Morse Set Classification and Hierarchical Refinement using Conley Index

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

Abstract—Morse decomposition provides a numerically stable topological representation of vector fields that is crucial for their rigorous interpretation. However, Morse decomposition is not unique, and its granularity directly impacts its computational cost. In this paper, we 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 graph*, as well as the connection regions between Morse sets. The computation is fast because the most expensive computation is concentrated on a small portion of the domain. Furthermore, the present work allows the generation of a topologically consistent hierarchy of MCGs, which cannot be obtained using a global method.

The classification of the extracted Morse sets is a crucial step for the construction of the MCG, for which the *Poincaré index* is inadequate. We make use of an upper bound for the Conley index, provided by the Betti numbers of an index pair for a translation along the flow, to classify the Morse sets. This upper bound is sufficiently accurate for Morse set classification and provides supportive information for the automatic refinement process. An improved visualization technique for MCG is developed to incorporate the Conley indices. Finally, we apply the proposed techniques to a number of synthetic and real-world simulation data to demonstrate their utility.

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

1 Introduction

Numerically stable topology of vector fields is required for the rigorous interpretation of the 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 (or differential) topology of vector fields consists of special trajectories which are either points (i.e. *fixed points*), loops (i.e. *periodic orbits*), or curves (i.e. *separatrices*). Chen et al. [2] have shown that trajectory-based topology is sensitive to noise and error introduced during data acquisition and processing. To overcome this, they advocate *Morse decomposition* as a more reliable representation of vector field topology. The result of the Morse decomposition of a vector field is an acyclic directed graph called a *Morse Connection Graph (MCG)*. The nodes of this graph are *Morse sets* and the edges show the connectivity (direct flow paths) between the nodes. The Morse sets in an MCG enclose all regions of flow recurrence, in particular all periodic orbits and fixed points of the flow. The connection regions between Morse sets envelop separatrices. An MCG may provide a seemingly coarser topological structure of a vector field than the trajectory-based topology, such as a *vector field skeleton* [14] and *Entity Connection Graph (ECG)* which consists of fixed points, periodic orbits and their connectivity [1]. However, the detailed topological structure obtained in the trajectory-based topology [1], [14] could be the artifact of the numerical error or noise in the original data. Showing such detailed but unreliable topology may provide misleading information (Figure 3, left). On the other hand, the MCG tends to be more numerically stable than the trajectory-based topology. This is because both the Morse sets and connection regions of the MCG are less sensitive to noise and error than their respective counterparts in trajectory-based topology. Figure 3 shows the comparison of the stability of MCGs versus ECGs under different integration schemes in the analysis of a slice from the diesel engine simulation. We refer the readers to [1] for a more thorough discussion.

To compute an MCG, we first encode the flow dynamics into a directed graph. We refer to the graph as FG (Flow combinatorialization Graph). The nodes of an FG are the polygonal primitives of the space discretization (e.g., triangles), and the edges indicate the mapping between polygons with respect to the flow (Figure 5). The Morse sets are the strongly connected components of the FG. To capture the accurate dynamics of the flow in FG, Chen et al. introduce the idea of a $\tau-$map by tracking the image of each polygonal primitive under the translation by τ along the flow to compute the mapping between polygons (reviewed in Section 3.1). This construction is computationally expensive.

The MCG of a vector field is not unique, and the granularity

- *Guoning Chen is with Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112.*
E-mail: chengu@sci.utah.edu
- *Qingqing Deng, and Eugene Zhang are with Oregon State University, Corvallis, OR 97331.*
E-mail: {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.

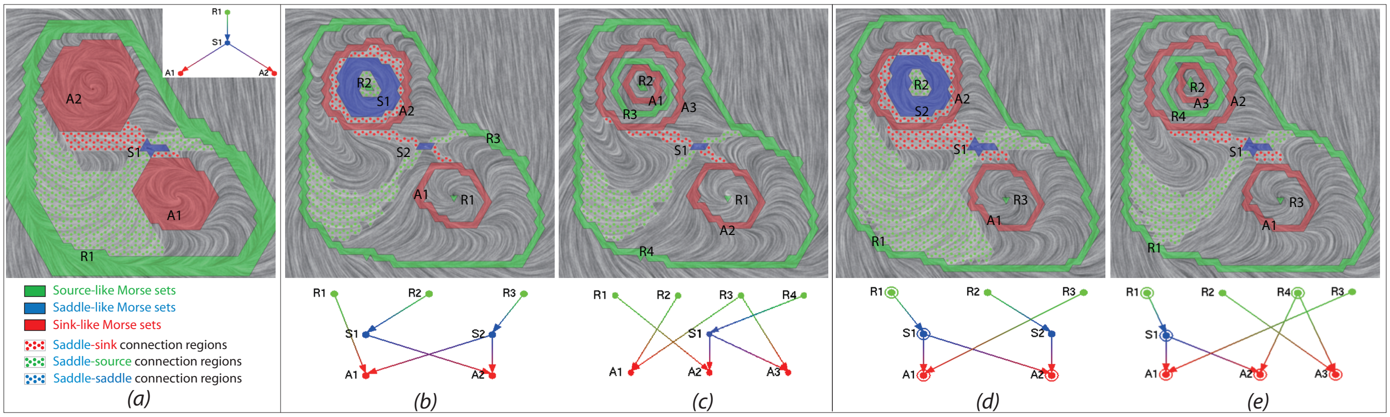


Fig. 1: The MCGs of an analytic vector field using the geometry-based method (a), the global computation ((b):$\tau = 0.1$ and (c) $\tau = 0.2$) and our hierarchical refinement framework ((d): $\tau_{max} = 0.1$ and (e) $\tau_{max} = 0.2$). Our refinement method takes 10.11 seconds for $\tau_{max} = 0.2$, while the global method uses 22.88 seconds with the same τ value. Note that our refinement method obtains comparable results to the global method but with a more coherent representation of the topology. This is because the global method re-labels all the Morse sets after each computation, while our hierarchical refinement modifies only one Morse set (one node of the graph) after each iteration starting from the MCG shown in (a). In particular, in (d) $R3$ is obtained from the refinement of $A1$ in (a), $S2$ and $R2$ are from $A2$ in (a). In (e), $R4$ and $A3$ are the results of refining $S2$ in (d). In addition, we point out that the MCG in the left visualization (a)-(c) lacks the ability to distinguish between a source-like Morse set and a periodic orbit-like source Morse set (e.g., $R1$ and $R4$). The new Morse set classification and visualization based on the Conley index provides such information.

of the MCG directly relates to its computational cost, most of which is spent on computing the image of every polygon in the domain under the induced flow for a certain time τ using the $\tau-$map approach. The larger the τ, the finer the MCG is, but also the more computationally expensive it becomes. Since an optimal τ value is typically unknown for a given flow, obtaining a high quality MCG requires several experiments using increasingly larger τ values which can make such a procedure computationally intractable in practice. Also, the results obtained by using different τ values are often hard to relate to each other because of inconsistent labeling and an unclear relationship between the respective Morse sets. Figure 1 (b) and (c) illustrate these challenges.

To address this issue, we propose an efficient Morse decomposition framework based on a hierarchical refinement process. In this framework, we first compute an MCG 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 a *local update* with increasing τ values. The Morse sets in the refined region are then incorporated into the original MCG, and the connection regions between these Morse sets and neighboring Morse sets are also refined using larger τ values. See Figure 2 for an example. This not only yields substantial acceleration of performance but also preserves topological consistency between two successive MCGs (Figure 2). Note that our refinement process does not increase the resolution of the mesh.

Another limitation of the existing Morse decomposition procedure [2] is that it does not provide adequate classification of the extracted Morse sets. This classification method uses the flow directions at the boundary of each Morse set to characterize them into source-like, sink-like, and saddle-like regions. Compared to the ECG [1] (Figure 3, left), this classification cannot distinguish a Morse set containing a repelling periodic orbit from that containing a source (e.g. $R1$ (a source) and $R4$ (a repelling periodic orbit) in Figure 1(c)).

We use the Conley index, a measure of flow complexity inside a Morse set, to obtain a more complete classification of Morse sets (Section 3.2). To reduce expensive computation associated with the Conley index [15], we propose a simple and efficient technique to compute an upper bound of the Conley index given a Morse set. Our experiments show that this upper bound coincides with the Conley index for most Morse sets in the example datasets shown in Section 6. The MCGs augmented with the Conley index information for each Morse set are shown in Figures 1 (d)(e) and 3 (right). This visualization allows the user to distinguish between Morse sets containing different flow features (e.g., $R1$ and $R4$ in Figure 1 (e)). Beside Morse set classification, the Conley index can also be used to guide the aforementioned refinement process (Section 7) by giving higher priority to Morse sets that contain complicated dynamics.

Note that during the initial stage of our framework, some Morse sets from the geometry-based method may contain multiple flow features but with a trivial Conley index since the flow features cancel each other and cause the whole Morse set to behave like a regular flow (i.e. featureless). To overcome the possible loss of features, we retain all the Morse sets consisting of more than one triangle even though they may have a trivial index. This guarantees no loss of features at the beginning of the refinement because a single triangle with the trivial Conley index does not contain any flow recurrent feature (due to the linear constraints of our problem), and thus, can be ignored.

In what follows, we first review the related work on vector field topology in Section 2. Section 3 provides the

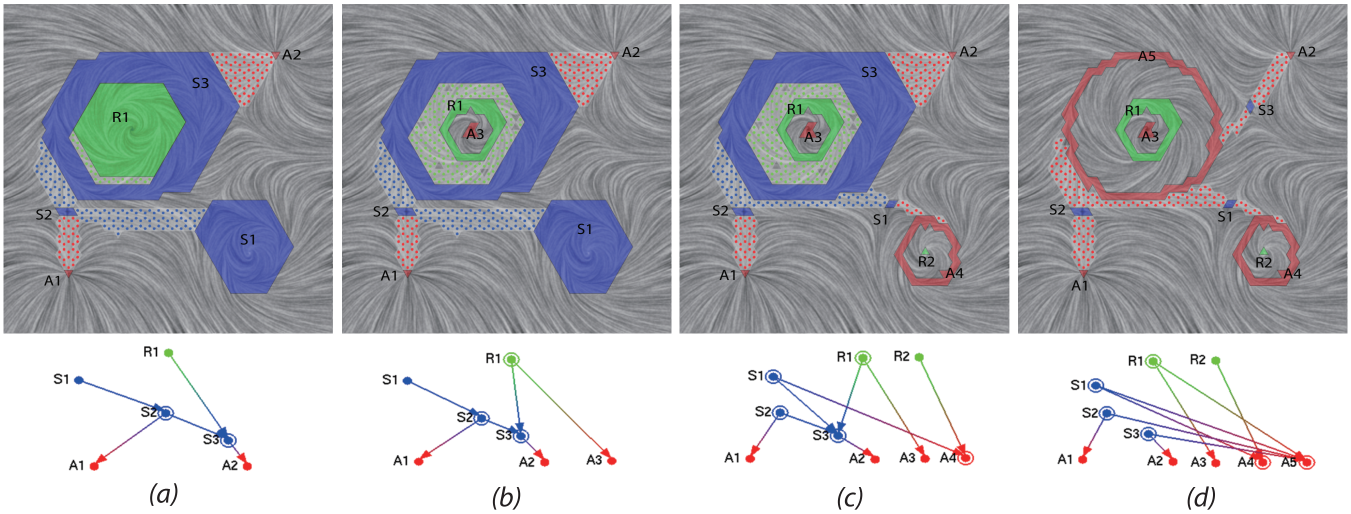


Fig. 2: An example of hierarchical Morse decomposition of a vector field: (a) MCG obtained from the geometry-based method [1], (b) local refinement of R1 in (a) with $\tau = 0.05$, (c) local refinement of S1 in (b) with $\tau = 0.1$, and (d) local refinement of S3 in (c) with $\tau = 0.1$. The meaning of the color coding is provided in Figures 1 and 3. Note that the connection regions (dotted regions) are also refined during the process. The corresponding MCGs are provided in the bottom row. The information of the Conley index of each Morse set is also visualized (a)-(d). This provides the user with 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.

background on Morse decomposition and the Conley index. In Section 4, we introduce our pipeline of hierarchical Morse decomposition. Section 5 provides the detail of local updates needed by the Morse set and connection region refinements. In Section 6 we describe an efficient algorithm to compute an upper bound on the Conley index of a Morse set. In Section 7, this upper bound is used (together with the areas of Morse sets) to control the Morse decomposition refinement process. The proposed framework has been applied to a number of synthetic and engine simulation datasets from industry. 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

Topological analysis of 2D vector fields has received much attention since its introduction to the visualization community by Helman and Hesselink [14]. In terms of fixed point extraction, Tricoche et al. [35] and Polthier and Preuß [23] present efficient algorithms to locate fixed points in a vector field. Scheuermann et al. extend the work on first-order fixed points to higher-order fixed point analysis using *Clifford algebra* and present solutions to higher-order fixed point visualization [26], [27]. 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 [37]. This technique has also been extended to 3D vector fields [38] and time-dependent flows [39]. Theisel et al. [32] present a mesh independent approach to compute periodic orbits. Recently, Chen et al. [1] present efficient algorithms to construct a more complete topological skeleton of vector fields, the *Entity Connection Graph (ECG)* by incorporating periodic orbits. Later, Chen et al. [2] study the instability of trajectory-based vector field topology and, for the first time, propose Morse decomposition for vector field topology computation which leads to more reliable interpretation of vector field topology. Reininghaus et al. [25] apply the combinatorial theory of Forman [10] to study the topology of 2D flows. The combinatorial topology is defined as the extrema nodes and the paths that connect them in a simplicial graph (i.e. a combinatorial vector field) converted from the original flow. In contrast, Morse decomposition studies a directed graph encoding the flow mapping between polygonal elements, which is the outer approximation of the original flow. This guarantees that the MCG not only contains the more accurate topology but also tolerates a certain amount of error and uncertainty of the data. While the present work is closely related to Chen et al.'s work on Morse decomposition, it is distinct in its hierarchical refinement framework for Morse decomposition using local computation that enables faster computation and consistent topology refinement. Furthermore, we provide an efficient algorithm to estimate the Conley index for Morse sets, which is not discussed in [2].

A different approach to Morse decomposition is presented in [29]. In their method, the input vector field is first approximated by a piecewise constant (PC) one, i.e. constant in the interior of every triangle. The Morse decomposition is computed from the transition graph that represents the trajectories of the PC vector field. While the PC-based analysis is more efficient and produces Morse sets of sub-triangle precision, trajectories of the PC approximation represent a relatively poor approximation of the true trajectories of the original system, roughly corresponding to Euler's method with step sizes proportional to the grid size. In contrast, the method

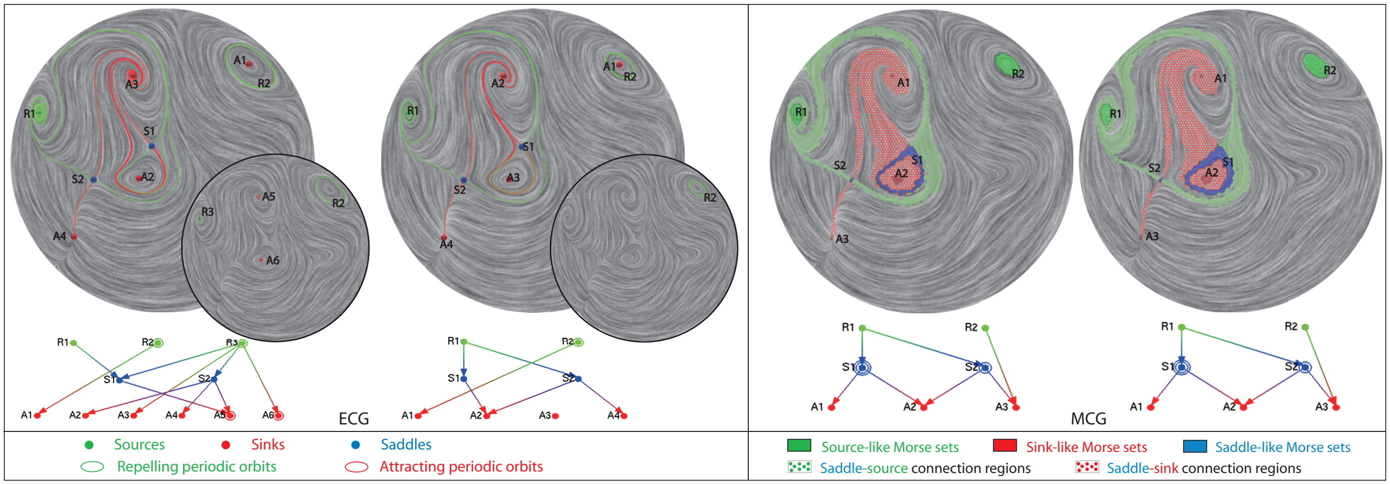


Fig. 3: This figure provides the ECGs [1] (left) and MCGs [2] (right) of a cross section of a diesel engine simulation. This cross section is cut at the 25% of the length of the cylinder from the top where the intake ports meet the chamber. The images to the left show two ECGs computed using the Runge-Kutta integration scheme [24] of the second order (RK2) and of the fourth order (RK4), respectively. The two inlets show the detected periodic orbits under different integration schemes. As can be observed, ECGs are numerically unstable and can provide misleading information. As a comparison, the MCGs of the same field are computed using RK2 (middle-right) and RK4 (right-most), respectively. Despite the different sizes of the Morse sets in the two MCGs, their topological graphs are identical, indicating the stable extraction of these features. We refer the readers to [2] for a more detailed discussion.

presented in this paper can work with any integration method and any step size. Therefore, the results can be expected to be closer to the true Morse sets for the input vector field.

2.2 Morse-Smale Complex

Morse theory has been introduced by Edelsbrunner et al. [9], [8] for scalar field topology. 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 curl-free 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 the topological skeleton of vector fields [14] where the obtained cells are known as a *basin*. Recent work on the Morse-Smale complex for the analysis of scalar fields can be found in [12], [13]. 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 3). The focus is the reliable identification of flow recurrence. In contrast to a Morse-Smale complex which addresses scalar fields (curl-free fields), Morse decomposition handles general vector fields.

2.3 Multi-scale Processing of Vector Fields

The presented work can be considered as a means of multi-scale processing of vector fields, an active research topic. There are two directions in multi-scale 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 topological simplification in visualization was done by De Leeuw and Van Liere [4]. They make use of a distance metric to determine the pair of fixed points to be cancelled. In follow-up work, they perform topological simplification based on an area metric [6]. These techniques are applied to two important applications from vector field simulation [5]. Tricoche et al. [34] 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 [19] may be combined with the visualization result. This method is later extended to time-dependent 2D flows [36]. Tricoche et al. [35] also present a topological simplification method very similar to De Leeuw and Van Liere [6]. However, simplifications are achieved by actually modifying the vectors of the original underlying data field. Theisel et al. [31] present an algorithm for compressing vector fields while preserving their topology. Later, they combine both topological simplification and topology preserving compression techniques [30]. Tong et al. [33] propose multi-scale decomposition of a vector field using Hodge decomposition and then smooth each component independently before summing them. Zhang et al. [40] 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 scenario. For an overview of related work on vector field topology, see Laramee et al. [17].

3 Background

In this section, we review the relevant concepts related to Morse decomposition for completeness.

3.1 Morse Decomposition and Morse Connection Graph (MCG)

Consider a vector field V on a manifold $\mathscr{M}$ whose solution defines a map $\varphi : \mathbb{R} \times \mathscr{M} \to \mathscr{M}$. A trajectory through a point $x_0 \in \mathscr{M}$ is a curve on $\mathscr{M}$ that is obtained by solving the initial value problem $\dot{x} = V(x)$, $x(0) = x_0$. A Morse decomposition is a collection of disjoint closed sets (called Morse sets) that

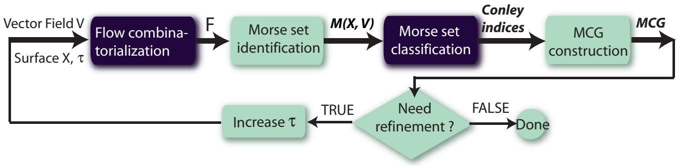


Fig. 4: The MCG computation pipeline using Morse decomposition. Note that the modules colored in black are the focus in this paper. More specifically, we introduce the idea of local flow combinatorialization and provide an efficient algorithm for the computation of an upper bound of the Conley index.

together contain all the recurrent dynamics of the flow induced by the vector field. More precisely, sets M_i, $i \in \{1, 2, \ldots, N\}$ form a Morse decomposition if and only if the trajectory of any point is either (i) entirely contained in one of the Morse sets or (ii) contained in some Morse set M_i for large enough negative times and in some other Morse set M_j, with $j > i$, for large enough positive times. Intuitively, (ii) means that the trajectory of any point outside the Morse sets can only move from a set with lower subscript to a set with a higher subscript. (ii) excludes any recurrent dynamics outside the Morse sets, making it gradient-like [3]. In practice, the partial order between Morse sets can be represented as an acyclic directed graph called a *Morse connection graph*, or MCG.

An indexing of Morse sets consistent with the above definition can be obtained from the MCG by means of topological sort. For the example shown in Figure 2 (a), the Morse sets forming a Morse decomposition can be $M_1 = R_1$, $M_2 = S_1$, $M_3 = S_2$, $M_4 = S_3$, $M_5 = A_2$, $M_6 = A_1$. Clearly, the MCG contains more restrictions on connecting trajectories than the resulting sequence (linear ordering) of Morse sets and therefore carries more information about the structure of the flow.

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 [35]. On curved surfaces, we borrow the interpolation scheme of Zhang et al. [40], 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 first converted into a directed graph (i.e. FG), denoted by $\mathscr{F}$, through flow combinatorialization. The nodes of $\mathscr{F}$ are the individual triangles of the mesh where the vector field is defined. The directed edges of $\mathscr{F}$ indicate the flow mapping relations between triangles. For instance, if there is a directed edge $T_1 \to T_2$, some particles inside triangle T_1 can reach T_2 following the flow. In other 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 flow mapping (directed edges) between neighboring triangles by considering the flow behavior across each triangle edge. In contrast, the $\tau-$map approach keeps track of the image of each triangle over a constant time τ to obtain the flow mapping. Figure 5 illustrates these two approaches. The red curved closure (Figure 5 right) illustrates the real image $\mathscr{I}_T$ of T. The set of triangles that intersect with $\mathscr{I}_T$ is referred to as the *outer approximation* of this image.

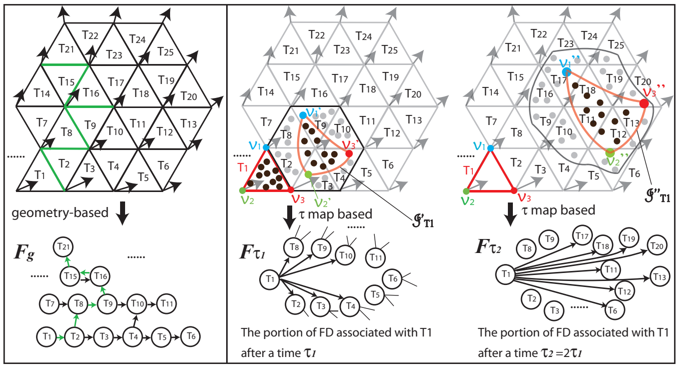

T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18 T19 T20 T21 T22 T23 T24 T25
......
geometry-based
F_g
v1
v2
v3
v1'
v2'
v3'
τ map based
$\mathscr{I}_{T1}$
F_{τ_1}
The portion of FD associated with T1 after a time τ1
$\mathscr{I}''_{T1}$
F_{τ_2}
The portion of FD associated with T1 after a time τ2 =2τ1

Fig. 5: Two approaches to flow combinatorialization: a geometry-based approach (left) and the τ-map (middle and right). Each node in the directed graphs corresponds to a triangle of the mesh. The images of triangle T_1 in the right two figures are shown as the red closures. The set of triangles that intersect with each image of T_1 is referred to as the *outer approximation* of this image [2]. Note that the geometry-based method produces an FG which is a super set of an FG using the $\tau-$map approach. As such, the FG from a $\tau-$map approach is *finer* and more accurate than the FG of the geometry-based method. For instance, the path, colored in green, from T_1 to T_{21} in the left FG does not reflect the accurate flow map (the right FG's). The right two figures also illustrate how an increasing τ value will make a better approximation of the flow map. The brown dots (middle figure) are particles enclosed in T_1 and advected by the flow. The gray dots are particles inside the outer approximation (union of the triangles $T_2, T_3, T_4, T_8, T_9, T_{10}$) of the image of T_1 over τ_1 but outside of the true image of T_1 (the red closure). They represent the error between the outer approximation and the true image. To obtain the outer approximation of the image of T_1 over $\tau_2 = 2\tau_1$, we can use the middle FG computed using τ_1. This is equivalent to advect the particles inside the previous approximation (all the gray and brown dots) over τ_1. This leads to the dark gray closure in the right figure. However, directly tracking the image of T_1 over τ_2 produces a smaller set of triangles containing the real image of it (the red closure in the right), which has smaller error than the one containing the dark gray closure.

Second, the strongly connected components are extracted from the directed graph $\mathscr{F}$. These strongly connected components correspond to the regions enclosing flow recurrence. The Conley index of each region is computed. Those regions with non-trivial Conley indices contain the Morse sets of interest [16]. The strongly connected components consisting of more than two triangles are also considered, since they may contain multiple features that cancel each other such that the flow at the boundary acts like a regular flow. For instance, in Figure 2 (a), the Conley index of Morse set S_1 is trivial. However, it contains a saddle and a source whose total indices sum to zero.

Third, each strongly connected component corresponding to the region with either a non-trivial Conley index or more than one triangle is collapsed into a single node. This reduces the original graph into a quotient graph. From the quotient graph, the final MCG can be computed through path searching between these strongly connected components using standard graph search algorithms.

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 work. In addition, they do not show the relation between the connections (dotted regions) of Morse sets and the separatrices in the differential topology of vector fields. We will provide such discussion in Section 5.2.

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 relabels the MCG such that there is no correspondence of Morse sets between any two computations. We address this challenge by introducing an automatic local update scheme for flow combinatorialization and MCG computation.

3.2 Conley Index

The computation and visualization of MCGs requires the ability to classify the extracted Morse sets. Chen et al. simply resort to the direction of the directed edges emanating from the Morse sets to classify them [2]. The more accurate classification requires the introduction of a topological descriptor called the Conley index. In our approach the Conley index is used to classify Morse sets and identify which Morse sets need further refinement (Section 7).

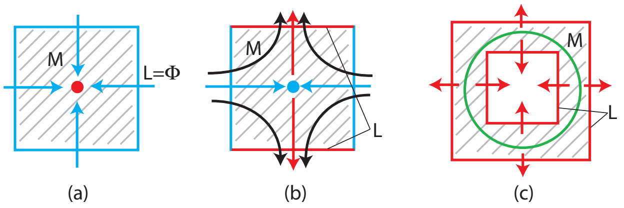


Fig. 6: A number of simple examples of isolating blocks M (shaded regions) containing (a) a sink $(1,0,0)$, (b) a saddle $(0,1,0)$, and (c) a repelling periodic orbit $(0,1,1)$. Red lines represent the exit sets L.

Computing the Conley index of a set M is particularly simple 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 (traced in the reversed direction) is outside M. Similarly, x is an exit point if its trajectory is outside M for all sufficiently small positive times (traced in the flow direction).

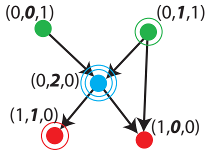
The Conley index of M can be defined as the homology of the quotient space M/L, where L is the *exit set* consisting of all exit points [15]. Intuitively, the quotient space can be obtained from M by collapsing all points in L into a single point (e.g. collapsing the two red segments of Figure 6 (b) into a point). (M,L) is an *index pair* in the sense of [3]. In what follows, we represent the Conley index of M as a sequence of Betti numbers of M/L [7]. 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, called *exit edges*. Thus, the Conley index has the form of $CH_*(M) = (\beta_0, \beta_1, \beta_2)$ since other Betti numbers are all zero based on the assumption of two-dimensional manifolds. More detailed explanation of how the Betti numbers of a quotient space are computed is provided in Section 6.

Note that for a 2D flow, β_0 and β_2 cannot be both positive. Further, $\beta_0, \beta_2 \leq 1$ for an isolated block with one connected component (e.g., one Morse set). To that end, 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, 2 and 3). In addition, a number of fundamental 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 the Conley index is a more general topological descriptor for the characterization of different flow features than the Poincaré index [35]. Particularly, the Poincaré index of M is $\beta_0 - \beta_1 + \beta_2$ [21].

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]. Since β_0, β_2 can only be either 0 or 1 and have been used to classify Morse sets, we describe how we visualize the first Betti number β_1. Particularly, we visualize it using concentric circles. For instance, if $\beta_1 = 1$, the corresponding 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 (e.g. the MCG in Figure 1(e) versus the one in 1(c)). The resulting MCGs have similar appearance as ECG with one difference being 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 circles (see the ECGs and MCGs in Figure 3).



4 Pipeline of the Hierarchical Morse Decomposition

Theory of dynamical systems shows that an *isolating neighborhood* (a polygonal region under a discrete setting) exists for

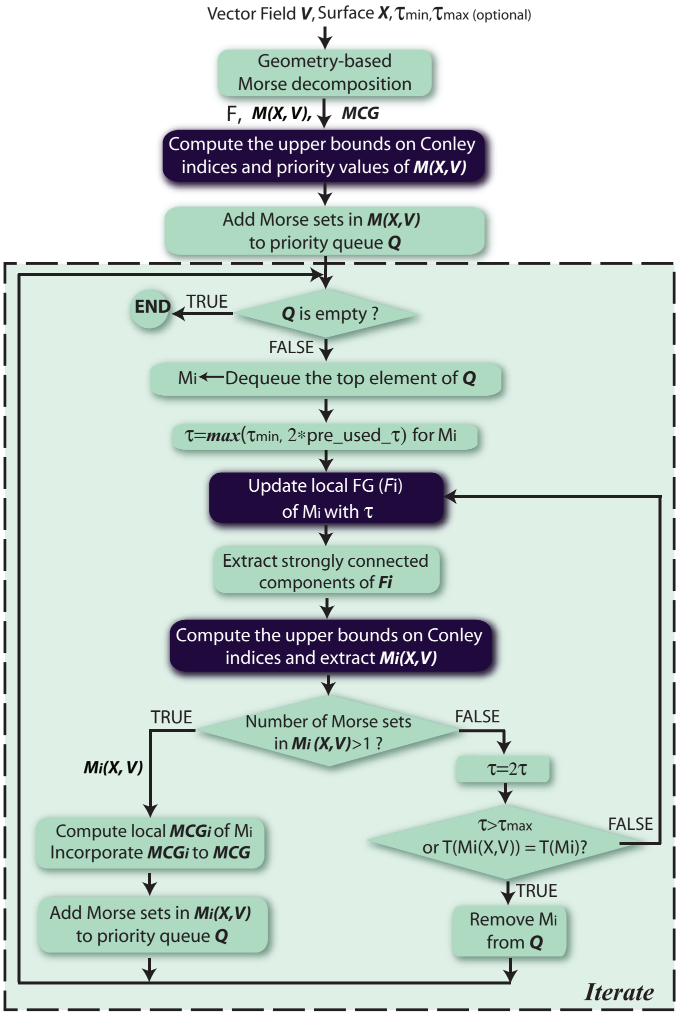

Vector Field V, Surface X, τmin, τmax (optional)
Geometry-based Morse decomposition
F, M(X,V), MCG
Compute the upper bounds on Conley indices and priority values of M(X,V)
Add Morse sets in M(X,V) to priority queue Q
END
TRUE
Q is empty ?
FALSE
Mi ← Dequeue the top element of Q
τ=max(τmin, 2*pre_used_τ) for Mi
Update local FG (Fi) of Mi with τ
Extract strongly connected components of Fi
Compute the upper bounds on Conley indices and extract Mi(X,V)
TRUE
Number of Morse sets in Mi (X,V)>1 ?
FALSE
Mi(X, V)
τ=2τ
Compute local MCGi of Mi
Incorporate MCGi to MCG
τ>τmax or T(Mi(X,V)) = T(Mi)?
FALSE
TRUE
Add Morse sets in Mi(X,V) to priority queue Q
Remove Mi from Q
Iterate

Fig. 7: 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).

each Morse set [16]. 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 spatially varying τ values. Next we describe our pipeline.

First, an MCG is computed from the FG using a geometry-based flow combinatorialization. We denote the Morse sets in this MCG as $M(X,V)$. Second, we compute an upper bound of the Conley index of each detected Morse set (Section 6) as well as a priority value based on this upper bound and the area of the Morse set (Section 7). Third, all Morse sets are placed in a priority queue Q. If Q is not empty, we enter an iterative process.

We remove the top element M_i from Q and set $\tau = \tau_{min}$ (τ_{min} can be provided by the user) or $2 \times pre\_used\_\tau$, a previously used value if M_i has been refined before. We then update the local FG, $\mathscr{F}_i$ by conducting a $\tau-$map based flow combinatorialization with τ within M_i. Next, we extract the strongly connected components of $\mathscr{F}_i$ and identify the Morse sets $M_i(X,V)$ by computing the upper bounds of the Conley indices of these components. If the number of Morse sets in $M_i(X,V)$ is larger than 1, which means the selected Morse set is refined, we construct a local MCG_i and incorporate MCG_i into MCG and add $M_i(X,V)$ into Q . 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. Otherwise, M_i can not be further refined for the given τ_{max} and will be removed from Q. This refinement process repeats until Q is empty. Note that in order to allow the refinement to exhaustively apply the values of τ up to τ_{max}, τ_{min} can be computed as $\tau_{max}/2^k$ ($k \in \mathbb{N}$).

Figure 7 illustrates this pipeline. The modules colored in black are the focus of this paper, with the implementation of local flow combinatorialization provided in Section 5, the computation of the Conley index discussed in Section 6, and the computation of the priority value introduced in Section 7. This pipeline proceeds in a hierarchical fashion and is expected to produce an MCG (Figure 1 (e)) similar to the one produced by the previous manual τ-map approach [2] with respect to a globally applied τ (Figure 1 (c)). More importantly, the intermediate and final MCGs produced using this pipeline are guaranteed to be topologically consistent. In contrast, the MCGs generated with different τ values using a global computation scheme lack such consistency (Figure 1 (b)(c)). After refining Morse sets (i.e. regions with flow recurrence), the connection regions that connect them are similarly refined (Section 5.2).

5 Local Flow Combinatorialization

The key to our hierarchical Morse decomposition approach is the ability to locally modify the flow combinatorialization. In this section, we show that locally updating the FG in a particular fashion will not affect the flow structure with respect to MCG outside of the bounded Morse neighborhood of interest. In addition, the local update scheme can be adapted to refine the connection regions (the dotted regions in Figures 1 and 2).

5.1 Refine Morse Sets

In this section, we discuss how to refine a Morse set M_i locally. First, we update the flow combinatorialization graph $\mathscr{F}$. The update procedure replaces all edges out of every triangle in M_i by edges computed using a larger τ value. The new edges are obtained using a function identical to *construct_multivaluemap* in [2], but only edges out of triangles in M_i are constructed.

Having updated the flow combinatorialization graph, we are set to refine the Morse decomposition. This is done by replacing M_i with strongly connected components of the restriction $\mathscr{F}_i$ of the refined flow combinatorialization graph $\mathscr{F}$ to M_i. The set of vertices of $\mathscr{F}_i$ consists of triangles in M_i and the set of edges contains edges of $\mathscr{F}$ that start and end at a triangle in M_i. Clearly, the resulting strongly connected components are subsets of M_i.

The refinement process works by increasing the accuracy of the flow combinatorialization graph. To illustrate this, assume that $\mathscr{F}_1$ and $\mathscr{F}_2$ are flow combinatorializations for τ and 2τ

(respectively). For a given triangle Δ, both graphs provide an upper bound (i.e. the outer approximation) on the image D of the triangle under translation by time 2τ along the flow. To obtain an upper bound on D from $\mathscr{F}_1$, one can follow two-edge paths from Δ in $\mathscr{F}_1$ and union the triangles at the endpoints of these paths. The upper bound provided by $\mathscr{F}_2$ is the union of all endpoints of edges in $\mathscr{F}_2$ starting at Δ. The later bound is generally better (i.e. smaller) than the former (see Figure 5, middle and right, for an illustration).

Figure 2 provides an example of the MCG refinement using our local updating scheme. We start from an MCG (left) which is computed from a flow combinatorialization using the geometry-based method. Next, we perform a local update inside the extracted Morse sets with larger τ values. 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.

5.2 Refine Connection Regions

After refining the Morse sets, we refine the connection regions between the newly created Morse sets and their original neighbors in the MCG before refinement. We first review the computation of connection regions given an FG.

5.2.1 Compute Connection Regions

We are interested in the connections starting from saddle-like Morse sets. Consider an FG and its Morse sets $M(X,V)$. If $M_j \in M(X,V)$ is a saddle-like Morse set, we grow a region R^+ from it following the outgoing edges of FG until it reaches another Morse set $M_k (k \neq j)$. Next, we grow a region R^- from M_k following the outgoing edges of $-FG$ (the inverse graph of FG). $R = R^+ \cap R^-$ is a connection region containing all the paths connecting M_j and M_k. This process continues until we have identified all the connection regions starting from M_j. A similar process is conducted to take care of the other (both outgoing and incoming) connection regions from M_j.

5.2.2 Connection Region Refinement

While the connection regions can be shown to contain all connecting trajectories between two Morse sets (Section 10.2), they tend to be large if computed using the geometry-based method or a small value of τ, similar to a Morse set generated using the geometry-based method or a small τ value. Therefore, we present a refinement technique to improve the precision of these regions.

Let R be a connection region for two Morse sets M_j and M_k. Recall that R consists of all paths from M_j to M_k in a flow combinatorialization graph FG. The edges of the FG are obtained using either the geometry-based method or the τ-map with some positive value of τ. The refinement procedure removes all edges out of nodes in $A := R \cup M_j \cup M_k$ and into nodes in R from FG. These edges are replaced with edges out of nodes in A into R computed using a larger value of τ (ideally, larger than the maximum τ used to compute a removed edge, but this is not required for the containment property proved in Section 10.2). Let FG' be the updated graph. The region connecting M_j and M_k is recomputed locally, by growing regions out of M_j and M_k within R. The refined connection region R' is the union of all paths in R connecting the two Morse sets M_j and M_k in the updated graph FG'.

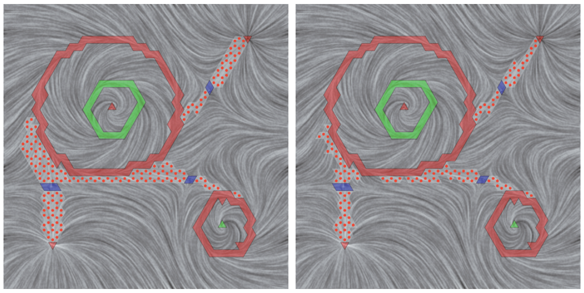


Fig. 8: This figure shows the connection regions before (left) and after refinement (right).

This connection region refinement can be performed either during the refinement of Morse sets or after all the Morse sets have been refined. We elect to do the latter in our pipeline because of the simplicity of implementation and faster computation. Figure 8 provides an example comparing the connection regions before and after refinement.

6 Conley Index Computation

As described in Section 3.2, the Conley index is easy to compute for an isolating block M. Morse sets computed using the geometry method are isolating blocks [1], but this is not true for Morse sets computed using the τ-map approach. However, an upper bound on the Conley index (called an *estimate* of the index later on), that tends to be the same as the index itself, is easy to obtain for such Morse sets.

The construction described here is based on [28]. Let L' be a union of edges of M obtained as follows. In order to decide if an edge e on the boundary of M is contained in L', take a triangle T incident upon e (i.e. e is an edge of T) and outside M. Edge e is in L' if and only if there is an edge of FG that *starts* at a triangle in M and *ends* at T. Note that (M, L') is not an index pair for the flow, in the sense of Section 3.2. In particular, one cannot claim that the Conley index is the same as the Betti numbers of M/L'. However, in Section 10.1 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 9. Consider a boundary edge e_1 shared by triangles $T_1 \in M$ and $T_9 \notin M$. There is no directed edge pointing from a triangle in M into T_9. Edge e_1 is not contained in L'. Now consider edge e_3 and triangle $T_{11} \notin M$. There is a directed edge from $T_4 \in M$ to T_{11}. Hence, $e_3 \subset L'$.

We now turn to the description of the computation of the Betti numbers for M/L'. The zero-dimensional Betti number β_0' is equal to the number of connected components in M that are disjoint with L'. If M is connected, then β_0' is zero if $L' \neq \emptyset$ and 1 otherwise. β_2' is equal to the number of connected components of M whose entire boundary is contained in L'.

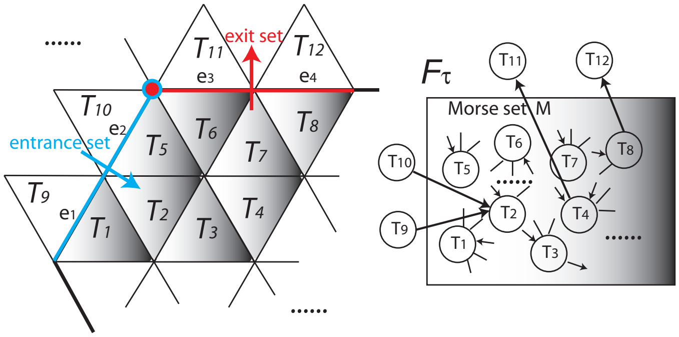
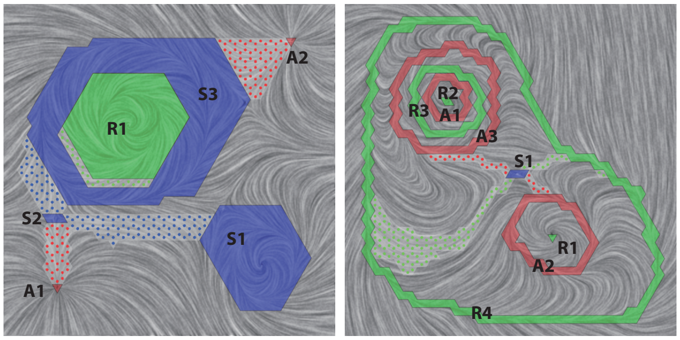


Fig. 9: This illustrates the classification of boundary edges. The image to the left provides a portion of the mesh with a Morse set M inside the shadow region. The image to the right provides the configuration of a discrete map (i.e. a flow combinatorialization) $\mathscr{F}_\tau$. Note that we ignore the inner configuration of the graph inside the Morse set M because it does not affect the classification.



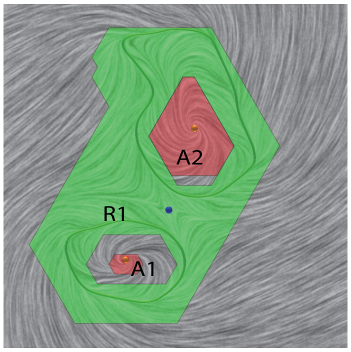
Morse set		Exit set				Upper Bound of Conley Index		
	χ(M)	β0(L′)	β1(L′)	β2(L′)	χ(L′)	β′0	β′1	β′2
R1	1	1	1	0	0	0	0	1
S1	1	1	0	0	1	0	0	0
S2	1	2	0	0	2	0	1	0
S3	0	1	0	0	1	0	1	0
A1	1	0	0	0	0	1	0	0
A2	1	0	0	0	0	1	0	0

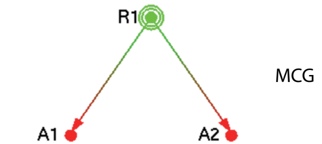
Morse set		Exit set				Upper Bound of Conley Index		
	χ(M)	β0(L′)	β1(L′)	β2(L′)	χ(L′)	β′0	β′1	β′2
R1	1	1	1	0	0	0	0	1
R2	1	1	1	0	0	0	0	1
R3	0	2	2	0	0	0	1	1
R4	0	2	2	0	0	0	1	1
S1	1	2	0	0	2	0	1	0
A1	0	0	0	0	0	1	1	0
A2	0	0	0	0	0	1	1	0
A3	0	0	0	0	0	1	1	0

Fig. 10: The computed upper bounds of the Conley indices of all Morse sets extracted from two analytical vector fields. The image to the left shows the results using a geometry-based method, while the image to the right provides the results of an MCG derived from a τ-map with $\tau = 0.2$. 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.

By [7], $\chi(M/L') = \beta_0' - \beta_1' + \beta_2' = \chi(M) - \chi(L')$, whereby $\chi(X)$ we mean the Euler characteristic of X. $\chi(M)$ ($\chi(L')$) is equal to the number of triangles minus number of edges plus number of vertices in M (respectively, L') and therefore is easy to compute. Since β_0' and β_2' are already known, β_1' can be determined from the equation $\beta_0' - \beta_1' + \beta_2' = \chi(M) - \chi(L')$.

Examples of upper bounds for the synthetic vector fields are shown in Figures 10 and 11. Note that for both datasets, the upper bound is the same as the Conley index for each Morse set.





Morse set		Exit set				Upper Bound of Conley Index		
	χ(M)	β0(L′)	β1(L′)	β2(L′)	χ(L′)	β′0	β′1	β′2
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

χ(M): Euler characteristic of M
χ(L′): Euler characteristic of boundary exit set L′
β0(L′), β1(L′), and β2(L′) are the Betti numbers for L′
β0, β1, β2 are the Betti numbers of the upper bound

Fig. 11: 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 refined further.

7 Morse Set Identification for Further Refinement

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

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

A motivation for this metric is that the more complex indices typically indicate complex flow characteristics which require further refinement (e.g., Figure 11, R_1).

The topology metric alone 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, a finer MCG reveals that it contains a saddle and an attracting periodic orbit and therefore should be refined. To handle such cases, we make use of a *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 (therefore larger area) may contain more detailed dynamics. For instance, Figure 12 (leftmost) shows the result of the Morse decomposition of the gas engine simulation 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. Further refining this Morse set reveals more detailed structure.

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

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

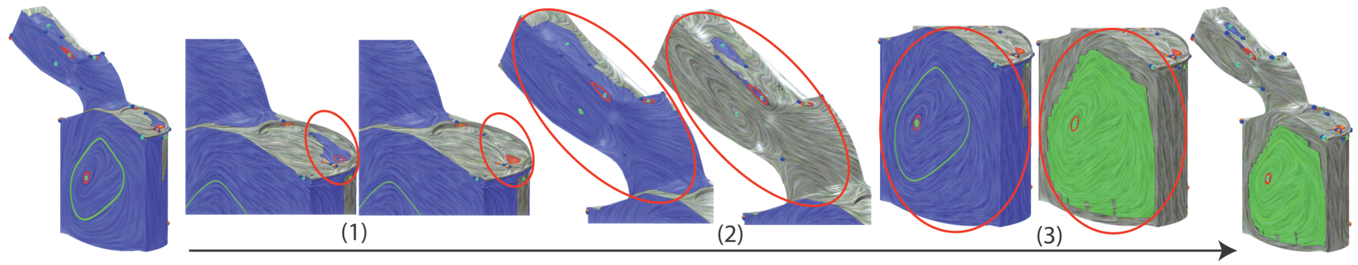


Fig. 12: This figure illustrates the refinement process of the MCG from a gas engine simulation. Left-most shows the MCG of a geometry-based approach. From (1)-(3) we refine the circulated Morse set with $\tau = 0.2$, 0.4, and 0.4, respectively. The MCG obtained using a global $\tau = 0.4$ shows the similar result (right-most). The color coding is provided in Figure 1.

This priority value is used to determine the order of the refinement of Morse sets in the current MCG. The larger the value, the earlier the Morse set will be refined. Morse sets containing one 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 $R3$ and $R4$ in Figure 10 (right)). In this case, further refinement will discover that no more Morse sets can be extracted. The system then removes these Morse sets from the priority queue Q (Figure 7). Note that once local refinement has been applied to a Morse set without success (until τ_{max} or a user specified maximum number of trials), it will not be processed again. This rule ensures the refinement process terminates even if $\tau_{\text{max}} = \infty$ (see Section 8.3).

Figure 12 provides the result of the consecutive refinement for a gas engine simulation. Note that the combination of the topology and geometry metrics in Eq. 1 may not be intuitive. However, it works well for all the examples we have examined. Exploring a better combination is a possible direction for future work.

8 Applications

We have applied our automatic hierarchical refinement framework to a number of analytic and real-world simulation datasets. In what follows, we provide and discuss the results.

8.1 Results

Figures 1 (d) and (e) provide the results of an analytic dataset. This planar dataset consists of $6,144$ triangles. Our experiment takes 10.11 seconds to return the result given $\tau_{max} = 0.2$. The global method with $\tau = 0.2$ takes 22.88 seconds to compute. Note that both methods return the same MCG and similar Morse sets with the difference of a few featureless triangles (i.e. triangles with gradient-like flow).

Although the performance gain on planar data is about a factor of 3, we have observed better performance in the analysis of the simulation data (see Table 1). Figures 13, 14, and 15 provide the analysis results of the present method and the global method as the comparison for a gas engine, diesel engine, and cooling jacket simulations, respectively. All these data are the extrapolated boundary velocity fields that are obtained through a 3D simulation inside the respective models [18], [20]. In particular, the gas engine simulation dataset consists of $26,298$ triangles. Our hierarchical refinement process takes 65.97 seconds to analyze, while the global method takes 273.2 seconds. They both use $\tau_{max} = 0.4$. Note that both analyses successfully extract the circular pattern at the back of the cylinder of the engine. However, more detailed information will need a larger τ value. For illustration, we decompose the refinement in steps as shown in Figure 12. From (1)-(3) we refine the circulated Morse set with $\tau = 0.2$, 0.4, and 0.4, respectively.

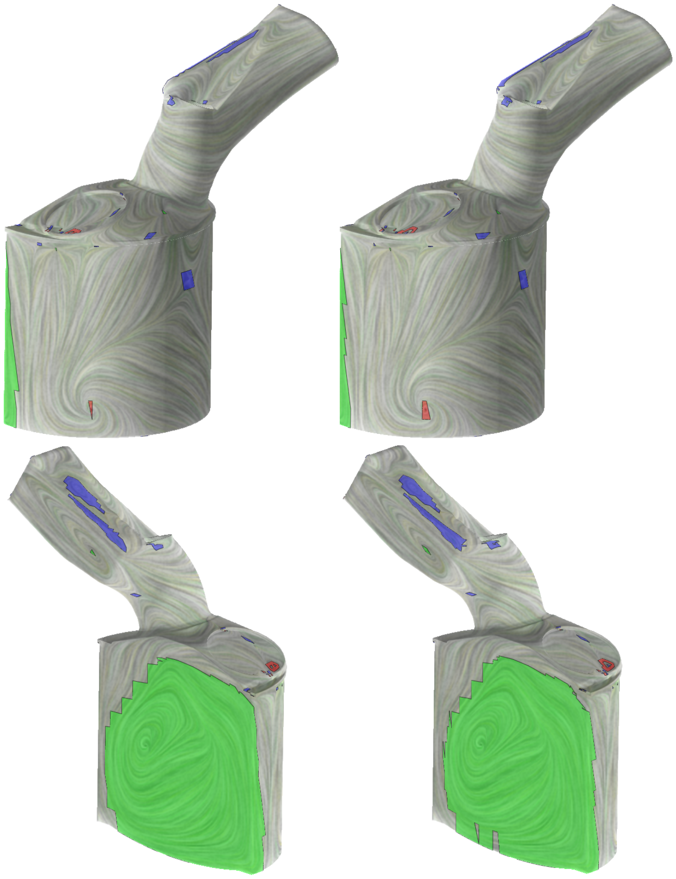


Fig. 13: This figure shows the results of the Morse decompositions of a gas engine simulation dataset using a global method (left column), and the present automatic refinement (right column). Both methods employ $\tau_{max} = 0.4$. Note that both methods produce comparable decomposition with a slight difference of the shapes and sizes of the obtained Morse sets. The color coding is provided in Figure 1.

The diesel engine simulation looks for patterns of the

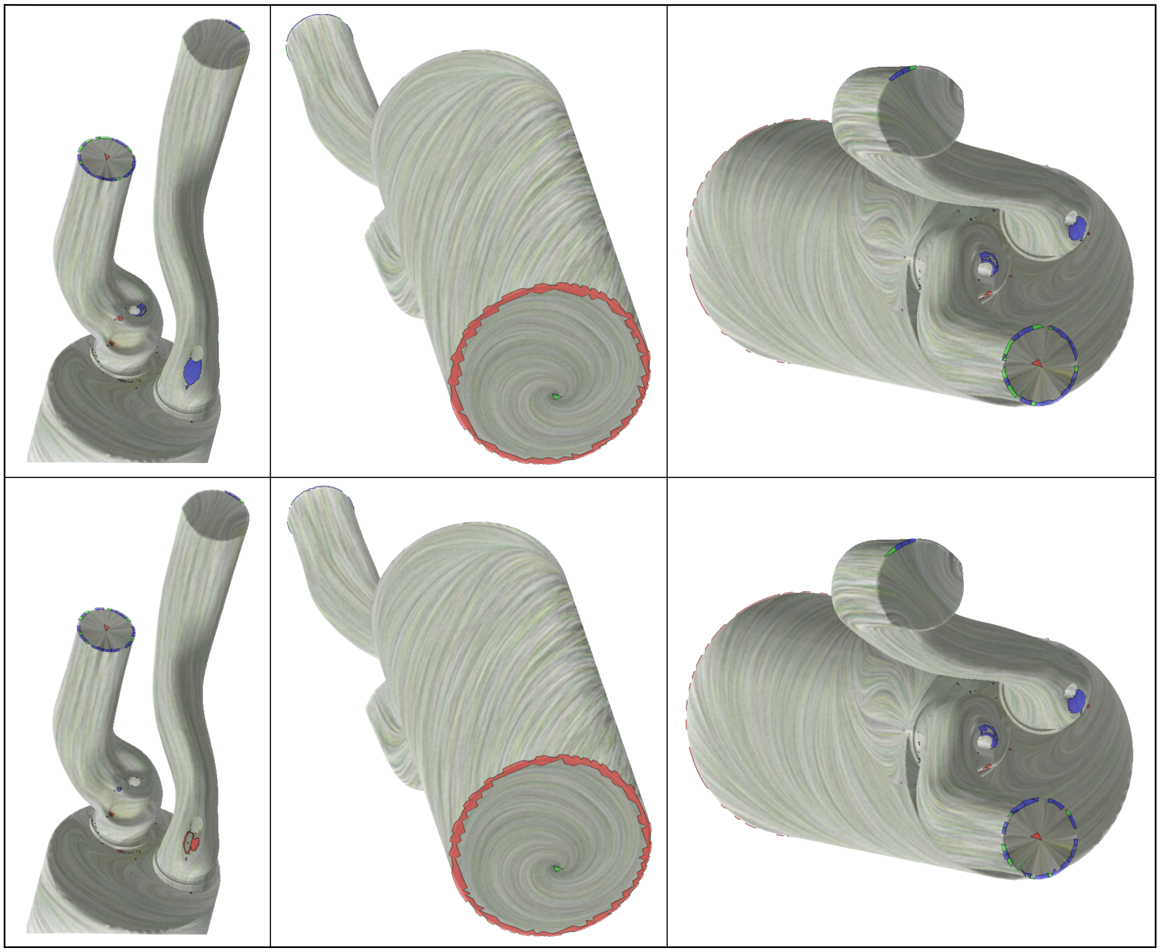


Fig. 14: This figure compares the results of the Morse decompositions of a diesel engine simulation dataset using the present automatic refinement (top) with the global method (bottom). Both analysis uses $\tau_{max} = 0.4$. The color coding is provided in Figure 1.

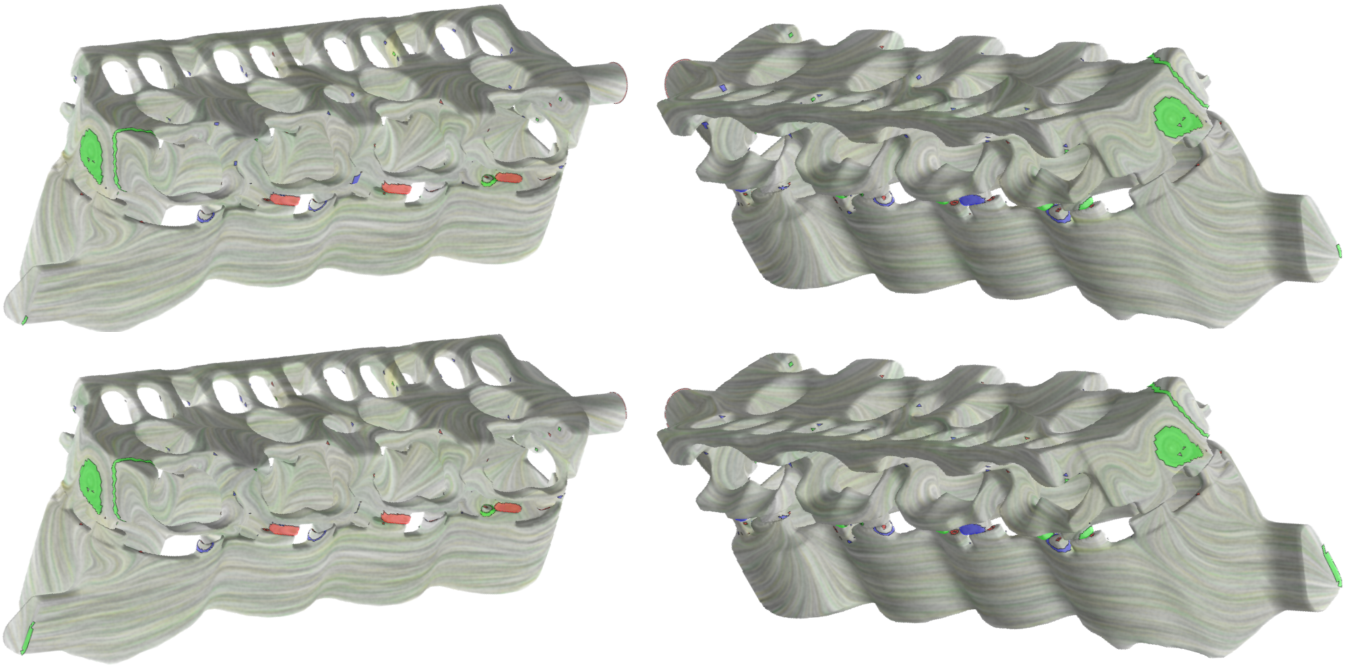


Fig. 15: The results on a cooling jacket simulation using the hierarchical refinement framework (top) and the global method (bottom). Both computations use $\tau_{max} = 0.4$. The color coding is provided in Figure 1.

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 PC with Intel(R) Xeon(R) 2.33GHz dual processors and 8GB RAM. Note that we compare only the performance of the automatic refinement framework with the global τ approach with the $\tau = \tau_{max}$. Note that the timing does not include the time for connection region refinement.

Dataset	#	Fig.	Global Update			Local Update			Speed-Up
name	polygons		τ	#Morse sets	time(s)	τ_{max}	#Morse sets	time(s)	Factor
gas engine	26,298	13	0.4	51	273.2	0.4	51	65.97	4.14
diesel engine	221,574	14	0.4	193	991.1	0.4	192	96.30	10.29
cooling jacket	227,868	15	0.4	246	1551.3	0.4	244	435.2	3.56
planar data I	6,144	1	0.2	7	22.88	0.2	7	10.11	2.26
planar data II	6,144	2	0.2	10	17.61	0.2	10	5.52	3.19

combustion flow rotating around the axis of the cylinder [1], [20]. Both our hierarchical analysis and the global method can correctly identify this pattern at the bottom of the engine (middle column of Figure 14). However, our method takes only 96.30 seconds for this dataset with $221,574$ triangles, compared to the 991.1 seconds using a global method ($\tau = 0.4$ for both methods).

The cooling jacket simulation dataset possesses $227,868$ triangles and a complex geometry. Figure 15 provides the results using our hierarchical refinement framework and a global τ-map approach. We can see that most of the extracted features (the colored regions) are regions of the geometry above and below the cooling jacket gasket (between the cylinder block and head). This is where the flow exhibits the most complex behavior [18]. These regions exhibit a number of swirling flow patterns which are detrimental to effective heat transfer away from the engine block. Both the hierarchical refinement and the global computation return similar results. However, the former takes 435.2 seconds while the latter takes 1551.3 seconds.

Table 1 provides the timing information of the automatic refinement of the MCGs (connection region refinement is not included) of the data used in this paper. 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. Regardless, our automatic refinement framework exhibits better performance time. Note that all times are measured on a PC with Intel(R) Xeon(R) 2.33GHz dual processors and 8GB RAM. To achieve fast computation, the first-order Euler integrator has been employed for all FG computations in Table 1. In addition, for all tests in Table 1 $\tau_{min} = 0.1$ is used to start the refinement. From these results, we see that our method improves the analysis with up to one order of magnitude speed-up for the real simulation data.

8.2 Discussion

Global Method vs. Adaptive Framework

We have observed some discrepancy between the global method and the presented adaptive refinement in some of the results, although they are generally comparable. This small discrepancy between the global τ MCG and hierarchical MCG is caused by the difference of the computed FG's. The hierarchical refinement uses adaptive τ values over the domain, while the global method uses a uniform τ. Also, the numerical inaccuracy incurred during the computation of the samples along edges for estimating the outer approximation, especially in a flow with high divergence or stretching, contributes to this difference [2]. Addressing this issue is one of our future work directions.

Despite this small discrepancy, the adaptive refinement is better. It is faster, more versatile, and more supportive of the resulting Morse decomposition's complexity. Therefore, it is fully capable of performing topologically consistent multi-scale visualization.

Performance Analysis

From the experiments, we have observed different performance gains for different data (see Table 1), due to the varying complexity of the flows. For instance, if the flow contains highly rotational (swirling) behavior, the obtained Morse sets with a smaller τ value can correspond to relatively large regions in terms of area (e.g., the back of the cylinder of the gas engine simulation in Figure 12). It is likely that more computation is needed in order to refine them. On the other hand, if the flow is mostly gradient-like (curl-free) (e.g., the cylinder body of the diesel engine simulation in Figure 14), the computation is typically much faster.

Comparison between MCGs and ECGs

Figures 3 and 16 respectively provide the comparison of the MCGs and ECGs of the diesel engine and gas engine simulations. To evaluate the stability of both methods, different integration schemes were employed to compute the ECGs and MCGs. From the results, we see that the ECG computation returns different topological structures of the same field, highlighted by the different extractions of periodic orbits in both examples. On the other hand, MCGs provide relatively stable structures under different integration schemes. This demonstrates that MCGs are a more reliable tool than ECGs in the analysis of flow structure.

Although MCGs need not extract as detailed information as the trajectory-based topology, we point out that in some engineering applications, identifying the regions with certain flow behavior is more important than computing the exact detailed structure. For instance, in engine design, determining whether the simulated combustion forms the ideal flow patterns is crucial for the evaluation of the design quality of

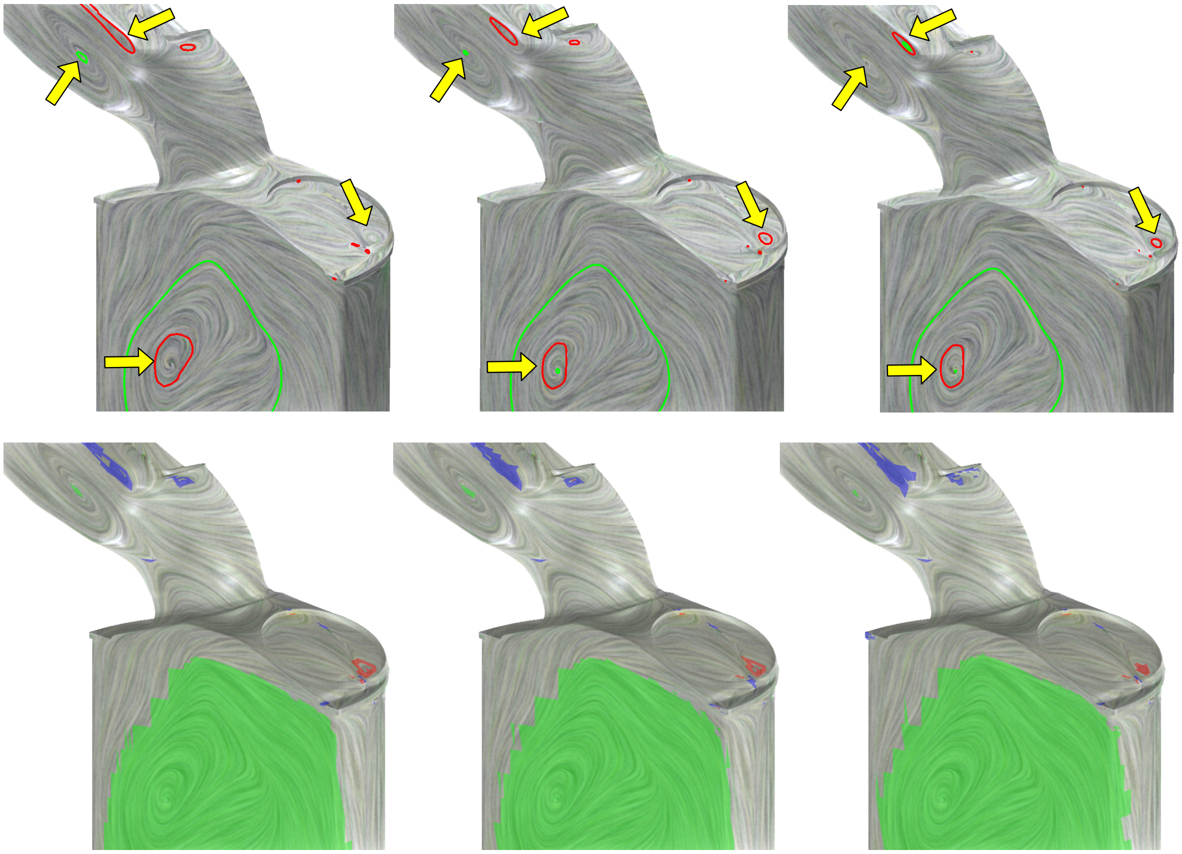


Fig. 16: This figure provides the ECGs and MCGs of the gas engine simulation computed using first-order Euler (left), RK2 (middle), and RK4 (right) integrators (respectively) [24]. Note that the ECGs show different structures in terms of the number of periodic orbits: 20 (Euler), 21 (RK2), and 19 (RK4). Whereas, MCGs computed with $\tau = 0.3$ show relatively stable results.

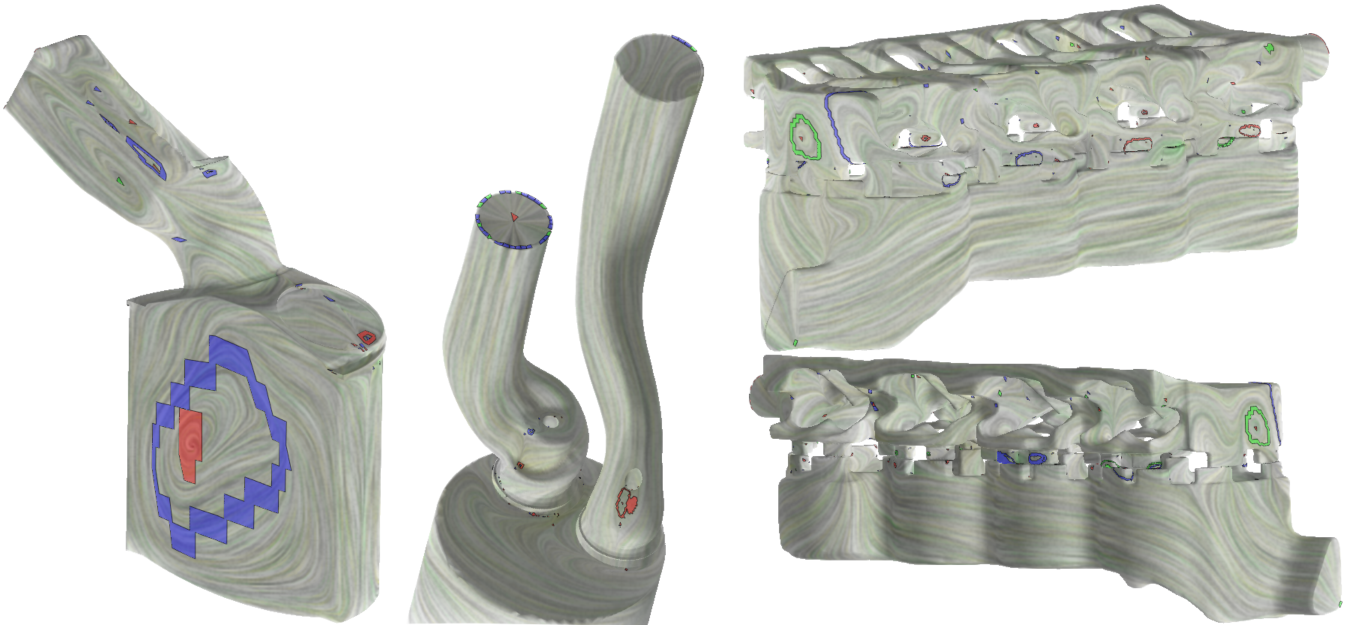


Fig. 17: This figure provides the Morse decompositions of the gas and diesel engine simulations, and cooling jacket simulation using the refinement framework without setting τ_{max}. The analysis takes about $213s$, $1012s$, and $4524s$ for these datasets, and returns MCGs with 63, 210, and 265 Morse sets, respectively.

the engine blocks. This can be achieved by identifying the desired flow behavior at the boundary geometry of the engine [1]. The Morse set at the back of the cylinder of the gas engine (Figure 12) indicates such an ideal behavior. In this case, it is more useful to know there is certain swirling flow at the back of the cylinder than to detail the number or types of

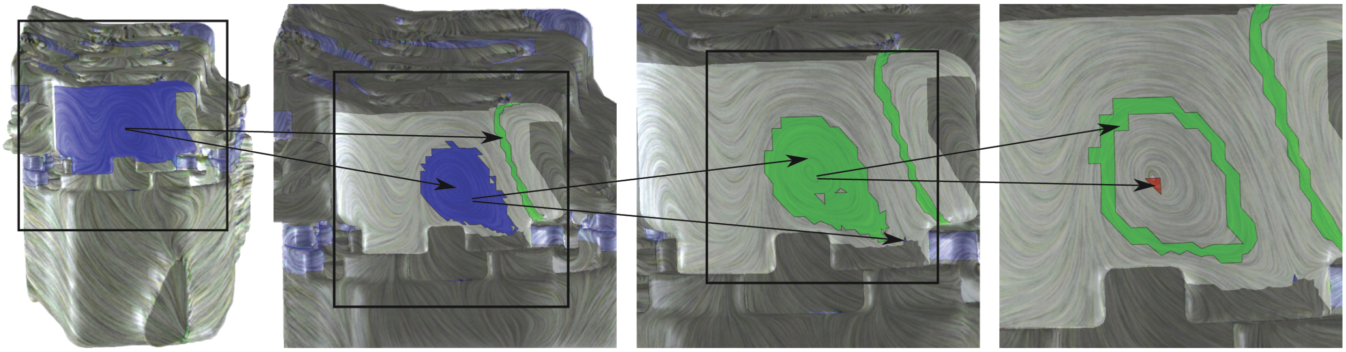


Fig. 18: The hierarchical representation of MCG on a portion of the cooling jacket. The arrows indicate the parent-children relations between the obtained Morse sets. The first MCG is computed using the geometry-based method. Then, $\tau = 0.2$, 0.4, and 0.8 are chosen to refine the corresponding Morse sets.

the periodic orbits. The same reasoning can be applied to the cooling jacket simulation where the regions of stagnant coolant flow are highlighted by the extracted Morse sets (Figure 15). We also note that if a more detailed structure is needed, the trajectory-based (differential) topology [1], [11], [20] can be extracted from the MCG with more certainty [1], [2].

8.3 Extensions

As indicated in Section 3.1, the global method requires the user to conduct a number of experiments with different τ values to obtain the optimal result. Although our previous discussion uses a τ_{max} to terminate the refinement process, this constraint is not necessary. For the further experiments, we have removed this constraint and let the refinement process continue until it converges. To guarantee convergence, we also set a threshold k for each selected Morse set. If the Morse set cannot be refined within k trials (with k increasing values of τ), it will be removed from the queue Q. In the results provided in Figure 17, we use $k = 4$. Parameter k can be increased for the extraction of more detailed structure. The maximum values of τ used for these CFD data are 3.2 (Figure 17, left), 12.8 (Figure 17, middle), and 6.4 (Figure 17, right) respectively. Such large values (compared to the previously used 0.4) are apparently difficult for the global method to predict and not necessary for the whole domain. Note that the cumulative integration errors with a large τ could compromise the accuracy of the obtained FG. Therefore, the obtained FG is no longer an outer approximation of the true dynamics, and the upper bound of the Conley index computation is no longer guaranteed, which leads to incorrect classification of Morse sets (e.g. the big blue ring-like Morse set in Figure 17, left). This is a limitation of the current Morse decomposition computation which we would like to investigate more in the future.

MCGs also provide a way to achieve hierarchical topology representation of vector fields. This hierarchy can be obtained by systematically increasing τ to compute the individual MCGs. However, the hierarchy obtained using a global method lacks a consistent topological relation between two successive levels, because each computation relabels all the Morse sets (see Figure 1, (b) and (c)). In contrast, the present framework updates the MCG locally and retains the consistent labeling, which provides the consistent hierarchy (see Figures 2 and 18). This consistent hierarchy of a vector field can assist multi-scale visualization of the flow, which we plan to investigate in future work.

9 Conclusion

In this paper, we have identified a major drawback of the previous τ-map based Morse decomposition method and proposed 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 the Conley index. The Conley index is a more general topological descriptor than the Poincaré index. In this work, we make use of the Conley index to classify the extracted Morse sets and present a combinatorial approach to compute this index. 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). This upper bound has been shown to be a good estimation to the true Conley index for all the examples used in the paper. We have proven the soundness of our hierarchical framework and provided examples of applications to the real simulation data which demonstrates the effectiveness of the framework.

The proposed framework improves the performance of Morse decompositions by up to one order of magnitude for the real simulation data. 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 multi-scale processing and visualization. We plan to investigate this in the future.

Extension to 3D and Unsteady Flow: The theory behind Morse decomposition and the Conley index is dimensionless, i.e., it is possible to extend our work to 3D steady vector fields. The implementation in 3D, however, is unlikely to be a trivial extension of 2D and will need more investigation. On the other hand, as a more rigorous form than a vector field skeleton, Morse decomposition is defined for steady vector fields. Extending it to unsteady fluid mechanics will require new findings from researchers in dynamical systems and fluid mechanics.

10 Theoretical Results

10.1 Result 1

The k-dimensional Betti number of the Conley index of a Morse set M obtained using the τ-map approach cannot be larger than the k-dimensional Betti number of M/L', where L' is determined as described in Section 6.

Proof: By the results of [28], the pair (M,L') is a valid index pair for the continuous *map* $\varphi_\tau = \varphi(\tau,.)$ (note that it is not necessarily a correct index pair for the flow). In [22], it is shown that the k-dimensional Betti number of the Conley index of the flow on M is equal to

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

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.

10.2 Result 2

Connecting trajectories between Morse sets M_j and M_k are contained in $M_j \cup M_k \cup R$, where R is the connection region

Proof: The $\alpha-$ and $\omega-$ *limit sets* of $x \in \mathscr{M}$ are closed subsets of $\mathscr{M}$ which can be defined as

$$\alpha(x) := \cap_{t<0}\mathrm{cl}(\varphi((-\infty,t),x)), \quad \omega(x) := \cap_{t>0}\mathrm{cl}(\varphi((t,\infty),x))$$

respectively. Let M_j and M_k be Morse sets and R be their connecting region (before refinement). R is the union of all paths in the FG that start in M_j and end in M_k. Take a point x_0 on the connecting trajectory between M_j and M_k, i.e. such that $\alpha(x_0) \subset M_j$ and $\omega(x_0) \subset M_k$. The trajectory of x_0 eventually enters M_k when followed forward in time. More precisely, there is a time T_0 such that $\varphi(x_0,t) \in M_k$ for all $t \geq T_0$. Let $\Delta_0, \Delta_1, \ldots, \Delta_k$ be consecutive triangles intersected by the section S of the trajectory $\varphi(x_0,t)$, with $t \in [0,T_0]$. By definition of the FG, there is an edge out of each of the triangles Δ_l to a triangle $\Delta_{l'}$ with $l' > l$ or to a triangle in M_k. To see this, assume there are no edges from Δ_l to a triangle in M_k. If the edges in FG out of Δ_l are geometry edges, one can take $l' = l+1$. Otherwise, edges out of Δ_l are obtained using the τ-map and l' is the index larger than l such that the corresponding triangle contains $\varphi(\bar{x},\tau)$, where $\bar{x}$ is the last point on the section S of the trajectory of x_0 and in Δ_l.

By the above reasoning, there is a path in FG from a triangle Δ containing x_0 to M_k. Similarly, by traversing the trajectory backward in time, one can argue that there is a path from M_j to Δ. This proves that $R \cup M_j \cup M_k$ (before refinement) contains all trajectories connecting M_j and M_k.

The statement above can be proved for regions resulting from refinement by induction on the number of refinement steps. Assume all connecting trajectories are contained in $M_j \cup M_k \cup R$. Now, update FG as described in Section 5.2. Let τ be the τ-value used in the update. Take x_0 on a connecting trajectory between M_j and M_k. Then, by following triangles containing points $\varphi(x_0,k\tau)$, $k = 0,1,\ldots$ until the first one in M_k, we obtain a path from a triangle Δ containing x_0 to M_k. Again, by reversing time, one can obtain a path from M_j to Δ. Hence Δ is in the region R' resulting from refinement of R. This proves that $M_j \cup M_k \cup R'$ contains all connecting trajectories between the two Morse sets.

Acknowledgments

We would like to thank Konstantin Mischaikow for the important discussions at the initial stage of this work. We are grateful for the valuable suggestions from Charles Hansen. We also appreciate Zhongzang Lin in helping pre-process the data and Edward Grundy and Timothy O'Keefe for proof-reading the paper. Finally, we wish to thank our anonymous reviewers for their constructive comments and suggestions. This work was supported by NSF IIS-0546881 and CCF-0830808 awards, and in part, by EPSRC research grant EP/F002335/1. Guoning Chen was partially supported by King Abdullah University of Science and Technology (KAUST) Award No. KUS-C1-016-04.

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] C. Conley. Isolated invariant sets and the Morse index. *CMBS Regional Conference Series in Mathematics*, 38, 1978.

[4] 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.

[5] W. de Leeuw and R. van Liere. Multi-level topology for flow visualization. *Computers and Graphics*, 24(3):325–331, June 2000.

[6] W. 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.

[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] R. Forman. Combinatorial vector fields and dynamical systems. *Mathematische Zeitschrift*, 228:629–681, 1998.

[11] C. Garth, R. S. Laramee, X. Tricoche, J. Schneider, and H. Hagen. Extraction and visualization of swirl and tumble motion from engine simulation data. In H. Hauser, H. Hagen, and H. Theisel, editors, *Topology-based Methods in Visualization, Mathematics and Visualization*, pages 121–135, Berlin Heidelberg, 2007. Springer.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] R. S. Laramee, C. Garth, H. Doleisch, J. Schneider, H. Hauser, and H. Hagen. Visual analysis and exploration of fluid flow in a cooling jacket. In *in Proceedings of IEEE Visualization 2005*, pages 623–630, 2005.

[19] 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.

[20] 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.

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

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

[23] 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.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[25] J. Reininghaus, C. Lowen, and I. Hotz. Fast combinatorial vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, to appear, 2010.

[26] 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.

[27] 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.

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

[29] A. Szymczak and E. Zhang. Robust Morse decompositions of piecewise constant vector fields. *IEEE Transactions on Visualization and Computer Graphics*, to appear, 2011.

[30] 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.

[31] 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.

[32] H. Theisel, T. Weinkauf, and H.-P. 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.

[33] 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.

[34] 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.

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

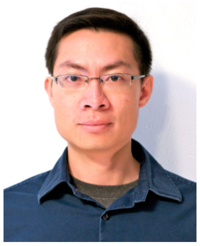
[36] X. Tricoche, G. Scheuermann, and H. Hagen. Topology-based visualization of time-dependent 2D vector fields. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym '01)*, pages 117–126, May 28–30 2001.

[37] 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.

[38] 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.

[39] 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.

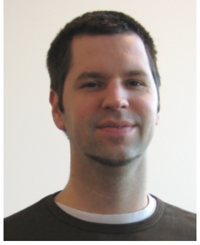
[40] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics*, 25(4):1294–1326, 2006.



Guoning Chen received a bachelors degree in 1999 from Xi'an Jiaotong University, China and a masters degree in 2002 from Guangxi University, China. In 2009, he received a PhD degree in computer science from Oregon State University. His research interests include scientific visualization, computational geometry and topology, and computer graphics. Currently, he is a post-doctoral research associate in Scientific Computing and Imaging (SCI) Institute at the University of Utah. He is a member of the IEEE.

Qingqing Deng received the bachelors degree and a masters degree in computer science from Chongqing University, China, in 2004 and 2007, respectively. She is currently working toward a PhD degree in computer science at Oregon State University. She is working with Eugene Zhang on non-photorealistic rendering and vector field visualization.

Andrzej Szymczak received the MS degree in Mathematics from the University of Gdańsk, Poland in 1994 and PhD degree in mathematics and an MS degree in computer science in 1999 from the Georgia Institute of Technology. Currently, he is an Assistant Professor in the Department of Mathematical and Computer Sciences at the Colorado School of Mines. His research interests include scientific visualization, computational topology, medical image analysis and computer graphics. He is a member of the IEEE.



Robert S. Laramee received a bachelors degree in physics, cum laude, from the University of Massachusetts, Amherst in 1997. In 2000, he received a masters degree in computer science from the University of New Hampshire, Durham. He was awarded a PhD from the Vienna University of Technology, Austria at the Institute of Computer Graphics and Algorithms in 2005. His research interests are in the areas of scientific visualization, computer graphics, and human-computer interaction. Currently he is a Senior Lecturer at the Swansea University (Prifysgol Cymru Abertawe), Wales in the Department of Computer Science (Adran Gwyddor Cyfrifiadur).

Eugene Zhang received his PhD degree in computer science in 2004 from Georgia Institute of Technology. He is currently an Associate Professor at Oregon State University, where he is a member of the School of Electrical Engineering and Computer Science. His research interests include computer graphics, scientific visualization, and geometric modeling. He received a National Science Foundation (NSF) CAREER award in 2006. He is a member of the IEEE and a senior member of the ACM.