words, there is a location near p such that a particle placed there can be advected by the flow and reach q. It has proven that any structures associated with recurrent dynamics of φ_λ, i.e. fixed points, periodic orbits, chaotic dynamics, must lie in the Morse sets by definition [14]. Efficient algorithms also exist for identifying the neighborhoods of the Morse sets [1], [2], [14]. The dynamics outside the Morse sets is gradient-like. Figure 2 (a) provides an example of the Morse decomposition of a vector field. In this case, $M = \{R_1,\ S_1,\ S_2,\ S_3,\ A_1,\ A_2\}$ and $M_\lambda = \{S_1 > S_2,\ S_2 > S_3,\ R_1 > S_3,\ S_2 > A_1,\ S_3 > A_2\}$.

3.2 Conley Index

The computation and visualization of MCG relies on the classification of the extracted Morse sets. Chen et al. simply resort to the direction of the directed edges emulating from the Morse sets to classify the Morse sets [2]. The more accurate classification requires the introduction of a topological descriptor called Conley index. Conley index is needed to

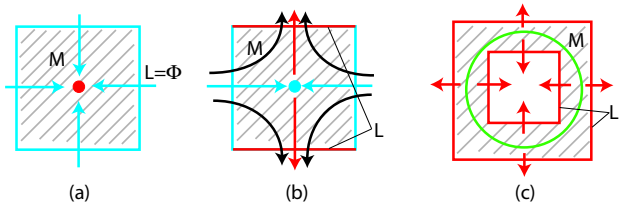


Fig. 3: A number of simple examples of isolating blocks. (a) shows a region containing a sink $(1,0,0)$; (b) is a region with a saddle $(0,1,0)$; (c) displays a ring-like region enclosing a repelling periodic orbit $(0,1,1)$. In all cases, M is the shadow region. Red lines represent the exit sets L.

classify Morse sets as well as to identify Morse sets for further refinement (Section 7).

The Conley index of a set M is easy to define if M is an *isolating block*, i.e. if every point x on the boundary of M is an *exit point* or an *entry point*. An entry point is a point x whose trajectory for sufficiently small negative times (trace in the upstream direction) is outside M. Similarly, x is an exit point if its trajectory is outside M for all sufficiently small positive times (trace following the downstream direction). Let S be the maximal invariant set in the isolating block M. The Conley index of S can be defined as the homology of the quotient space M/L, where L is the *exit set* consisting of all exit points [13]. In what follows, we call (M,L) and *index pair* and represent the Conley index of S as a sequence of Betti numbers of M/L. Let β_k be the k-dimensional Betti number. We assume that M is a subset of a two-dimensional manifold surface, a triangulation of M is available and that L is a union of boundary edges of M. Thus, the Conley index has the form of $CH_*(M) = (\beta_0, \beta_1, \beta_2)$. The higher Betti numbers are all zero based on the assumption of two-dimensional manifold. A few simple examples of isolating blocks with exit sets shown in red can be found in Figure 3. A number of important Conley indices in 2D flow analysis are as follows [1]:

$$
\begin{aligned}
x_0 \text{ an attracting fixed point(e.g.sink)} &\Rightarrow CH_*(x_0) = (1,0,0)\\
x_0 \text{ a saddle fixed point} &\Rightarrow CH_*(x_0) = (0,1,0)\\
x_0 \text{ a repelling fixed point(e.g.source)} &\Rightarrow CH_*(x_0) = (0,0,1)\\
\Gamma \text{ an attracting periodic orbit} &\Rightarrow CH_*(\Gamma) = (1,1,0)\\
\Gamma \text{ a repelling periodic orbit} &\Rightarrow CH_*(\Gamma) = (0,1,1)\\
M = \emptyset &\Rightarrow CH_*(M) = (0,0,0)
\end{aligned}
$$

From this we see that Conley index is a more general topological descriptor for the characterization of different flow features than the Poincaré index [29]. Particularly, the Poincaré index of M is $\beta_0 - \beta_1 + \beta_2$ [18].

In terms of Morse decomposition and MCGs, given the three Betti numbers of the Conley index, a Morse set can be classified as follows. If $\beta_0 = 1$, it is a sink-like Morse set (colored in red); if $\beta_2 = 1$, it is a source-like Morse set (colored in green); otherwise, it is a saddle-like Morse set (colored in blue) (see Figures 1 and 2, bottom rows).

To visualize the Conley index of each detected Morse set in the MCG, we make the following modification compared to the MCG visualization used by Chen et al. [2]. We visualize the value of the first Betti number of the Conley index using concentric circles. For instance, if $\beta_1 = 1$, the corresponding

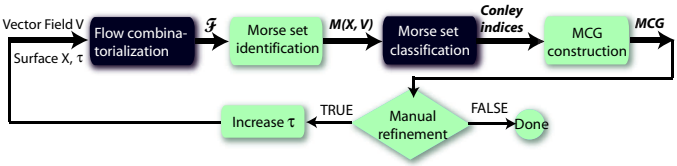


Fig. 4: The MCG computation pipeline using Morse decomposition. Note that the highlighted modules are the focus in this paper. More specifically, we introduce the idea of local flow combinatorialization and provide an efficient algorithm for the computation of Conley index.

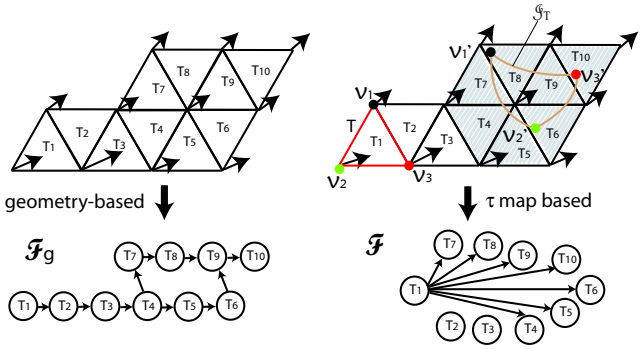
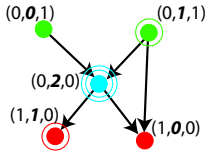


Fig. 5: Two approaches of flow combinatorialization (left) geometry-based approach and (right) τ-maps. In the directed graphs, each node corresponds to a triangle of the mesh. In the right figure, the red triangle $T = T_1$ is the starting triangle, the light brown curved closure is the real image of T. The set of triangles (shaded) that intersect with the image of T is referred to as the *outer approximation* of this image [2].

Morse set is visualized as a solid disk with a circle around it (see the inlet image for an example). Similarly, if $\beta_1 = N$, there will be N concentric circles around the solid disk. The color of the additional circles is determined by the type of the Morse set: green for source-like Morse sets, red for sink-like, and blue for saddle-like. This improved visualization of MCG enables the user to distinguish the periodic orbit like Morse sets from the source or sink like Morse sets. The resulting MCGs have similar appearance as ECG (Figure 1, left) with one difference on the visualization of saddles. Specifically, in an ECG saddles are displayed as blue disks, while in an MCG they are drawn as blue disks surrounded by a few blue circles (see Figures 1, left and right).



3.3 Computation of Morse Decomposition

We now turn to a review of the computation of Morse decompositions.

In this work, the underlying domain is represented by a triangular mesh. Vector values are defined at the vertices only, and interpolation is used to obtain values on the edges and inside triangles. For the planar case, we use the piecewise linear interpolation method [29]. On curved surfaces, we borrow the interpolation scheme of Zhang et al. [34], which guarantees vector field continuity across the vertices and edges of the mesh. These interpolation schemes support efficient flow analysis operations on both planes and surfaces.

Chen et al. [2] describe a pipeline for the computation of Morse decompositions of the given vector fields. In this pipeline, the input vector field is firstly converted into a directed graph, denoted by $\mathscr{F}$, through the flow combinatorialization process. The nodes of $\mathscr{F}$ are the individual triangles of the mesh where the vector field is defined. The directed edges indicate the flow mapping relations between triangles. For instance, if there exists a directed edge $T_1 \rightarrow T_2$, the particles inside triangle T_1 can be advected by the flow and enter T_2. In order words, $\mathscr{F}$ encodes the dynamics of the flow at a combinatorial level. There are two approaches to compute $\mathscr{F}$: the geometry-based approach [1] and the $\tau-$map approach [2]. The geometry-based method computes the directed edges by considering the flow behavior across each edge of each triangle. The $\tau-$map approach keeps track of the image of each triangle over a constant time τ to obtain the directed edges. Figure 5 illustrates these two approaches. The light brown curved closure (Figure 5 right) is the real image of T. The set of triangles (shaded) that intersect with the image of T is referred to as the *outer approximation* of this image. Second, the strongly connected components are extracted from the directed graph $\mathscr{F}$. These strongly connected components correspond to the triangle regions of flow recurrence. The Conley index of each triangle region is computed. Those regions with non-trivial Conley indices contain the Morse sets of interest [14]. Third, each strongly connected component corresponding to the region of non-trivial Conley index is collapsed into a single node which reduces the original graph into a quotient graph. From the quotient graph, the final MCG can be computed through path search between these strongly connected components using standard graph search algorithms. Note that the search starts from the node corresponding to source-like Morse sets or saddle-like Morse sets, and ends at saddle-like Morse sets or sink-like Morse sets. The complete algorithm of this pipeline is provided in [2]. We point out that Chen et al. do not provide the algorithm for Conley index computation which will be addressed in this presented work.

The computation of the Morse decomposition of a vector field typically requires repeating experiments with different τ's according to the user's requirements before a satisfying MCG is returned. Figure 4 illustrates such an iterative process. This manual process can be labor intensive and inconsistent in the sense that each trial recomputes the whole flow combinatorialization. We address this challenge by introducing an automatic local update scheme for the flow combinatorialization stage, which is the most expensive step in the pipeline.

4 Pipeline of the Hierarchical Morse Decompositions

Theory of dynamical systems shows that an *isolating neighborhood* (a polygonal region under discrete setting) exists for each Morse set [14]. In addition, Chen et al. [2] demonstrate that computing the flow combinatorialization does not require a constant τ value everywhere in the domain. This leads us to a local refinement scheme with various τ values. Next we describe our pipeline which is an improvement over the approach of Chen et al. [2].

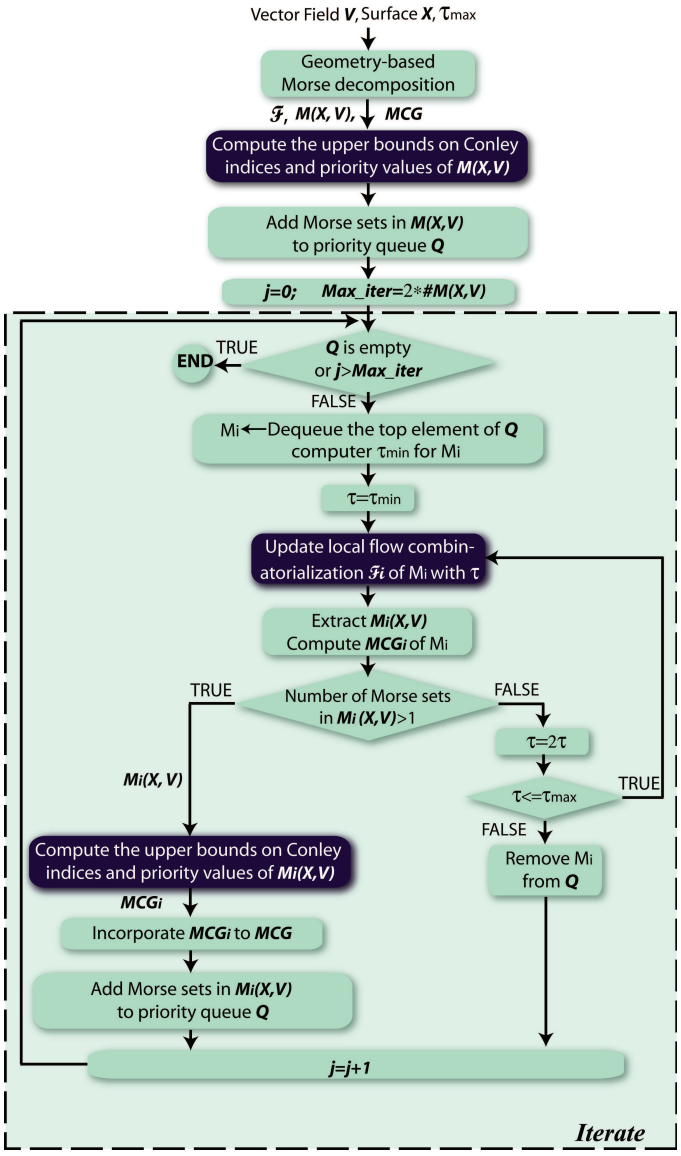


Fig. 6: The pipeline of the proposed locally hierarchical refinement of Morse decompositions of vector fields. Note that the highlighted modules are the focus in this paper (Section 5 for local refinement, Section 6 for Conley index computation, and Section 7 for the computation of the priority values).

First, an MCG is computed from the Morse decomposition using a geometry-based flow combinatorialization. Denote the resulting Morse sets as $M(X,V)$ associated with their connectivity graph MCG. Second, we compute an upper bound of the Conley index of each detected Morse set (Section 6) and compute a priority value based on this upper bound, the area of the Morse set, and the variance of vector magnitude inside it (Section 7). Third, we add these Morse sets into a priority queue Q with the Morse set having larger priority values close to the top of the queue. We also initialize $\tau = \tau_{min}$ and set $j = 0$. In practice, a good heuristic of τ_{min} is the ratio between the average edge length of local mesh of a Morse set and the minimum vector magnitude inside the Morse set. Fourth, if Q is not empty, we proceed as follows. Let M_i be the first Morse set in Q. We remove it from Q, and set $\tau = 2\tau$.

Then, we perform a τ-map based flow combinatorialization inside the bounded Morse neighborhood of M_i to obtain an updated directed subgraph $\mathscr{F}_i$. Next, the strongly connected components $M_i(X,V)$ of $\mathscr{F}_i$ are extracted and a local MCG_i is computed. If the number of Morse sets in $M_i(X,V)$ is larger than 1, which means selected Morse set is refined, we compute the upper bounds of the Conley indices of the new Morse sets and their priority values. Then, we incorporate MCG_i into MCG and add $M_i(X,V)$ into Q based on their priority values. If the number of Morse sets in $M_i(X,V)$ equals 1 and $2\tau \leq \tau_{max}$ (where τ_{max} is a user specified maximum τ), we set $\tau = 2\tau$ and proceed as before until either $M_i(X,V)$ contains more than one Morse set or $\tau > \tau_{max}$. If the selected Morse set can not be refined even if $\tau > \tau_{max}$, we remove this Morse set from the priority queue and do not consider it for future processing. Then, we proceed to step 4 and iterate the process until Q is empty, i.e., all Morse sets have been tested and found to be not refinable. Figure 6 illustrates this pipeline. The highlighted modules are the focus of this paper. The implementation of local flow combinatorialization is provided in Section 5. The computation of Conley index and its upper bound is discussed in Section 6. The computation of the priority value is introduced in Section 7. This pipeline proceeds in a hierarchical fashion and is expected to produce an MCG (Figure 1, right) similar to the one produced by previous manual τ-map approach with respect to globally applied τ (Figure 1, middle). More importantly, the intermediate and final MCGs produced using this pipeline are guaranteed to be topologically consistent. In contrast to that, the MCGs generated with different τ values using global computation scheme (i.e. Figure 4) lack such consistency.

5 Local Flow Combinatorialization

Similar to the general Morse decomposition, in the hierarchical Morse decomposition pipeline the local flow combinatorialization is a key step. In this section, we show that locally updating the flow combinatorialization will not affect the flow structure with respect to MCG outside of the bounded Morse neighborhood of interest. Before presenting the theorem, we first prove the follow lemma.

Lemma 5.1: Given a directed graph $G = (V,E)$. Let $V = \cup_i V_i$ be the decomposition of strongly connected components of G. Let $E_i = \{(v_p,v_q) \in E | v_p, v_q \in V_i\}$. Select an integer j, let $G' = (V,E')$ such that $E' = (E - E_j) \cup E'_j$ where E'_j contains edges whose end points are both in V_j and $E_j \neq E'_j$ (i.e. they represent different sets of edges). We make two claims:

1) Any strongly connected component V_i in G with $(i \neq j)$ is contained in a strongly connected component in G'.
2) Any strongly connected component V'_i in G' is contained in a strongly connected component in G.

If we can show 1 and 2 are true, then we immediately have the following: A strongly connected component V_i $(i \neq j)$ in G is also a strongly connected component in G'. On the other hand, V_j corresponds to possibly more than one strongly connected components.

Lemma 5.1 states that changing certain edges in the interior of a strongly connected component (i.e. a sub-graph) of a directed edge will not affect the other parts of the graph. Also, the updated sub-graph corresponding to the strongly connected component will contain at least one strongly connected component. Figure 7 provides such an example. In this example, a strongly connected component S_k in a directed graph is decomposed into two strongly connected components S'_{k1} and S'_{k2} after altering one directed edge inside S_k. Rigorous proof of Lemma 5.1 is provided in Appendix A.

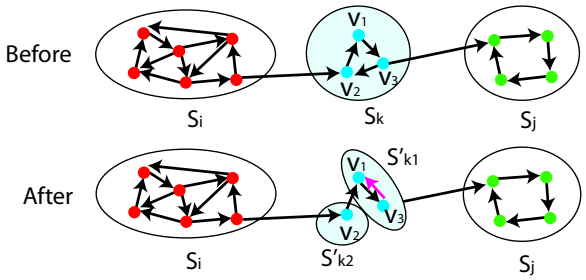


Fig. 7: An example for the illustration of Lemma 5.1. Note that after changing the edge $(v3 \rightarrow v2)$ to be $(v3 \rightarrow v1)$, the original strongly connected component S_k (the shaded region) decomposes to be two strongly connected components S'_{k1} and S'_{k2}.

Theorem 5.2: Consider a flow combinatorialization $\mathscr{F} = (V,E)$ of a vector field computed with respect to a τ. Let $M = \cup_i M_i$ be the set of extracted Morse sets from $\mathscr{F}$, $\mathscr{F}_i = (V_i,E_i)$ be the subgraph of $\mathscr{F}$ with V_i the set of triangles corresponding to the Morse neighborhood of M_i, and $E_i = \{(v_p,v_q) \in E | v_p, v_q \in V_i\}$. For an integer j, let $\mathscr{F}' = (V,E')$ such that $E' = (E - E_j) \cup E'_j$ where E'_j consists of edges that are computed using a $\tau' > \tau$ and whose end nodes are both in V_j and $E_j \neq E'_j$ (i.e. they represent different sets of edges). Then,

1) Any Morse set M_i of $\mathscr{F}$ with $(i \neq j)$ is a Morse set of $\mathscr{F}'$.
2) M_j is a) a Morse set, b) decomposed to be more than two separate Morse sets, or c) removed from M.

Combining 1 and 2 of Theorem 5.2 we prove: By fixing all Morse sets but one and replacing edges of the local flow combinatorialization graph for the given Morse set using a larger τ value, we obtain a refined Morse sets for that Morse set without changing the rest of $\mathscr{F}$ in practice. Consequently, the whole MCG is refined. Note that in some cases the selected Morse set needs not be a true Morse set (with non-trivial Conley index) and can be removed from the Morse set list after replacing its local flow combinatorialization graph with a more accurate one. This is because the flow combinatorialization constructed using a small τ or the geometry-based method is typically coarse and contains misleading information, e.g. inaccurate Morse sets. Nonetheless, this error can be corrected by a more accurate graph obtained using a larger τ. The proof of this theorem follows from Lemma 5.1. This theorem also provides the algorithm necessary to realize the local MCG refinement which we will describe in the following section.

5.1 Implementation

Given a Morse set M_i, we now discuss how to refine it locally. Let $\mathscr{F}_i$ be its local flow combinatorialization graph whose

Algorithm 1: Locally refining one Morse set

Input: M_i: the given Morse set
X_i: the list of triangles of Morse set M_i
$\mathscr{F}_i$: the local flow combinatorialization of M_i
$V(M_i)$: the local flow inside Morse set M_i
L: the maximum level for adaptive edge sampling
Output: MCG_i: the local MCG with updated Morse sets
Begin
$Delete\_Edges\_in\_\mathscr{F}_i(\mathscr{F}_i)$;
$\mathscr{F}_i \leftarrow Reconstruct\_\mathscr{F}_i(X_i, V(M_i), \tau, L)$;
$Extract\_SCC\_\mathscr{F}_i(\mathscr{F}_i)$;
$MCG_i \leftarrow Construct\_Local\_MCG_i(\mathscr{F}_i)$;
End

Local flow combinatorialization using τ-map
Routine: $Reconstruct\_\mathscr{F}_i(X_i, V(M_i), \tau, L)$

Input: X_i: the list of triangles of Morse set M_i
$V(M_i)$: the local flow inside Morse set M_i
τ: a positive real number
L: the maximum level for adaptive edge sampling
Output: $\mathscr{F}_i$: the local flow combinatorialization of M_i
Local variables: $\mathscr{F}_{temp}$: temporary $\mathscr{F}_i$
e: an edge in $\mathscr{F}_{temp}$
$node[2]$: two nodes that are connected by e
Begin
$\mathscr{F}_{temp} \leftarrow construct\_multivaluemap(V(M_i), X_i, \tau, L)$;
For each edge e in $\mathscr{F}_{temp}$
If $node[0] \notin X_i$ or $node[1] \notin X_i$
remove e from $\mathscr{F}_{temp}$;
EndFor
$\mathscr{F}_i \leftarrow \mathscr{F}_{temp}$;
End

nodes are the triangles of the Morse neighborhood of M_i and directed edges encode the flow dynamics in M_i. First, we remove all the edges that are completely in $\mathscr{F}_i$, i.e, the edges whose both end points are nodes in M_i. Second, we reconstruct $\mathscr{F}_i$ by applying the τ-map approach locally inside M_i. Third, we extract strongly connected components from the updated $\mathscr{F}_i$. Finally, we compute the subgraph MCG_i based on the extracted Morse sets and incorporate it into the original MCG. Algorithm 1 provides the pseudo code of this process. The reconstruction of local flow combinatorialization is crucial. To realize this, we implemented a function similar to the *construct_multivaluemap* routine [2] but input a bounded flow region $X_i \subset X$. The obtained flow combinatorialization graph $\mathscr{F}_{temp}$ is incorporated into $\mathscr{F}$ excluding edges not completely falling in X_i. This process is described in the routine $Reconstruct\_\mathscr{F}_i()$.

Figure 2 provides an example of the MCG refinement results using our local updating scheme. We start from an MCG (left) computed from a flow combinatorialization using a geometry-based method. Next, we perform a local update inside the extracted Morse sets with large area. Note that in addition to the refined Morse sets, the connection regions [2] between two Morse sets are refined due to the refinement of the underlying $\mathscr{F}$ which we use to compute the connection region (the dotted regions in Figure 2 (a)-(d)).

6 Conley Index Computation

As described in Section 3.2, the Conley index can be represented as three Betti numbers of the quotient space M/L. In this section, we will describe the algorithm for computing these three Betti numbers. Note that the computation of Betti numbers has been discussed by Delfinado and Edelsbrunner [6]. In their work, the Betti numbers for a simplicial complex (e.g. a triangulation in $\mathbb{R}^3$) are computed in an incremental fashion. Different from their work, we discuss the computation of the Betti numbers of a quotient space derived from a 2D simplicial complex.

6.1 Computing Betti numbers of a quotient space

β_0 simply counts the number of connected components in M/L that do not contain the distinguished point, i.e. the number of connected components in M that are disjoint with L. Thus, β_0 is easy to compute. In particular, if M is connected, then β_0 is zero if $L \neq \emptyset$ and 1 otherwise. Each of the three isolating blocks shown in Figure 3 has one connected component. The exit set is empty only for the leftmost block. Therefore, the β_0 for this block is 1. For the other two blocks, $\beta_0 = 0$.

β_2 is equal to the number of connected components of M whose entire boundary is contained in L. This is because if there is a boundary edge of M that is not in L, the pair (M, L) can be reduced to a homotopy equivalent pair where both sets are one-dimensional by means of elementary collapses [23]. On the other hand, the formal sum of all triangles in each connected component M_0 of M whose boundary is contained in L defines a generator of the two-dimensional homology group of M/L and these generators span the entire homology group [7]. For the first two isolating blocks shown in Figure 3, $\beta_2 = 0$ since not all their boundary edges are exit edges. For the rightmost block, $\beta_2 = 1$.

To compute β_1, we can view the quotient space M/L as a two-dimensional CW-complex whose two-, one- and zero-dimensional cells correspond to triangles, edges and vertices (respectively) in $M \setminus L$. Additionally, there is one special zero-dimensional cell that corresponds to L. Therefore, $\beta_k = 0$ for $k > 2$. Also, the Euler characteristic of the quotient space, $\chi(M/L)$, is easy to compute: it is equal to $n_2 - n_1 + n_0$, where n_i is the number of its i-dimensional cells of M/L with the distinguished point taken out. On the other hand, $\chi(M/L) = \beta_2 - \beta_1 + \beta_0$ ([7]). Since β_2 and β_0 are already known, this equation uniquely determines β_1. Note that the Euler characteristic of the quotient space can also be computed from the Euler characteristics of M and L: $\chi(M/L) = \chi(M) - \chi(L)$ [7]. Therefore, β_1 can be computed given the information above:

$$\beta_1 = \beta_0 + \beta_2 - (\chi(M) - \chi(L)) \tag{3}$$

Figure 3 provides a number of simple cases of Conley index computation. For instance, consider a region M containing a sink (Figure 3, left). Clearly $\chi(M) = 1$ since M is a topological disk and hence $\chi(M/L) = \chi(M) - \chi(L) = 1$. $\beta_0 = 1$ and $\beta_2 = 0$ according to the previous analysis. β_1 can then be determined to be 0 from equation 3. Hence the Conley index for a sink (represented as the vector of Betti numbers $(\beta_0, \beta_1, \beta_2)$) is $(1, 0, 0)$. Similarly, we can compute the Conley index of a saddle (middle) as $(0, 1, 0)$. In this case, the Euler characteristic of the quotient space is $\chi(M/L) = \chi(M) - \chi(L) = 1 - 2 = -1$. Now, let's examine the case of a repelling periodic orbit (right). In this case, $\beta_2 = 1$ since M is connected and L is equal to its

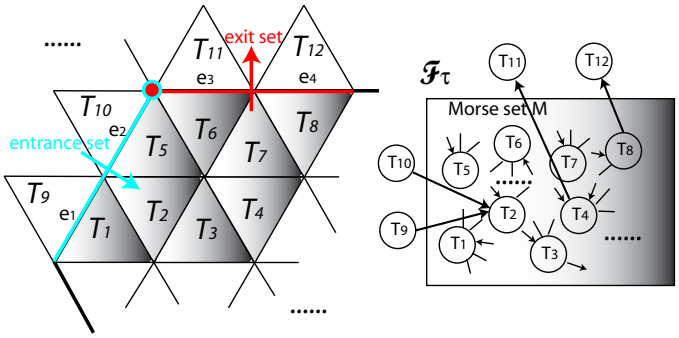
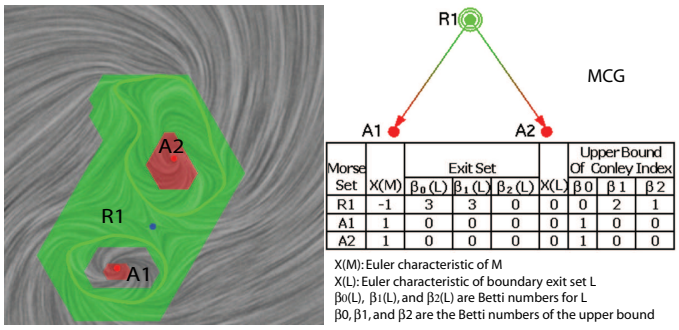


Fig. 8: This illustrates the classification of boundary edges. Image to the left provides a portion of the mesh with a Morse set M_i inside the shadow region. Right diagram provides the configuration of a discrete map (i.e. a flow combinatorialization) $\mathscr{F}_\tau$. Note that we ignore the inner configuration of the graph inside the Morse set M_i because it does not affect the classification.



Morse Set	X(M)	Exit Set β0(L)	Exit Set β1(L)	Exit Set β2(L)	X(L)	Upper Bound Of Conley Index β0	Upper Bound Of Conley Index β1	Upper Bound Of Conley Index β2
R1	-1	3	3	0	0	0	2	1
A1	1	0	0	0	0	1	0	0
A2	1	0	0	0	0	1	0	0

Fig. 9: This figure illustrates an example on how the upper bound of the Conley index can help identify Morse set with complex flow. Note that Morse set $R1$ has an upper bound on its Conley index as $(0,2,1)$. In the meantime, the flow in this Morse set contains two repelling periodic orbits (green loops) and a saddle (blue dot). Therefore, based on its upper bound, we determine that $R1$ should be further refined.

boundary. $\beta_0 = 0$ because the only connected component of M intersects L. Also, $\chi(M) = 0$ and $\chi(L) = 0$ (M is homotopy equivalent to a circle and L to two circles). Hence $\chi(M/L) = 0$ and $\beta_1 = \beta_2 + \beta_0 - \chi(M/L) = 1$. The index is $(0,1,1)$.

Any Morse set computed using the geometry method is an isolating block. Therefore, using the above computation returns the the actual Conley index of each such Morse set. However, Morse sets arising from flow combinatorialization are not guaranteed to be isolating blocks. Therefore, they require a different technique for Conley index estimation, described in the next section.

6.2 Conley index of Morse sets obtained with τ-map

In this section, we introduce a simple and efficient algorithm to compute an upper bound on the Conley index of a Morse set M obtained using the τ-map. In all cases we examine, our bound is identical to the Conley index (hence we call it the *estimate* of the Conley index) and is much simpler to compute. In addition, being an upper bound means that a Morse set containing complex dynamics will not be missed during the identification stage.

We start off by computing an index pair for a Morse set M using a simple adaptation of the algorithm of [23]. The index pair has the form (M,L) where L consists of boundary edges of M called *exit edges*. In order to decide if an edge e on the boundary of M is an exit edge, take a triangle T incident upon e and outside M. e is an exit edge if and only if there is an edge of the flow combinatorialization that *starts* at a triangle in M and *ends* at T. Let us stress that the index pair described above is an index pair for the *discrete* dynamical system that translates a point by time τ along the flow. It is not an index pair for the flow itself, in the sense of Section 3.2. In particular, one cannot claim that the Conley index is the same as the homotopy type of M/L. However, in Appendix B we give a proof that the Betti numbers of M/L are an upper bound for the Betti numbers of the Conley index of the flow on Morse set M.

An example is shown in Figure 8. Consider a boundary edge e_1 shared by triangles T_1 and T_9 with T_9 outside of M. There is no directed edge pointing from a triangle in M into T_9. Therefore, e_1 is not an exit edge. Now consider edge e_3. Triangle T_{11} is incident to e_3 and outside of M. There is a directed edge from T_4 inside M to T_{11}. Hence, $e3$ is an exit edge.

It is worth noting that this computation is essentially identical to the examples of Conley index computation shown in Figure 3. The only difference is the approach of determining the exit edges. In the examples of Figure 3, we make use of the flow at the edges to classify the edges, while in upper bound computation, the directed edges in the flow combinatorialization graph $\mathscr{F}$ are used for classification. Furthermore, the aforementioned computation on a discrete map (a flow combinatorialization graph) obtained using a geometry-based method returns the true Conley index of the given region based on the flow at the boundaries. Figure 9 provides such an example. We have applied our upper bound computation algorithm to two analytic datasets. Figure 10 provides the upper bounds of the Conley indices of the Morse sets extracted from two designed vector fields.

7 Identifying Morse Sets To Refine

In this section, we introduce two metrics used to identify Morse sets for refinement in the automatic framework.

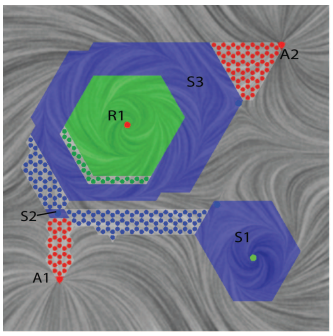
On the finest level of the hierarchy, one would like to obtain Morse sets that correspond to hyperbolic fixed points or periodic orbits (whenever possible). Thus, our first metric is defined as the distance of the Conley index of the Morse set to the closest Conley index of a hyperbolic fixed point or periodic orbit. More precisely, let $\mathscr{E}$ be the set of all possible indices of hyperbolic fixed points and periodic orbits, i.e. (Section 3.2)

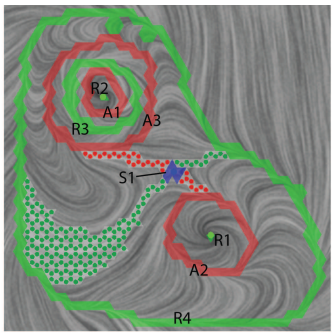
$$\mathscr{E} = \{(1,0,0),(0,1,0),(0,0,1),(1,1,0),(0,1,1)\}.$$

The *topology metric* of a Morse set M is defined by

$$\mathrm{tm}(M) = \min\left\{\sum_{k=0}^{2} |\beta_k(M) - \beta_k| : (\beta_0,\beta_1,\beta_2) \in \mathscr{E}\right\}.$$

The topology metric is not a sufficient refinement criterion. For instance, the Morse set $S3$ in Figure 10 (left) has the same Conley index $(0,1,0)$ as a region containing a saddle. However, detailed analysis shows it contains a saddle and





Morse Set	X(M)	Exit Set			X(L)	Upper Bound Of Conley Index		
		β_0(L)	β_1(L)	β_2(L)		β0	β1	β2
R1	1	1	1	0	0	0	0	1
S1	1	1	0	0	1	0	0	0
S2	1	2	0	0	2	0	1	0
A1	1	0	0	0	0	1	0	0
S3	0	1	0	0	1	0	1	0
A2	1	0	0	0	0	1	0	0

Morse Set	X(M)	Exit Set			X(L)	Upper Bound Of Conley Index		
		β_0(L)	β_1(L)	β_2(L)		β0	β1	β2
R1	1	1	1	0	0	0	0	1
R2	1	1	1	0	0	0	0	1
R3	0	2	2	0	0	0	1	1
A1	0	0	0	0	0	1	1	0
R4	0	2	2	0	0	0	1	1
S1	1	2	0	0	2	0	1	0
A2	0	0	0	0	0	1	1	0
A3	0	0	0	0	0	1	1	0

Fig. 10: The computed upper bounds of the Conley indices of all Morse sets extracted from two analytical vector fields. (left) shows the results using a geometry-based method, while (right) provides the results of an MCG derived from a τ-map with $\tau = 24$. Note that the upper bounds for the Morse sets in the left example are their actual Conley indices. In addition, in our experience the obtained upper bounds for the Morse sets computed from a τ-map approach are typically equal to the ideal Conley indices, such as, in the example to the right.

an attracting periodic orbit and therefore should be refined. To handle such cases, we make use of the *geometry metric*, defined as the number of triangles in the Morse set M and denoted by $\mathrm{gm}(M)$. It is intuitive that a Morse set containing a large number of triangles may contain more detailed dynamics. For instance, Figure 11 (left) shows the result of the Morse decomposition of the gas engine dataset using a geometry-based approach. Note that there is a Morse set at the back of the cylinder of the engine which covers a large portion of the engine surface.

Combining the above topology and geometry metrics, we define the priority $P(M)$ of a Morse set M by

$$P(M) = (1 + \mathrm{tm}(M))\mathrm{gm}(M). \quad (4)$$

This priority value is used to determine the order of the refinement of Morse sets in the current MCG. The larger the value is, the earlier the Morse set will be refined. Morse sets containing 1 triangle are not considered for refinement.

Ring-like regions containing a periodic orbit have larger P value since they contain many triangles (for example, Morse sets $R1$ and $R4$ in Figure 10 (right)). In this case, further refinement will discover that no more Morse sets can be extracted. The system then discards these Morse sets from the list of Morse sets being considered for refinement according to the pipeline (Figure 6).

Figure 11 provides the result of the consecutive refinement of a gas engine simulation data.

8 Applications

We have applied our automatic hierarchical refinement framework to a number of analytic and real world simulation datasets. Figure 1 (right) provides one result. This dataset consists of $6,144$ triangles. Our experiment took 7.58 seconds to return the result given $\tau_{max} = 28$. We compare our result with the one obtained using a manually refining process with $\tau = 7$, 14, and 28, respectively. This requires $11.15s$ computation time in addition to the user interaction time. Note that our result achieves the similar MCG structure to the one using a global τ-map approach.

We also provide the Morse decomposition results of a gas engine simulation dataset using the proposed hierarchical framework. This data set is the extrapolated boundary velocity fields that are obtained through a simulation of an in-cylinder flow [17]. Figure 11 provides the analysis results of this data set composed of $26,298$ triangles. From (1)-(3)of Figure 11 we refine the circulated Morse set with $\tau = 0.1$, 0.3, and 0.3, respectively. The times spent on these refinement are $1.469s$, $27.845s$, and $42.094s$, respectively. The MCG obtained using a global $\tau = 0.3$ shows the similar results (right-most) which took around $227s$. to compute. This is assuming that the user has already known $\tau = 0.3$ is sufficient for this case. Figure 12 compares the results of Morse decompositions using our automatic hierarchical refinement scheme versus manually adjusting τ values. In the setting of $\tau_{max} = 1$, our framework takes 158.5 seconds to compute the results. On the other hand, manually adjusting τ uses $215.29s$ for $\tau = 0.25$, $277.24s$ for $\tau = 0.5$, and $335.49s$ for $\tau = 1$ before obtaining the final result. This takes at least $828.02s$ in total.

Figure 13 provides the automatic simplification results of a diesel engine dataset with $221,574$ triangles. The running time for this analysis is $278.7s$ given $\tau_{max} = 0.4$, compared to the $1,326.08s$ using a global $\tau = 0.4$. From these results, we see the proposed automatic refinement framework achieve the analysis results about four times faster than the global τ scheme.

Table 1 provides the timing information of the automatic refinement that is applied to the two engine simulations. Note that we compare the performance of the automatic refinement framework only with the global τ approach with $\tau = \tau_{max}$. Additional time spent on smaller τ values and the user interactions for the global τ scheme is not considered. Even though, our automatic refinement framework exhibits better performance in time. Note all times are measured on a 3.6 GHz PC with 2GB RAM.

9 Conclusion

In this paper, we have identified a major drawback of the previous τ-map based Morse decomposition methods and propose a hierarchical refinement framework for the Morse decompositions of vector fields. More specifically, our re-computation is restricted to Morse sets identified through a hybrid metric that includes Conley index. Conley index is a more general topological descriptor than Poincaré index. In this work, we advocate the use of Conley index to classify the extracted Morse sets and introduce an efficient technique to compute Conley index for the analysis of 2D flow. In addition, we present an algorithm to compute the upper bound on the Conley index of a given Morse set based on a flow combinatorialization graph (a discrete map). We have proven

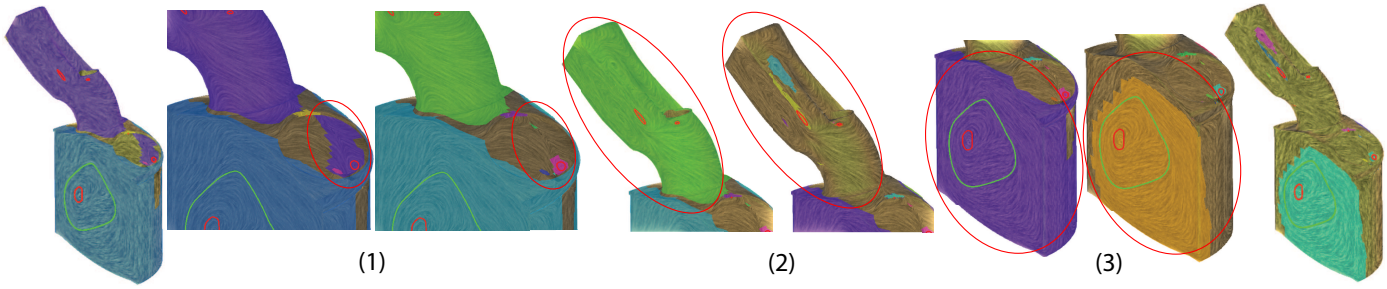


Fig. 11: This figure illustrates the refinement process of the MCG of a gas engine simulation. Left-most shows the MCG of a geometry-based approach. From (1)-(3) we refine the circulated Morse set with $\tau = 0.1$, 0.3, and 0.3, respectively. The times spent on these refinements are 1.469s, 27.845s, and 42.094s, respectively. The MCG obtained using a global $\tau = 0.3$ shows the similar results (right-most) which took around 227s to compute. This is assuming that the user already knows that $\tau = 0.3$ is sufficient for this case.

TABLE 1: The complexity and timing results for two CFD data simulating in-cylinder flow through a combustion engine. Times (in seconds) are measured on a 3.6 GHz PC with 2GB RAM. Note that we compare only the performance of the automatic refinement framework with the global τ approach with the $\tau = \tau_{max}$. Additional time spent on smaller τ values and the user interactions for the global τ scheme is not considered. Even though, our automatic refinement framework exhibits better performance in time.

Dataset name	# polygons	Fig.	Global Update			Local Update			Speed-Up Fa ctor
			τ	#Morse sets	time(s)	τ_{max}	#Morse sets	time(s)	
gas engine	26,298	12	1	62	335.49	1	62	158.5	2.12
diesal engine	221,574	13	0.4	207	1,326.08	0.4	152	278.7	4.76

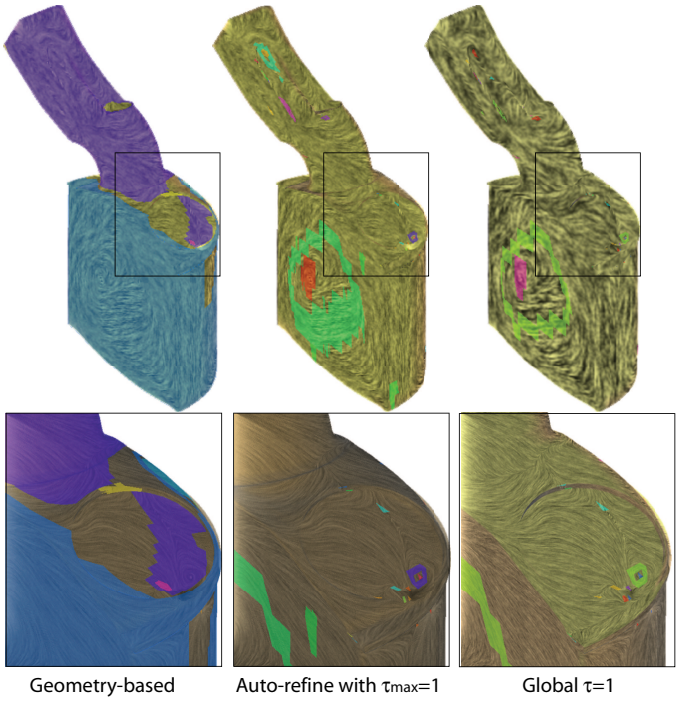


Fig. 12: This figure compares the results of the Morse decompositions of the flow field on the boundary geometry of a gas engine. Left: the result using the geometry-based method, middle: the result of the automatic refinement with $\tau_{max} = 1$, and right: the result using a global $\tau = 1$. The Morse sets are colored differently to emphasize their difference. Note that how the automatic framework produces the comparable results (middle) to the ones using a global τ (right).

the soundness of our hierarchical framework and provided examples of applications to the engine simulation data which demonstrates the effectiveness of the frame work.

The proposed framework improves the performance of Morse decompositions by a factor of 4. It also conducts the analysis in a topologically consistent fashion. In addition, the hierarchical framework and the computation algorithm for the upper bound of the Conley index raises a future direction in vector field simplification and multiscale processing and visualization. We plan to investigate this in the future.

APPENDIX A

Lemma 5.1:
Given a directed graph $G = (V,E)$. Let $V = \cup_i V_i$ be the decomposition of strongly connected components of G. Let $E_i = \{(v_p, v_q) \in E | v_p, v_q \in V_i\}$. Select an integer j, let $G' = (V,E')$ such that $E' = (E - E_j) \cup E'_j$ where E'_j contains edges whose end points are both in V_j and $E_j \neq E'_j$ (i.e. they represent different sets of edges). We make two claims:

1) Any strongly connected component V_i in G with $(i \neq j)$ is contained in a strongly connected component in G'.
2) Any strongly connected component V'_i in G' is contained in a strongly connected component in G.

If we can show 1 and 2 are true, then we immediately have the following: A strongly connected component V_i $(i \neq j)$ in G is also a strongly connected component in G'. On the other hand, V_j corresponds to possibly more than one strongly connected components.

Proof: To show (1) is true, let a, b be two nodes in V_i $(i \neq j)$, i.e., there are paths entirely contained in E_i from a to b and b to a. Given the construction of G', we know these paths are also contained in G'. Consequently, it is possible for a to reach b and vice versa using edges in G'. This means a and b are in the same connected component of G'.

To show (2) is true, assume that a and b are two nodes in the same connected component in G' but in different strongly connected components V_a and V_b with respect to G. This means

Fig. 13: This figure shows the results of Morse decomposition before (left) and after automatic hierarchical refinement (right) for the diesel engine simulation dataset. We set $\tau_{max} = 0.4$ for the hierarchical refinement. The Morse sets are colored differently to emphasize their difference. Note that how the automatic framework produces the comparable results (middle) to the ones using a global τ (right).

that there is a minimal oriented loop γ containing a and b using edges in E'.

Since $V_a \neq V_b$, one of them is different from V_j. Without loss of generality, assume $V_a \neq V_j$, i.e., $a \notin V_j$.

Let $s \in V_j$ be the last node on the loop γ before reaching a, and $t \in V_j$ the first node on γ after a. Note that either s and t both exist or neither exists. In the first case, we have a path from s to t using edges in $E - E_j$ through a, i.e, such a path exists in G. Note that s and t both belong to E_j, so there is a path from t to s with edges in E. Consequently, there is a loop from s to a with edges entirely in E. Consequently, a and s belong to the same strongly connected component V_j since $s \in V_j$. However, this contrasts our assumption $V_a \neq V_j$. In the second case, i.e., there are no nodes on γ that belong to V_j. Consequently, edges on γ are entirely contained in $E - E_j \subset E$. This means a and b belong to the same connected component. However, this contradicts our assumption that $V_a \neq V_b$. Q.E.D.

APPENDIX B

The Conley indices of the Morse sets obtained with τ-map are the upper bounds of the actual Conley indices of these Morse sets.

Proof: By the results of [23], the pair (M,L) is a valid index pair for the continuous map $\varphi_\tau = \varphi(\tau,.)$. In [19], it is shown that the k-dimensional Betti number of the Conley index of the flow on M is equal to

$$\lim_{n\to\infty} \text{rank}\varphi^n_{\tau,k} \quad ,$$

where $\varphi_{\tau,k}$ is the automorphism induced by φ_τ on the k-dimensional homology (with rational coefficients) of the quotient space M/L. The rank of an automorphism cannot be higher than the dimension of the vector space it acts on: in particular, the rank of $\varphi^n_{\tau,k}$ is less or equal than the k-dimensional Betti number of M/L for any n. Q.E.D.

ACKNOWLEDGMENTS

The authors would like to thank...

REFERENCES

[1] G. Chen, K. Mischaikow, R. S. Laramee, P. Pilarczyk, and E. Zhang. Vector Field Editing and Periodic Orbit Extraction Using Morse Decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):769–785, Jul./Aug. 2007.

[2] G. Chen, K. Mischaikow, R. S. Laramee, and E. Zhang. Efficient Morse Decompositions of Vector Fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):848–862, Jul./Aug. 2008.

[3] W. de Leeuw and R. van Liere. Visualization of Global Flow Structures Using Multiple Levels of Topology. In *Data Visualization '99 (VisSym '99)*, pages 45–52. May 1999.

[4] W. de Leeuw and R. van Liere. Multi-level Topology for Flow Visualization. *Computers and Graphics*, 24(3):325–331, June 2000.

[5] W. C. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In D. Ebert, M. Gross, and B. Hamann, editors, *IEEE Visualization '99*, pages 349–354, San Francisco, 1999.

[6] C. J. A. Delfinado and H. Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes. In *SCG '93: Proceedings of the ninth annual symposium on Computational geometry*, pages 232–239, New York, NY, USA, 1993. ACM.

[7] A. Dold. *Lectures on Algebraic Topology*. Springer-Verlag Berlin Heidelberg, New York, 1980. Proposition 5.9.

[8] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-smale complexes for piecewise linear 3-manifolds. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, pages 361–370, New York, NY, USA, 2003. ACM.

[9] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical morse complexes for piecewise linear 2-manifolds. In *SCG '01: Proceedings of the seventeenth annual symposium on Computational geometry*, pages 70–79, New York, NY, USA, 2001. ACM.

[10] A. Gyulassy, P.-T. Bremer, B. Hamann, and V. Pascucci. A practical approach to morse-smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008.

[11] A. Gyulassy, V. Natarajan, V. Pascucci, and B. Hamann. Efficient computation of morse-smale complexes for three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1440–1447, 2007.

[12] J. L. Helman and L. Hesselink. Representation and Display of Vector Field Topology in Fluid Flow Data Sets. *IEEE Computer*, 22(8):27–36, August 1989.

[13] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational homology*, volume 157 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004.

[14] W. D. Kalies, K. Mischaikow, and R. C. A. M. VanderVorst. An algorithmic approach to chain recurrence. *Found. Comput. Math.*, 5(4):409–449, 2005.

[15] R. 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.

[16] R. S. Laramee, H. Hauser, H. Doleisch, F. H. Post, B. Vrolijk, and D. Weiskopf. The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. *Computer Graphics Forum*, 23(2):203–221, June 2004.

[17] R. S. Laramee, D. Weiskopf, J. Schneider, and H. Hauser. Investigating swirl and tumble flow with a comparison of visualization techniques. In *Proceedings IEEE Visualization 04*, pages 51–58, 2004.

[18] M. Mrozek. Index pairs and the fixed point index for semidynamical systems with discrete time. *Fund. Math.*, 133:178–192, 1988.

[19] M. Mrozek. Leray functor and cohomological conley index for discrete dynamical systems. *Trans. Amer. Math. Soc.*, 318:149–178, 1990.

[20] 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.

[21] G. Scheuermann, H. Hagen, H. Krüger, M. Menzel, and A. Rockwood. Visualization of Higher Order Singularities in Vector Fields. In *Proceedings of IEEE Visualization '97*, pages 67–74, Oct. 1997.

[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] A. Szymczak. *Index Pairs: from Dynamics to Combinatorics and Back*. PhD thesis, Georgia Institute of Technology, 1999.

[24] H. Theisel, C. Rössl, and H.-P. Seidel. Combining Topological Simplification and Topology Preserving Compression for 2D Vector Fields. In *Pacific Graphics*, pages 419–423, 2003.

[25] H. Theisel, C. Rössl, and H.-P. Seidel. Compression of 2D Vector Fields Under Guaranteed Topology Preservation. In *Eurographics (EG 03)*, volume 22(3) of *Computer Graphics forum*, pages 333–342, Sept. 1–6 2003.

[26] H. Theisel, T. Weinkauf, H.-P. Seidel, and H. Seidel. Grid-Independent Detection of Closed Stream Lines in 2D Vector Fields. In *Proceedings of the Conference on Vision, Modeling and Visualization 2004 (VMV 04)*, pages 421–428, Nov. 2004.

[27] Y. Tong, S. Lombeyda, A. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition. In *ACM Transactions on Graphics (SIGGRAPH 03)*, volume 22, pages 445–452, July 2003.

[28] X. Tricoche, G. Scheuermann, and H. Hagen. A topology simplification method for 2d vector fields. In *Proceedings of IEEE Visualization 2000*, pages 359–366, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.

[29] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proceedings of IEEE Visualization 2001*, pages 159–166, 2001.

[30] X. Tricoche, G. Scheuermann, and H. Hagen. Topology-Based Visualization of Time-Dependent 2D Vector Fields. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym '01)*, pages 117–126, May 28–30 2001.

[31] 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.

[32] T. Wischgoll and G. Scheuermann. Locating Closed Streamlines in 3D Vector Fields. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym 02)*, pages 227–280, May 2002.

[33] T. Wischgoll, G. Scheuermann, and H. Hagen. Tracking Closed Streamlines in Time Dependent Planar Flows. In *Proceedings of the Vision Modeling and Visualization Conference 2001 (VMV 01)*, pages 447–454, Nov. 2001.

[34] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics*, 25(4):1294–1326, 2006.