

Multi-Scale Topological Analysis of Asymmetric Tensor Fields on Surfaces

Fariba Khan, Lawrence Roy, Eugene Zhang, Botong Qu, Shih-Hsuan Hung, Harry Yeh, Robert S. Laramée, and Yue Zhang

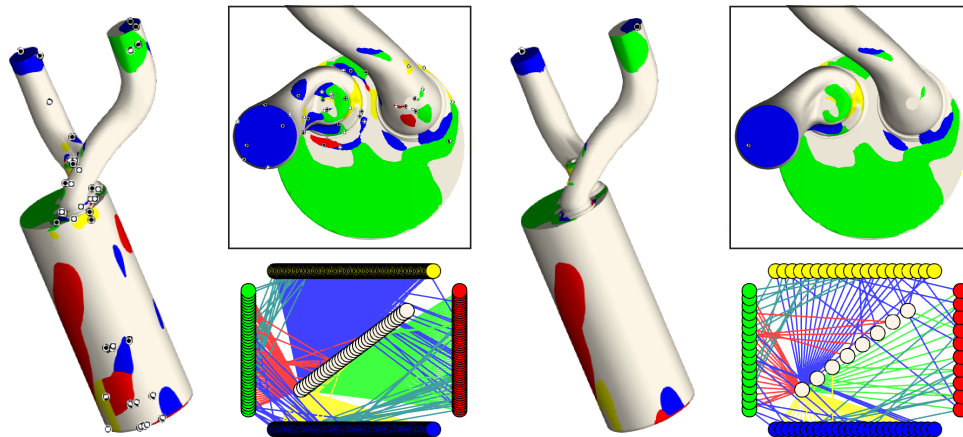


Fig. 1: The eigenvalue graph of the gradient tensor field of a simulated flow on the boundary of a diesel engine [19] (left) and the corresponding eigenvalue graph after the simplification of the gradient tensor field (right). In the original tensor field (left), there are a number of regions dominated by either expansion (colored yellow) and contraction (colored blue). However, such regions disappear after the simplification process (right), indicating that they are not as significant as regions dominated respectively by counterclockwise rotations (red), clockwise rotations (green), and pure shears (antique white).

Abstract—Asymmetric tensor fields have found applications in many science and engineering domains, such as fluid dynamics. Recent advances in the visualization and analysis of 2D asymmetric tensor fields focus on pointwise analysis of the tensor field and effective visualization metaphors such as colors, glyphs, and hyperstreamlines.

In this paper, we provide a novel multi-scale topological analysis framework for asymmetric tensor fields on surfaces. Our multi-scale framework is based on the notions of eigenvalue and eigenvector graphs. At the core of our framework are the identification of atomic operations that modify the graphs and the scale definition that guides the order in which the graphs are simplified to enable clarity and focus for the visualization of topological analysis on data of different sizes. We also provide efficient algorithms to realize these operations. Furthermore, we provide physical interpretation of these graphs.

To demonstrate the utility of our system, we apply our multi-scale analysis to data in computational fluid dynamics.

Index Terms—Tensor field visualization, tensor field topology, 2D asymmetric tensor fields, 2D asymmetric tensor field topology, eigenvalue graphs, eigenvector graphs

1 INTRODUCTION

- Fariba Khan is with Oregon State University. E-mail: khanfari@oregonstate.edu.
- Lawrence Roy is with Oregon State University. E-mail: royl@eecs.oregonstate.edu.
- Eugene Zhang is with Oregon State University. E-mail: zhange@eecs.oregonstate.edu.
- Botong Qu is with Oregon State University. E-mail: qub@oregonstate.edu.
- Shih-Hsuan Hung is with Oregon State University. E-mail: hungsh@oregonstate.edu.
- Harry Yeh is with Oregon State University. E-mail: harry@enr.oregonstate.edu.
- Robert S. Laramée is with Swansea University. E-mail: R.S.Laramee@swansea.ac.uk.
- Yue Zhang is with Oregon State University. E-mail: zhangyue@oregonstate.edu.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

Asymmetric tensor fields have found a wide range of applications such as fluid mechanics [26, 35, 36]. In these applications, a vector field plays a critical role, such as the velocity vector field of a fluid flow and the displacement vector field of an object undergoing deformation. While direct vector field visualization techniques such as glyphs [17], streamlines [19], and vector field topology [16] can provide much insight into the vector field, the gradient tensor of the vector field (an asymmetric tensor field) can provide important and complementary information [26, 35].

Existing research on asymmetric tensor fields focuses on the visualization aspect [26, 35], such as finding proper glyph representations [2, 27] to show local eigenvalue and eigenvector variations in the tensor field. There has been relatively little attention given to the topological structures in the asymmetric tensor field.

Lin et al. [21] introduce two graph-based representations for planar asymmetric tensor fields and apply them to slices of a diesel engine simulation [19]. However, their work is limited to planar datasets, at a single scale, and without physical interpretation. This makes the impact of their work rather limited as many datasets from fluid mechanics involve vector and tensor fields on curved surfaces, such as the diesel engine and cooling jacket simulations [19]. In addition, due to the complexity of and the noise in the data, the topological graphs for

simulation data can be rather complex, making it difficult for domain scientists to differentiate between important and less important features. A multi-scale analysis is needed to address these difficulties.

In this paper, we provide a multi-scale topological representation using these graphs for data in the plane or on curved surfaces. In addition, we provide efficient algorithms to extract the graphs from simulation datasets on curved surfaces. To enable a multi-scale topological representation, we identified a set of atomic operations with which nodes in the graphs (corresponding to feature regions and points in the data) can be merged. We provide algorithms to realize these atomic operations. Moreover, to enable automatic multi-scale analysis, we have defined measures with which the next pair of nodes in a given graph is selected for simplification.

In addition to the aforementioned analysis and algorithms, we also work with domain experts in fluid dynamics for the physical interpretation of our multi-scale analysis. The utility of our approach is demonstrated with a number of application datasets.

2 PREVIOUS WORK

Our research addresses the multi-scale analysis and visualization of asymmetric tensor fields. In this section, we review existing research for asymmetric tensor field visualization as well as multi-scale analysis in surface visualization and scalar field topology.

2.1 Asymmetric Tensor Field Visualization

Asymmetric tensor fields are a relatively new subject of study in tensor field visualization. In contrast, the study of symmetric tensor fields is extensive and a thorough review of symmetric tensor field visualization is beyond the scope of this paper. We refer our readers to [15, 23, 24] and references therein.

To the best of our knowledge, the first research on tensor field visualization is conducted by Delmarcelle and Hesselink [7, 8]. They extend the notion of vector field topology [16] to 2D symmetric tensor fields and provide the definition and classification of degenerate points. To deal with topological noise in the tensor field, a number of techniques have been proposed [30–32].

The research on 2D asymmetric tensor fields starts with the pioneering work of Zheng and Pang [36], who introduce the notions of *real domain* and *complex domain*. To visualize the tensor field inside the complex domain where the eigenvectors are complex-valued, they introduce the notion of *dual-eigenvectors*. They also point out the importance of *circular points*, which are degenerate points in 2D asymmetric tensor fields.

For a more systematic study of 2D asymmetric tensor fields, Zhang et al. [35] introduce the notions of *eigenvalue manifold* and *eigenvector manifold*. These notions are applied to visualize asymmetric tensor fields using a combination of glyphs and hyperstreamlines [26]. Recent research on asymmetric tensor fields has focused on the design of proper glyphs [1, 12, 28].

One of the exceptions is the work by Lin et al. [21], who propose to use two topological graphs to represent the structure in an asymmetric tensor field. Their research is exploratory in nature and is limited to planar data at a single scale without clear physical interpretation. Our research is inspired by their work, and we strive for a novel multi-scale analysis framework for the topology of asymmetric tensor fields defined on curved surfaces. In addition, we provide physical interpretation of our multi-scale topological analysis in conjunction with domain scientists.

2.2 Multi-Scale Analysis

Multi-scale visualization has been carried out in a number of areas within visualization.

There has been much work in multi-resolution mesh representation [6]. Hoppe [13] introduces the notion of progressive meshes, a data structure that allows a mesh to be inspected at different resolutions, given the distance to the viewer. He also provides smooth transition between meshes at different resolutions using the geomorphing technique. At the core of his technique is the realization that the only atomic operation needed to build the multi-resolution data structure

is the *edge collapse* operation. After this work, the focus on multi-resolution mesh generation shifts to finding the proper measure for the resolution. Garland and Heckbert [10] propose the use of the quadric measure which aims at maintaining the overall curvature of the underlying surface. This measure is later extended to account for color and texture information [11] and self-occlusion of the surface [34].

Scalar field topology is another area in which multi-scale analysis has been carried out [3, 4, 29]. The atomic operation identified for multi-scale scalar field topology is the so-called critical point cancellation, where two critical points with opposite Poincaré indexes are removed simultaneously from the field. This reduction impacts the topological structure of the scalar field, such as *Morse-Smale complexes*, *contour trees*, and *Reeb graphs*. The measure for canceling a pair is based on *persistence*, a measure proposed by Edelsbrunner et al. [9].

3 ASYMMETRIC TENSOR FIELD TOPOLOGY

In this section, we review the two graphs defined by Lin et al. [21] to describe the topology of 2D asymmetric tensor fields.

Zhang et al. [35] re-parameterize the set of 2×2 asymmetric tensors $\left\{ \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \mid T_{11}, T_{12}, T_{21}, T_{22} \in \mathbb{R} \right\}$ with

$$\gamma_d \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \gamma_r \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} + \gamma_s \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} \quad (1)$$

where $\gamma_d = \frac{T_{11}+T_{22}}{2}$, $\gamma_r = \frac{T_{21}-T_{12}}{2}$, and $\gamma_s = \frac{\sqrt{(T_{11}-T_{22})^2+(T_{12}+T_{21})^2}}{2}$ are the strengths of the isotropy, rotation, and anisotropy in the tensor, respectively. In addition, θ encodes the directions of the anisotropy.

The eigenvectors of a tensor are expressed in terms of γ_r , γ_s , and θ . In particular, whether the tensor has real-valued eigenvectors or complex-valued eigenvectors is entirely dependent on the quantity $\gamma_s^2 - \gamma_r^2$. If this quantity is positive, then the eigenvectors are real-valued. On the other hand, if the quantity is negative, then the eigenvectors are complex-valued. The former corresponds to cases where the pure shear component in the tensor is stronger than that of the rotation, while the latter is the reverse. Such a distinction is important for fluid dynamics [26, 35, 37]. In addition, Zhang et al. [35] point out that regardless of whether having real- or complex-valued eigenvalues, it is important to differentiate tensors corresponding to counterclockwise (CCW) rotations from those corresponding to clockwise (CW) rotations. This leads to a combination of four types of tensor behavior as illustrated in Figure 2 (a): $W_{r,n}$, $W_{r,s}$, $W_{c,n}$, and $W_{c,s}$. The subscripts r and c denote the real and complex domains, and n and s represent the northern and southern hemispheres, respectively. A real or complex region in the northern hemisphere has a CCW rotational flow, while a region in the southern hemisphere has a CW rotational flow. In Figure 2, $\phi = \arctan(\frac{\gamma_r}{\gamma_s})$ transitions from pure CW rotation ($\phi = -\frac{\pi}{2}$), to pure shear ($\phi = 0$), to pure CCW rotation ($\phi = \frac{\pi}{2}$).

Given a tensor field T , it introduces a map ρ from the domain of the tensor field to the eigenvector manifold. The inverse of this map leads to a classification of the tensor field into the aforementioned four types of regions. Lin et al. [21] define the eigenvector graph as follows. A node in the graph is a maximal, connected region in the domain inside which the points have the same tensor behavior, i.e. $W_{r,n}$, $W_{r,s}$, $W_{c,n}$, or $W_{c,s}$. The node is therefore labeled by this classification. Two nodes are connected by an edge if their corresponding regions share one common boundary curve. Note that not all pairs of regions can be neighbors. The only possible adjacent pairs are: (a) $W_{c,n}$ and $W_{r,n}$, (b) $W_{r,n}$ and $W_{r,s}$, as well as (c) $W_{r,s}$ and $W_{c,s}$. Consequently, no three regions with different types of behavior can share a common point.

In addition, Lin et al. [21] include degenerate points in this graph. A degenerate point is where $\gamma_s = 0$, i.e., pure rotation. A degenerate point is either a wedge or a trisector. Note that a degenerate point can only occur inside complex-valued regions, i.e., $W_{c,n}$ and $W_{c,s}$. Therefore, a node corresponding to a degenerate point can have an edge only connecting to its container region. Figure 3 illustrates this with an example tensor field.

The eigenvalues of an asymmetric tensor are expressed in terms of γ_d , γ_r , and γ_s . Note that $\gamma_s \geq 0$ while γ_r and γ_d can be either positive or

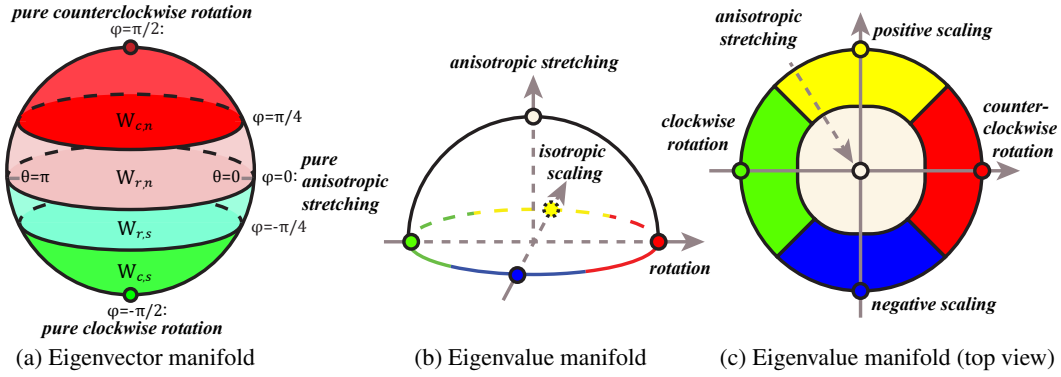


Fig. 2: This figure illustrates the notions of eigenvector manifold (a) and eigenvalue manifold (b and c), replicated from [35]. In the eigenvector manifold (a sphere), there are four types of tensor behavior: complex eigenvalues with CCW rotations (red), real eigenvalues with CCW rotations (pink), real eigenvalues with CW rotations (cyan), and complex eigenvalues with CW rotations (green). In the eigenvalue manifold (a hemisphere), there are five types of tensor behavior that show the underlying vector field is dominated by: expansion or positive scaling (yellow), contraction or negative scaling (blue), CCW rotation (red), CW rotation (green), and pure shear or anisotropic stretching (antique white).

negative. This leads to five types of tensor behavior, i.e. dominated by expansion (D^+), contraction (D^-), CCW rotation (R^+), CW rotation (D^-), and pure shear (S). The eigenvalue manifold (a hemisphere) is illustrated in Figure 2 with a side view (b) and a top view (c). Similar to the eigenvector graph, in the eigenvalue graph a node is a maximal, connected region whose dominant tensor behavior is the same for all points in the region. This leads to five types of regions. Also similar to eigenvector graphs, not all pairs of regions can be adjacent. In fact, S can be adjacent to the other four types of regions R^+ , R^- , D^+ , and D^- . Among the latter four regions, R^+ cannot be adjacent to R^- and D^+ cannot be adjacent to D^- . Unlike eigenvector graphs, in eigenvalue graphs there are often the cases where three types of regions share a common point, which Lin et al. [21] refer to as a junction point.

Note that the analysis based on the eigenvector manifold is *not* completely unrelated to that based on the eigenvalue manifold. The former is based on the interplay between the rotation and shearing components in the tensor, while the latter takes into account rotation, shearing, and isotropic scaling. Consequently, certain relationships exist between regions from the two types of manifold analysis. For example, an R^+ region indicates that γ_r is larger than both $|\gamma_d|$ and γ_s . Consequently, such a region must reside completely inside a $W_{c,n}$ region. Similarly, an R^- region must be inside a $W_{c,s}$ region. On the other hand, a shearing region indicates that γ_s is larger than both $|\gamma_d|$ and $|\gamma_r|$. Consequently, an S region must be contained by the union of $W_{r,n}$ and $W_{r,s}$. In contrast, a D^+ or D^- region can intersect both the real domain and the complex domain.

3.1 Modification and Correction

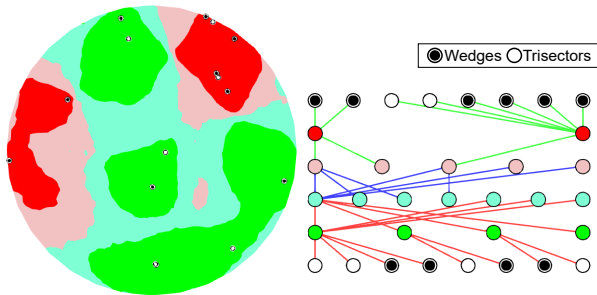


Fig. 3: This figure shows the visualization of a slice in the diesel engine simulation (left) and its corresponding eigenvector graph (right). A degenerate point can only have an edge connecting to its container region where $\gamma_s = 0$, i.e., a green or red region.

There are two places where we modify or correct the definition of the eigenvalue graph.

First, Lin et al. [21] claim that the only possible adjacency configuration in the eigenvalue graph is through junction points. That is, it is impossible to have the case where one region is enclosed by another. We have observed that this is false, i.e. it is possible for one region to be enclosed by another. The upper-right figure shows one such example (orange box) where a red region is contained entirely inside the antique white region without any junction point.

Second, we now include degenerate points in the eigenvalue graphs and note that they cannot occur in the S -type regions.

Given these new modifications and extensions, we describe our multi-scale topological analysis of asymmetric tensor fields on surfaces in the following sections.

4 TOPOLOGICAL GRAPH CONSTRUCTION

In this section we describe our algorithms to extract the eigenvector and eigenvalue graphs from a given asymmetric tensor field T defined on a triangular mesh M that is associated with an underlying mesh. Note that often T is derived as the spatial gradient of a vector field V which can be a flow velocity or displacement field in mechanics.

The input mesh M to our system consists of the set of vertices V , the set of edges E , and the set of triangles F . The asymmetric tensor field T is a 2×2 matrix defined at each vertex in V , expressed in a local coordinate system of the tangent plane at the vertex. There are two possibilities. The mesh M represents either a planar domain or a curved surface. In the first case, the tensor field is piecewise linear, i.e. inside each triangle, the tensor value at a point p with barycentric coordinates b_1 , b_2 , and b_3 will have a tensor value of $b_1 T_1 + b_2 T_2 + b_3 T_3$ where T_i ($i = 1, 2, 3$) are the tensor values at vertex i of the triangle. Note that this leads to a linear tensor field inside the triangle, i.e. each component of the tensor field T_{ij} is linear in terms of the coordinates of points in the triangle.

In the second case, i.e. when M represents a curved surface, the situation is more complicated. As pointed out in [33], linear interpolation of the tensor values inside each individual triangle can lead to field discontinuity along edges between adjacent triangles. This is due to the fact for meshes representing curved surfaces, the *Gaussian curvature* of a vertex can be non-zero. The Gaussian curvature of a vertex v_i is defined as $2\pi - \sum_{j \in W(i)} \alpha_j$ where $W(i)$ is the collection of triangles incident to vertex v_i and α_j is the angle of the corner in triangle j that is incident to v_i . Figure 4(a) shows one example of a vertex with a non-zero Gaussian curvature. For a more detailed discussion of this problem, we refer the reader to [33].

To generate a continuous vector field from tensor values defined at the vertices of M , Zhang et al. [33] propose a non-linear interpolation

scheme that guarantees the continuity of the resulting vector field. The key idea behind this non-linear interpolation scheme is to treat each vertex with a non-zero Gaussian curvature in the triangle as a line segment, parameterized by points on the edge in the same triangle opposite to the vertex. For each point on the virtual line segment (Figure 4(b)), a tensor value is given so that the tensor values along the virtual line segment are continuous. Furthermore, between adjacent triangles incident to the same vertex, their tensor values match along the common edge. As pointed out by Zhang et al. [33], for a triangle with three vertices with non-zero Gaussian curvatures, it is possible to have two singularities in a triangle, which makes it difficult to extract the singularities. To avoid this, they divide a triangle into four by placing a new vertex at the mid-point of each edge in the original triangle. Note that each of the new vertices has a zero Gaussian curvature. This leads to four new triangles, each of which has at most one vertex with a non-zero Gaussian curvature. Therefore, inside the new triangles, there can be at most one singularity which can be extracted efficiently. This interpolation scheme is later extended to 2D symmetric tensor fields [32], rotational symmetry fields [25], and 2D asymmetric tensor fields [26, 35].

However, the aforementioned non-linear interpolation scheme is *non-polynomial*, which makes it difficult to extract boundaries between regions in the eigenvector and eigenvalue graphs. To overcome this limitation, we develop a modified interpolation scheme that leads to a quadratic interpolation inside the triangles. If vertex v_0 of triangle f has a non-zero Gaussian curvature, we transfer the tensor from v_0 to its two incident edges of the triangle f , thereby, creating two points, v_{01} and v_{02} , very close to v_0 with the tensor field defined at them (see Figure 4(b) for an illustration). Then, we use a bilinear interpolation scheme to calculate interpolated tensor at p from tensors at v_{01} , v_{02} , v_1 and v_2 . If f does not have any vertex with a non-zero Gaussian curvature, we use linear interpolation inside f as described in [21].

Our interpolation scheme results in either a linear (degree one) or bilinear (degree two) polynomial asymmetric tensor field inside a triangle, where the degree of the polynomial is either two or four. To construct the eigenvector graph, we need to identify the nodes in the graph, i.e. each maximal, connected region with the same tensor behavior, as well as the edges in the graph, i.e. which pairs of regions are adjacent to each other. Both tasks require the extraction of boundaries between regions of different types. Once boundaries are extracted, regions are identified along with adjacency information. Next, we create a node for each region and connect adjacent regions' nodes with an edge to obtain our graph representation which is not achieved for curved surfaces in prior work.

4.1 Eigenvector Graph Construction

There are three types of boundaries in the eigenvector manifold: (a) between $W_{r,n}$ and $W_{r,s}$, (b) between $W_{r,n}$ and $W_{c,n}$, and between (c) $W_{r,s}$ and $W_{c,s}$. The first type of boundary is characterized by $\gamma_r = 0$, while the latter two by $\gamma_r^2 - \gamma_s^2 = 0$. Note that $\gamma_r = \frac{T_{21} - T_{12}}{2}$ is a linear function of the tensor entries in the asymmetric tensor field, while $\gamma_r^2 - \gamma_s^2$ is a quadratic function in terms of the tensor entries.

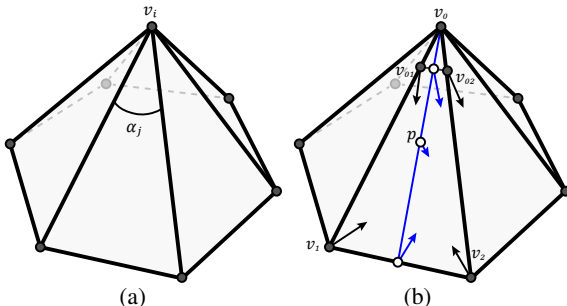


Fig. 4: (a) The left figure shows an example of non-zero Gaussian curvature. (b) The right figure illustrates the calculation of the interpolated vector at p from vectors at v_{01} , v_{02} , v_1 and v_2 .

In a triangle with a linear asymmetric tensor field, γ_r is thus a linear function of the coordinates of the points inside the triangle. Extraction of $\gamma_r = 0$ can be achieved using the Marching Triangle method.

In contrast, in the same triangle $\gamma_r^2 - \gamma_s^2$ is a quadratic function of the coordinates of the points inside the triangle. There is no guarantee that the curve $\gamma_r^2 - \gamma_s^2 = 0$ intersects the boundary or that it intersects each edge only once. To address this complexity, we develop the following procedure as illustrated in Figure 5. At the core of our method is the observation that the function $\gamma_r^2 - \gamma_s^2$ is quadratic inside the triangle, thus also a quadratic function along each line in the plane containing the triangle. Consequently, there are either zero or two solutions on a line. There are lines on which the two solutions coincide. Such lines are special in that the curve $\gamma_r^2 - \gamma_s^2 = 0$ will be contained entirely on one side of the line. This is essentially the same as identifying lines on which the projected function $\gamma_r^2 - \gamma_s^2 = 0$ has a zero discriminant. Note that the discriminant is also a quadratic function of the local coordinates of the points inside the triangle. Consequently, there are at most two such lines in the triangle. We divide the triangle into smaller triangles using these lines and compute points on the edges of the subdivided triangles that satisfy $\gamma_r^2 - \gamma_s^2 = 0$.

Such points are labelled as either a + point ($\gamma_r > 0$) or a - point ($\gamma_r < 0$). The former is on the boundary between $W_{r,n}$ and $W_{c,n}$ regions while the latter is between $W_{r,s}$ and $W_{c,s}$ regions. The number of + points must be even (counting multiplicity) along the boundary of a sub-triangle, so must the number of - points. We now pair the + points such that points on the same edge cannot be paired. Figure 6 illustrates this process with an example in which a loop internal to the triangle can be extracted. Similarly, all the - points are paired. Our algorithm then performs tracing of region boundary curves from these intersection points to generate the boundary curves.

For a triangle with a bilinear (quadratic) asymmetric tensor field inside, $\gamma_r = 0$ is now a quadratic function with respect to the coordinates of points in the triangle. Consequently, we can use the same idea for finding $\gamma_r^2 - \gamma_s^2 = 0$ in a linear tensor field (also quadratic with respect to the coordinates of points). To compute $\gamma_r^2 - \gamma_s^2 = 0$ for a bilinear asymmetric tensor field, we observe that the boundary is now a *quartic* function of the coordinates of points, which is still a polynomial. Therefore, the same scheme can be used, except now there can be up to four special line separators per triangle instead of two in the quadratic case. However, we can still compute special points on each of these dividers, and pair them the same way as we handle the quadratic function case.

Once the boundary curves have been generated, we connect boundary segments from adjacent triangles through common points on shared edges between the triangles. This leads to a set of loops or open curves that intersect the boundary of the domain. Each of these loops or open curves is considered one integral curve. We now use these curves to divide the domain into regions, each of which is a node in the eigenvector graph. Two regions sharing a common segment of boundary curves are given an edge between their corresponding nodes in the graph.

In addition, degenerate points are considered as nodes in the graph. They are extracted using the technique of Zhang et al. [35]. The triangle containing a degenerate point and the location of the degenerate point inside the triangle enables us to identify the region that contains it. In the eigenvector graph, the node corresponding to the degenerate point is connected by an edge to the node corresponding to the container region.

Following Lin et al. [21], we arrange the nodes in one of the six rows. From the top to bottom, the rows correspond to: (1) degenerate points $W_{c,n}$ regions, (2) $W_{c,n}$ regions, (3) $W_{r,n}$ regions, (4) $W_{r,s}$ regions, (5) $W_{c,s}$ regions, and (6) degenerate points inside $W_{c,s}$ regions. Note that with this arrangement, an edge can only exist between two nodes in adjacent rows, thus reducing the number of unnecessary edge crossings. This completes the construction of the eigenvector graph.

4.2 Eigenvalue Graph Construction

The construction for the eigenvalue graph is similar to that of the eigenvector graph. There are now eight types of boundaries, each

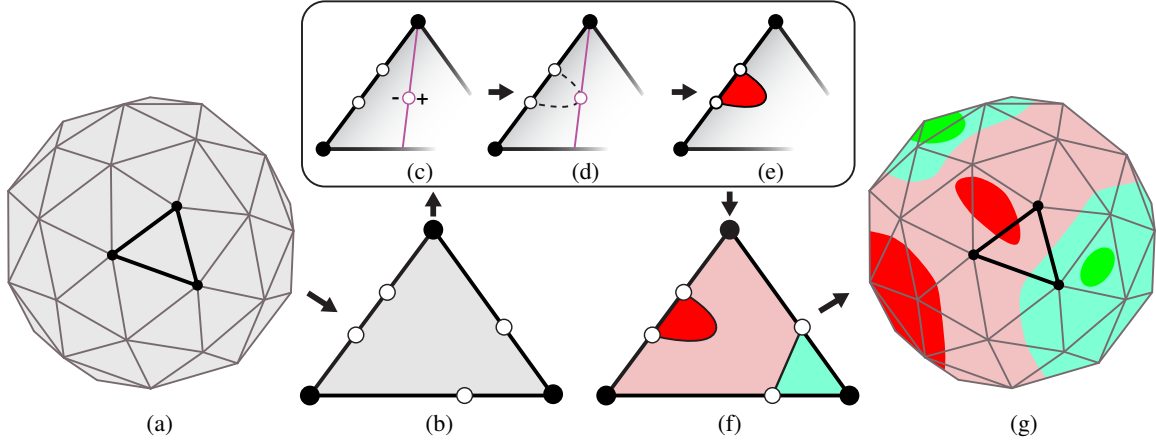


Fig. 5: Overview of region extraction: (a) given a tensor field defined on a surface of triangular mesh, (b) each edge of the triangles is visited to identify possible intersections (white disks) with region boundaries, (c) lines with two coinciding solutions (magenta) are detected within the triangle, (d) these lines partition the triangle such that boundary curve resides completely on one side of the line. The boundary can now be traced from one intersection to the other without suffering from the numerical errors, (e-f) regions are constructed within the triangle, (g) and finally, a merging process is performed to consolidate adjacent subregions with the same type into a single region on the surface.

characterized by one of the following equations:

1. $\gamma_r - \gamma_d = 0$: boundary between an $R^+ - D^+$ pair and between an $R^- - D^-$ pair.
2. $\gamma_r + \gamma_d = 0$: boundary between an $R^+ - D^-$ pair and between an $R^- - D^+$ pair.
3. $\gamma_r^2 - \gamma_s^2 = 0$: boundary between an $R^+ - S$ pair and between an $R^- - S$ pair.
4. $\gamma_d^2 - \gamma_s^2 = 0$: boundary between a $D^+ - S$ pair and between a $D^- - S$ pair.

The extraction of these curves separately is the same as that for eigenvector graphs. However, in the eigenvalue graph there are the junction points that are incident to three regions of different types. Consequently, we need to identify these points (possibly inside triangles). We first extract the region boundaries listed above inside the triangles. To locate junction points where $\gamma_s = \gamma_r = \gamma_d$, we solve two systems of polynomial equations, i.e., intersection of the boundaries defined by $\gamma_s = \gamma_r$ coupled with $\gamma_d = \gamma_r$ and intersection of the boundaries defined by $\gamma_s = \gamma_r$. Other types of junction points can be computed in a similar fashion. We can have at most two junction points in a triangle with a linear asymmetric tensor field and up to four junction points in a triangle with quadratic asymmetric tensor field. If there exist some junction points within a triangle, we need to find which of our previously extracted region boundaries inside the triangle connect to these points. The process is trivial when a junction point is exactly on a boundary i.e. $\gamma_d = \pm\gamma_r$ boundaries, we connect that junction point to the intersection points of that boundary on the edge. However, to find the nearest $\gamma_s = \gamma_r$ and $\gamma_s = \gamma_d$ boundary curves to each junction point, we need to trace these region boundaries inside the triangle. During tracing, each boundary curve is approximated by line segments (polyline) and we find the closest line segments of each type $\gamma_s = \gamma_r$ and $\gamma_s = \gamma_d$ to

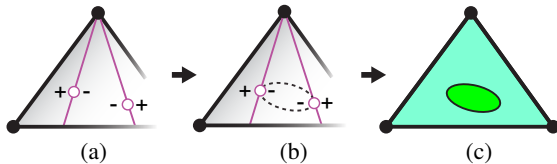


Fig. 6: An example of an internal elliptic loop. To detect this loop, we partition the triangle using line segments that contain repeated solutions to the region boundary equations (a). Next, We trace the boundaries and extract the region.

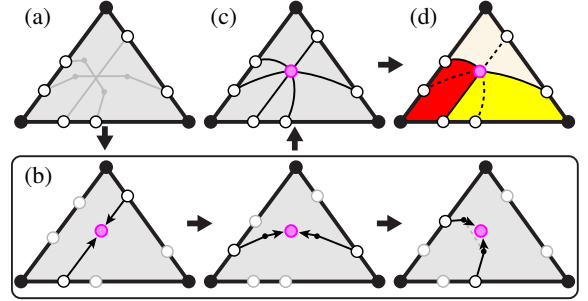


Fig. 7: This figure illustrates the steps of connecting the separately extracted intersecting region boundaries at a junction point. (a) We start with a triangle with three region boundaries. (b) We divide the $\gamma_d = \pm\gamma_r$ boundary on which junction point is located, and connect the nearest $\gamma_s = \gamma_r$ and $\gamma_s = \gamma_d$ boundary to the junction point. (c) The junction point is now on three intersected boundary curves. (d) We merge the adjacent regions of the same type.

individual junction points. These segments are then further segmented at respective junction points. Consequently, each junction point inside the triangle is now on three boundary curves. Figure 7 illustrates the steps of junction point placement on corresponding region boundaries and merging adjacent regions of the same type.

Once the boundary curves have been identified, we use them as described in eigenvector graph construction to divide the domain into regions, each of which is assigned a node in the eigenvalue graph. Then, an edge is assigned for each segment of common boundary between two regions. Given the constraints on the adjacency between the five types of regions, we derive five groups. The top group, in a row, is used to host all the D^+ nodes, while the bottom group, also in a row, hosts all the D^- nodes. The right group and left group, both in a column, host R^+ and R^- nodes, respectively. The middle group, shaped as a rectangle, hosts all the S nodes. Such an arrangement is designed to reduce the number of edge crossings.

5 MULTI-SCALE TOPOLOGICAL ANALYSIS

Once the eigenvector and eigenvalue graphs have been constructed from the input data, we convert them to a multi-scale data structure. As in the case of multi-resolution mesh representation [13] and multi-scale scalar field topology [4], we need to address the following fundamental questions in order to enable multi-scale asymmetric tensor field

analysis.

1. What is the set of atomic operations that use the bifurcations to reduce of the graph complexity?
2. How to realize these atomic operations?
3. How to decide on the order in which the graph is reduced?

We will address these challenges first for the eigenvector graphs.

5.1 Eigenvector Graph Simplification

In an eigenvector graph, there are two kinds of nodes: a region of a particular type ($W_{c,n}$, $W_{r,n}$, $W_{r,s}$, $W_{c,s}$) or a degenerate point. Furthermore, a degenerate point must be inside either $W_{c,n}$ or $W_{c,s}$, and no three different types of regions can be mutually adjacent (i.e. no junction points). Based on these observations, we have identified the following set of atomic operations:

1. Region annexation: in which two adjacent regions are combined into one region whose type is inherited from the container region. Semantically, the container region annexes the other region.
2. Region connection: two regions of the same type inside the same container region are joined into one region.
3. Degenerate point pair cancellation: in which two degenerate points (one wedge and one trisector) are removed from the field simultaneously.

Degenerate points must reside in complex domains, where eigenvector fields are complex-valued. Instead, dual-eigenvectors are considered for complex domains, which are the eigenvectors of a rotated version of the symmetric part of the asymmetric tensor field [26]. Consequently, degenerate point pair cancellation in an asymmetric tensor field is equivalent to the same operation for a symmetric tensor field. We adapt algorithms in [32].

Next, we describe our algorithms for region annexation and region connection in detail.

5.1.1 Algorithms

Figure 8(a) shows an example of region annexation, in which the inner region B , which must be a topological disk, is merged with the container region A .

In essence, we wish to change the type of asymmetric tensors in B to that of A . Recall that the type of tensors (i.e. $W_{c,n}$, $W_{r,n}$, $W_{r,s}$, $W_{c,s}$) is based on a single quantity $\phi = \arctan(\frac{\gamma_r}{\gamma_s})$. To realize this, we first find a topological disk $C \supset B$ that is contained in A . We then perform Laplacian smoothing [33] on ϕ on the interior vertices of C while holding the values of ϕ fixed on the boundary of C . Finally, we modify the tensor values on the interior vertices of C based on the new ϕ values.

Note that the boundary of C is in A , thus ϕ on ∂C leads to the same classification as A (e.g., $W_{c,n}$ as in Figure 8(a)). Furthermore, Laplacian smoothing on C leads to a harmonic function whose minimal and maximal values can only occur on the boundary [14]. This means that after Laplacian smoothing, all the interior vertices of C have their new ϕ values between the minimum and maximum ϕ values on the boundary of C . Since the ϕ values on the boundary of C lead to the same tensor classification, so will the new ϕ values inside C . Therefore, after smoothing, all the vertices in C , including the whole region B , will now have the same type as A , finishing the annexation process.

To construct C , we grow B by adding one adjacent triangle at a time until C is a topological disk, the boundary vertices of B are in the interior of C , and the boundary vertices of C are inside A .

Given a vertex in C with an asymmetric tensor $\gamma_d \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \gamma_r \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} + \gamma_s \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix}$, the original ϕ is $\arctan(\gamma_r/\gamma_s)$. With the newly computed ϕ' , we simply replace the tensor by $\gamma_d \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \gamma_r' \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} + \gamma_s' \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix}$ where $\gamma_r' = \sqrt{\gamma_r^2 + \gamma_s^2} \sin \phi'$ and $\gamma_s' = \sqrt{\gamma_r^2 + \gamma_s^2} \cos \phi'$.

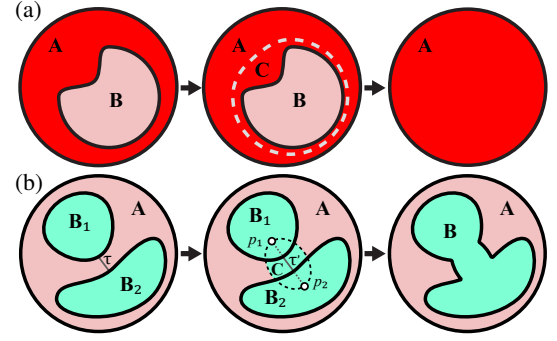


Fig. 8: This figure shows the two operations needed for the eigenvector graph simplification (a) region annexation and (b) region connection.

The region connection operation, illustrated in Figure 8(b), is realized in a similar fashion. Given two regions B_1 and B_2 of the same type inside the same container region A , we wish to join the region by building a tunnel region (Figure 8(b), middle) inside A that connects B_1 and B_2 . We employ a process similar to that of region annexation. First, we find the tunnel region that is a topological disk, intersects the interior of B_1 and B_2 , and is contained in A . Next, we perform Laplacian smoothing on ϕ on the tunnel region. Finally, we modify the tensor values inside the tunnel region based on the new ϕ value. Note that the last step is the same as that for region annexation.

To compute the tunnel region, we first compute a shortest path τ that connects the boundary of B_1 and B_2 such that τ is inside A . This can be achieved using Dijkstra's algorithm [5] on the underlying mesh. The only adaptation is to ensure that when adding an edge to the graph, we need to ensure that the edge is inside A .

Next, we compute the one-ring neighborhood of τ , which leads to region that contains τ in its interior. We further add triangles one at a time until the region is a topological disk, is inside A and does not completely enclose B_1 or B_2 (second step in Figure 8(b)). This is the region C .

We now extend τ in both directions until it intersects C at p_1 and p_2 , which are inside B_1 and B_2 , respectively. Next, we perform Laplacian smoothing on ϕ on the interior vertices of the extended path τ' with p_1 and p_2 's current ϕ values as the boundary condition. This results in new ϕ values on all the vertices on τ' leading to the same tensor classification as that of B_1 and B_2 's. Next, we perform the second Laplacian smoothing on ϕ on the region C , this time fixing the ϕ values at the boundary of C and on τ' .

Once we use the new ϕ values to modify the tensor field inside C , B_1 and B_2 are now connected (last step in Figure 8(b)).

5.1.2 Scale and Cost

With the aforementioned atomic operations we can reduce the complexity in the eigenvector graph and thus the tensor field. There are now two challenges that we need to address. First, when to stop the automatic simplification process given a user-specified scale? Second, when there are multiple atomic operations available, which one should be performed next? We solve both of these challenges by considering the benefit and cost of each atomic operation.

The benefit of an atomic operation is measured by the reduction in a topological complexity of the regions involved, which we divide into two components: the region complexity and the field complexity.

We measure the region complexity using $\sum_{R \in \mathcal{R}} ((1 - \chi(R)) + \frac{b(R)}{2})$ where $\chi(R)$ is the Euler characteristic and $b(R)$ is the number of intersection segments of R with the physical domain boundary. We measure the field complexity of the region by $\frac{d(T)}{2}$ where $d(T)$ is the number of degenerate points in our tensor field T . This metric was chosen because it counts the number of atomic operations needed for simplification, as every atomic operation decreases this value by at most one.

The cost of an atomic operation is that it modifies the field, making it less similar to the original field. Thus, we define the cost based on the

minimum modification to the field to make the topological change. The cost is the sum of $p(v)$ for all vertices in A , where $p(v) = |\gamma_s - |\gamma_r||$ if annexing a region or connecting two regions bounded by $\gamma_s = |\gamma_r|$ boundary and $p(v) = |\gamma_r|$ if annexing a region or connecting two regions bounded by $\gamma_r = 0$. In this equation, A is the minimum set of vertices to change, i.e. the region to be removed or the path to be connected. This $p(v)$ is then the minimum change to γ_r needed to shift region types for vertex v , and the sum measures the overall change.

The next atomic operation is automatically selected with the least cost-to-benefit ratio. The automatic simplification process stops when the cost-to-benefit ratios of all remaining editing operations exceed a user-specified value.

5.2 Eigenvalue Graph Simplification

Comparing to eigenvector graphs, eigenvalue graphs involve three quantities instead of two: γ_d , γ_s , and γ_r . This leads to the existence of junction points incident to three different types of regions. Therefore, there are now six atomic operations:

1. Region annexation without junction points: in which two adjacent regions are combined into one region whose type is inherited from the container region.
2. Region annexation with junction points: in which two adjacent regions are combined into one region whose type is inherited from the container region. Unlike the previous operation, there is a third region inside the container region that is adjacent to the region to be annexed (Figure 9(a)).
3. Region connection of the same type: two regions of the same type inside the same container region are joined into one region.
4. Region connection of different types: two non-adjacent regions of different types inside the same container region are made neighbors. In this case, two junction points are created between the regions (Figure 9(b)).
5. Region split of different types: two adjacent regions of different types inside the same container region are split so that they are no longer adjacent to each other. This is the reverse of the previous operation, in which two junction points are removed (Figure 9(c)).
6. Degenerate point pair cancellation: in which two degenerate points (one wedge and one trisector) in the same container region are removed from the field simultaneously.

The degenerate point pair cancellation is a relatively straightforward adaptation from its counterpart in the eigenvector graph. Next, we focus on the algorithms for the five region-based operations.

5.2.1 Algorithms

The region annexation operation without junction points for the eigenvalue graphs is similar to the region annexation operation for the eigenvector graphs (see Figure 8(a)). The only difference is that when performing Laplacian smoothing we also need to consider γ_d .

Without loss of generality, we assume the dominant component of the container region A is γ_r . Therefore, we aim at having region B also dominated by γ_r , i.e. $|\gamma_r| > \max(|\gamma_d|, \gamma_s)$ after the operation. We therefore define quantity $\delta = \arctan(\frac{|\gamma_r|}{\max(|\gamma_d|, \gamma_s)})$ when performing Laplacian smoothing. Once the new δ is computed for the region C enclosing B , we update the tensor values for the vertices in C by the new δ values similar to how we update the field after computing ϕ in eigenvector graph simplification.

For the operation to connect regions B_1 and B_2 of the same type that are inside the same container region A , the algorithm is essentially the same as that for eigenvector graphs (see Figure 8(b)), except we use the aforementioned δ as the quantity to perform Laplacian smoothing.

We now discuss the three remaining operations, which are specific to the eigenvalue graphs. As they all involve junction points, we introduce the notion of *junction point index*. There are four types of junction points, as illustrated in Figure 2(c): (1) (S, D^+, R^+) , (2) (S, D^-, R^+) , (3) (S, D^+, R^-) , and (4) (S, D^-, R^-) . In addition, for each type of

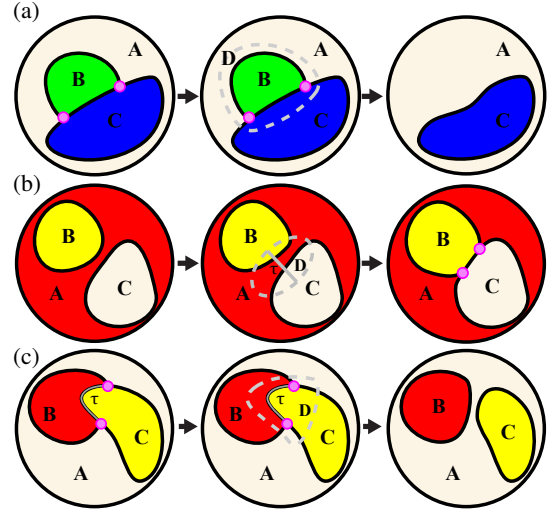


Fig. 9: This figure demonstrates the three additional operations needed for the eigenvalue graph simplification (a) region annexation with junction points, (b) region connection of different types, and (c) region split of different types.

junction points, there are two kinds. For example, when traversing through the associated regions of a (S, D^+, R^+) -type junction point in the counterclockwise orientation, one may encounter S , D^+ , and R^+ regions in this order (positively indexed) or in the reverse order (negatively indexed).

For region annexation with junction points (Figure 9(a)) we note that the junction pair must have opposite junction point indexes. To realize this operation, we compute a topological disk D that encloses B (see Figure 9(a)). In addition, we require D to also intersect region C , the third region that is inside A and is adjacent to B but does not enclose it. Note that this ensures that D necessarily contains the two junction points among A , B , and C . We now perform the Laplacian smoothing operation on δ with the boundary of D as the boundary condition. In this case, δ is based on the dominant component of A . This leads to the annexation of B by A while C remains, although not necessarily identical to that before the operation.

To connect two regions B and C of different types as shown in Figure 9(b), we again follow the algorithm for region connection to a tunnel region D and a path τ inside the tunnel region so that its ends are inside B and C , respectively. Furthermore, τ touches the boundary of D on both ends. We then perform Laplacian smoothing on the path τ on δ where $\delta = -\arctan(\frac{|\gamma_r|}{\min(|\gamma_d|, \gamma_s)})$ assuming that A , B and C are dominated by rotation, expansion, and pure shear, respectively. This ensures that along the path τ , the dominant γ component of A is smaller than both the dominant γ components of B and C . Finally, we perform the second round of Laplacian smoothing using the same δ on the tunnel region. This results in the connection of regions B and C inside A , with the creation of two junction points. Note that the junction points have opposite indexes.

To split two regions B and C of different types along their common boundary as depicted in Figure 9(c), let τ be the curved path connecting the two junction points between B and C . The two junction points must have opposite indexes. We then compute a topological disk D enclosing τ and intersects B and C but encloses neither. Next, we perform Laplacian smoothing on δ based on A 's dominant type on τ with its end points being the boundary condition. Then, we perform a second round of Laplacian smoothing on the same δ , now with τ and the boundary of D as the boundary condition. This leads to the removal of the junction point pair and the split of B and C .

5.2.2 Scale and Cost

Again, we need to measure the cost and benefit of our atomic operations for eigenvalue graphs. The addition of junction points requires an

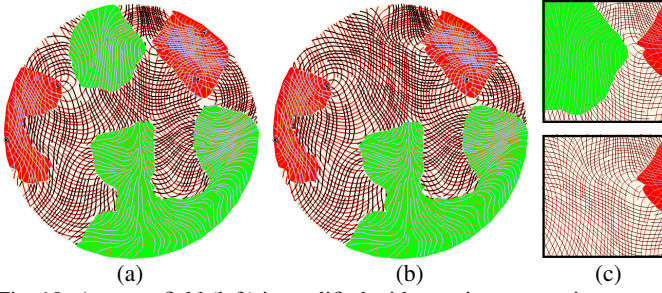


Fig. 10: A tensor field (left) is modified with a region annexation operation (right). Note that the $W_{c,s}$ region in the original field (left: near the upper side of the boundary) disappears after the operation (right, same location). In addition to visualize the tensor field using colors indicating the dominant tensor behavior, we also use hyperstreamlines to visualize the eigenvector fields in the real domain and pseudo-eigenvector fields in the complex domain. Major and minor eigenvectors are colored in black and red, respectively. In addition, major and minor pseudo-eigenvectors are colored in light blue and orange, respectively. In (c), we zoom in and compare the tensor fields around the removed region.

additional term in the region complexity metric used to measure the benefit of an operation. We add $\frac{J}{2}$, where J is the total number of junction points.

However, the cost of an atomic operation needs to be more sophisticated in order to measure the minimum change necessary, as there are multiple components to be considered. When removing a γ_X dominant region (for some component X)

$$\text{Cost} = \sum_{v \in A} \min(|\gamma_X| - |\gamma_Y|, |\gamma_X| - |\gamma_Z|), \quad (2)$$

where Y and Z are the other two components, and A is the region to be removed. This is the minimum change to γ_X to make it smaller in magnitude to one of the other two components.

Similarly, when connecting two γ_X dominant regions through a γ_Y dominant region, the cost is $\sum_{v \in A} (|\gamma_Y| - |\gamma_X|)$, where A is the set of vertices on the path between the two regions to be merged. This measures how much $|\gamma_X|$ needs to be increased along the path to merge the two regions.

To cancel a junction point pair between γ_Y and γ_Z dominant regions, the cost is $\sum_{v \in A} (\max(|\gamma_Y|, |\gamma_Z|) - |\gamma_X|)$, with A being the path connecting the two junction points, as along this path we update the tensors such that γ_X is bigger than the existing dominant component.

Finally, when connecting a γ_Y region to a γ_Z region through an intermediate γ_X region to create two new junction points we use Equation 2 to remove the γ_X dominant part of the path between the two regions. This also measures the change needed for this operation.

5.3 Field Modification Evaluation

Each of our editing operations modifies the tensor values at a set of vertices. Given the many operations applied to generate the multi-scale framework, impacting many if not all the vertices in the mesh, we wish to understand how much the field has been changed. Figure 10 shows the visual comparison of a slice of the diesel engine data (Figure 1) and a modified version after the green region (dominated by clockwise rotation) near the top has been annexed by the white region (dominated by pure shear). Note that the two fields are the same except around the green region. In addition, we wish to numerically measure the difference between the two fields. To do so, we define the *relative error* of the field after simplification T' to the field before simplification T as $\sum_{v \in V} \frac{\|T'(v) - T(v)\|}{\|T(v)\|}$ where V is the set of vertices in the mesh. In this example, the relative error is 0.199 thank to the relatively large area of the annexed region.

The same measure can be used to measure the difference between the original tensor field and the one after the multi-scale analysis. In our case, this overall error is 0.171 for the diesel engine data (Figure 1) and 0.296 for the open channel simulation (Figure 11).

6 PERFORMANCE

We estimate the performance as the cost of the eigenvector, eigenvalue graph, and multi-scale construction. We perform the experiments on a computer with an i7-8750H @2.20GHz CPU, 32 GB of RAM, and an NVIDIA GeForce GTX 1070 w/ Max-Q Design GPU. The complexity of the proposed system is proportional to the number of triangles in the mesh. In general, the time to compute an eigenvector graph is on the order of seconds while for eigenvalue graphs the cost is on the order of minutes. The major computational bottleneck lies in the topological analysis and region extraction on the triangles. The construction of the multi-scale framework is also on the order of minutes. However, after the one-time construction, our system can interactively visualize the manifold and graph of the eigenvector and eigenvalue at an arbitrary scale.

7 APPLICATIONS

In this section, we show results of applying our multi-scale analysis to a diesel engine simulation and an open channel flow simulation and provide physical interpretation of our analysis by domain experts. In the Appendix, additional results and physical interpretation are provided, namely, a cooling jacket simulation and the Sullivan vortex.

Diesel Engine Simulation In the flow within a diesel engine cylinder, the ideal pattern of flow motion takes on the form of swirl. In this case, rotational motion occurs about an imaginary axis. In the case of swirl flow, the axis is coincident with the cylinder axis. In order to generate swirl, fluid enters the combustion chamber from the intake ports. Later on in the engine cycle, the kinetic energy associated with this motion is used to generate turbulence for mixing of fresh oxygen with evaporated fuel [20].

The more turbulence generated, the better the mixture of air and fuel, and thus the more stable the combustion itself. Ideally, enough turbulent mixing is generated such that 100% of the fuel is burned. The swirl motion should be maximized to maximize turbulence. From the point of view of the mechanical engineers designing the intake ports, the ideal flow pattern leads to beneficial conditions including: improved mixture preparation, a higher EGR (Exhaust Gas Ratio) which means a decrease in fuel consumption, and lower emissions. However, too much swirl can displace the flame used to ignite the fuel, cause irregular flame propagation, or result in less fuel combustion. As such, a balance must be achieved between generating enough swirl flow and not displacing the flame used to ignite the flow. A controlled flow motion is used to get stable and reproducible conditions at each engine cycle.

One of the consequences of combining the intake port geometry with combustion chamber is associated with mesh resolution. In order to resolve the flow in the region of the intake ports, a very high-resolution mesh is required compared to the combustion chamber. The consequence of this is that a lot of noise is produced in terms of flow behavior in this area (top view). This is where the multi-scale simplification of topological features is very useful. We can see from the results, that many of the noisy flow patterns are reduced while the major characteristics of the flow can be preserved. By major characteristics, we mean flow associated with the focus swirl behavior.

Furthermore, we can observe considerably more flow characteristics with the tensor field view of the flow than that of the traditional vector field view alone. For example, in addition to the major regions of clockwise rotation at the surface of the combustion chamber (a positive sign) we also notice the areas of negative and positive scaling in Figure 1 (a negative sign). These areas deviate from the ideal pattern of swirl motion and, in fact, the engineers are most concerned with those regions of flow that reduce the turbulent mixing of the flow. Ultimately, it is these areas that the engineers study the most and as a consequence will modify the mesh geometry and thus re-run the simulation.

Open Channel Flow Figure 11 visualizes a steady open-channel flow disturbed by a submerged Gaussian-shaped sill on the channel bed. In (c) the side view of the flow data is shown, where the antique white intensity encodes the flow velocity magnitude. Notice the sill at the bed (inside the highlighted box). The flow direction is from left to right, and the submerged sill spans across the channel. One important

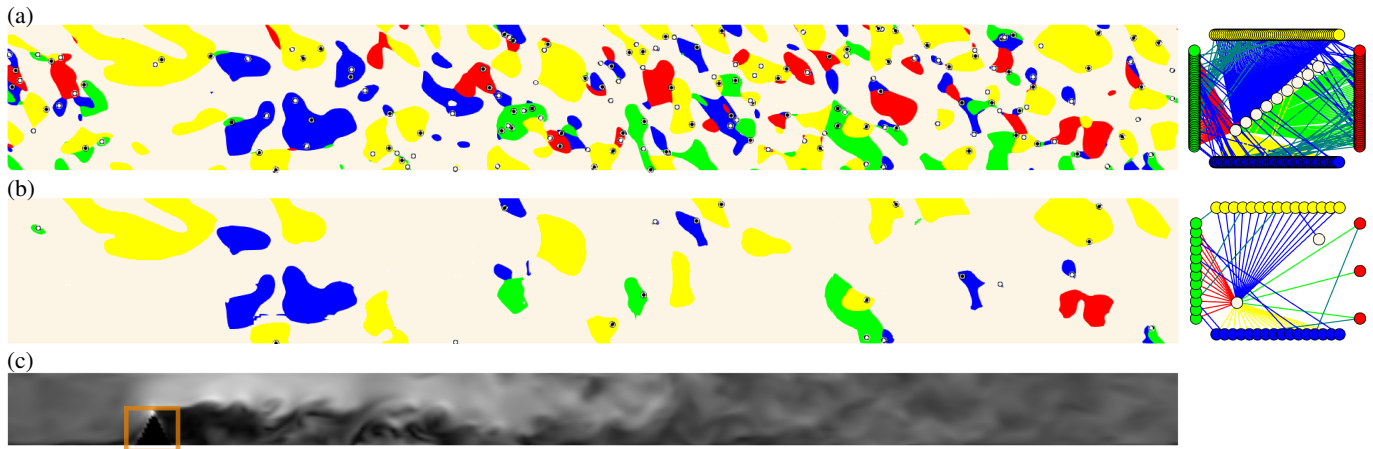


Fig. 11: An open channel flow is visualized. There is a sill at the bed of the channel (c: inside the box). While the sill location is not clear from the visualization of the flow velocity gradient tensor at the water surface (a), it becomes more clear after applying our multi-scale analysis to the original field (result shown in (b)). Notice the persistent large patches of upwelling (yellow, flow expansion at the water surface) and downwelling (blue, flow contraction at the water surface) around the sill.

research question regarding this flow is whether an object placed at the bed can be identified by observation of the water surface.

Figure 11(a) visualizes the flow at the water surface using the eigenvalue manifold (Figure 2(b-c)). Notice the mixed combination of kinematic characteristics (rotations, scalings, and expansions) together with the numerous degenerate points, which makes the interpretation difficult. Moreover, it is not clear where the sill is given the existence of upwelling (yellow regions, flow expansion at the water surface) and downwelling (blue regions, flow contraction at the water surface) throughout the channel. Using our multi-scale analysis (result shown in Figure 11(b)), the most important features remain after field simplification. The location of the sill on the channel bed is clearly identified by the distinct, large patches of up-welling and downwelling flow patterns immediately downstream of the sill location. Our multi-scale analysis shows promises in helping the detection of submerged objects from the water-surface.

There are additional important flow features emerged from the multi-scale analysis. First, it appears that once the predominant upwelling and downwelling flow patterns are formed by the disturbance of the sill on the bed, smaller but persistent patches of expansion and contraction appear periodically on the surface according to Figure 11 (b). Such a phenomenon is not obvious from the visualization of the original, un-simplified tensor field (Figure 11(a)). Second, the degenerate points represent the line vortices surrounded by the irrotational flows, i.e. highly concentrated vortices that are normally penetrated the air-water interface. The persistent presence of the vertical line vortices (degenerate points in Figure 11 (b)) is due to the density discontinuity on the stress-free surface, and is well observed and documented [22].

The aforementioned observations cannot be easily made without multi-scale analysis.

8 CONCLUSION

In this paper, we develop a novel multi-scale topology-driven analysis framework for asymmetric tensor fields on surfaces. Our analysis is based on the eigenvalue and eigenvector graphs introduced by Lin et al. [21], and we provide algorithms to perform this analysis on surfaces, in a multi-scale fashion. At the core of our analysis is the identification of atomic operations needed to reduce the complexity of the graphs, making important features more pronounced in the visualization. We also develop efficient and robust algorithms to construct eigenvalue and eigenvector graphs and perform atomic operations to reduce their complexity. These are made possible with some novel insights on the junction point classification. In addition, we provide physical interpretation of our multi-scale topological analysis that was absent from the work of Lin et al. [21].

Our approach is not without limitations. For example, our graph

construction algorithm requires the subdivision of a triangle. This leads to three meshes: the original, one for the eigenvector graphs, and one for the eigenvalue graphs. It is therefore our goal to pursue an approach that either does not require triangle subdivision, or in a way that is uniform for both the eigenvalue and eigenvector graphs. Another limitation of our current approach is that an excessive number of degenerate points in the field can hamper the region-based editing operations. To handle this, we involve a pre-processing step to remove some degenerate points first. In the future, we wish to make our region-based operations more flexible.

In this paper, we base our multi-scale analysis on the notions of eigenvalue manifold and eigenvector manifold [35], which are related. Combining the two manifolds into a single analysis framework can be a fruitful future research direction. In addition, we plan to investigate other feature definitions such as the zero level set of the determinant of an asymmetric tensor field. Such an investigation will be conducted through close collaborations with experts in relevant science and engineering domains.

In addition, we wish to investigate *irreducible* eigenvalue and eigenvector graphs given an asymmetric tensor field. This is needed for the development of algorithms guaranteed to reduce any asymmetric tensor field to an irreducible state. During this investigation, we expect additional editing operations to be identified in order to achieve an extreme simplification.

Finally, extending our work to 3D asymmetric tensor fields is a natural next step, and we plan to pursue this in our future research.

ACKNOWLEDGMENTS

We wish to thank our anonymous reviewers for their valuable feedback and suggestions. We also thank Zhongzang Lin for the help that we have received on the research project. Lily Shellhammer contributed to the voice recording of the supplementary video. This research is partially supported by NSF awards (# 1566236) and (# 1619383).

REFERENCES

- [1] C. Auer, J. Kasten, A. Kratz, E. Zhang, and I. Hotz. Automatic, tensor-guided illustrative vector field visualization. In *Proc. IEEE Pacific Visualization Symposium 2013*, pp. 265 – 275, 2013. doi: 10.1109/PacificVis.2013.6596154
- [2] D. B Ennis, G. Kindlmann, I. Rodriguez, P. Helm, and E. Mcveigh. Visualization of tensor fields using superquadric glyphs. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 53:169–76, 01 2005. doi: 10.1002/mrm.20318

- [3] H. Bhatia, B. Wang, G. Norgard, V. Pascucci, and P. Bremer. Local, smooth, and consistent jacobi set simplification. *CoRR*, abs/1307.7752, 2013.
- [4] P. T. Bremer, B. Hamann, H. Edelsbrunner, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):385–396, July 2004. doi: 10.1109/TVCG.2004.3
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Dijkstra’s algorithm. In *Introduction to Algorithms 2nd edition*, chap. 24, pp. 595–599. MIT Press.
- [6] L. De Floriani, P. Magillo, E. Puppo, and D. Sobrero. A multi-resolution topological representation for non-manifold meshes. In *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications, SMA ’02*, pp. 159–170. ACM, New York, NY, USA, 2002. doi: 10.1145/566282.566307
- [7] T. Delmarcelle and L. Hesselink. Visualizing second-order tensor fields with hyperstream lines. *IEEE Computer Graphics and Applications*, 13(4):25–33, July 1993.
- [8] T. Delmarcelle and L. Hesselink. The Topology of Symmetric, Second-Order Tensor Fields. In *Proceedings IEEE Visualization ’94*, 1994.
- [9] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, Nov 2002. doi: 10.1007/s00454-002-2885-2
- [10] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’97*, pp. 209–216. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997. doi: 10.1145/258734.258849
- [11] M. Garland and P. S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings of the Conference on Visualization ’98, VIS ’98*, pp. 263–269. IEEE Computer Society Press, Los Alamitos, CA, USA, 1998.
- [12] T. Gerrits, C. Rössl, and H. Theisel. Glyphs for general second-order 2d and 3d tensors. *IEEE Trans. Vis. Comput. Graph.*, 23(1):980–989, 2017. doi: 10.1109/TVCG.2016.2598998
- [13] H. Hoppe. Progressive meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’96*, pp. 99–108. ACM, New York, NY, USA, 1996. doi: 10.1145/237170.237216
- [14] D. Ioffe and R. Pinsky. Positive harmonic functions vanishing on the boundary for the laplacian in unbounded horn-shaped domains. *Transactions of the American Mathematical Society*, 342(2):773–791, 1994.
- [15] A. Kratz, C. Auer, M. Stommel, and I. Hotz. Visualization and analysis of second-order tensors: Moving beyond the symmetric positive-definite case. *Computer Graphics Forum*, 32(1):49–74, 2013. doi: 10.1111/j.1467-8659.2012.03231.x
- [16] J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *Computer Graphics and Applications, IEEE*, 11:36–46, 06 1991. doi: 10.1109/38.79452
- [17] D. H. Laidlaw, R. M. Kirby, C. D. Jackson, J. S. Davidson, T. S. Miller, M. da Silva, W. H. Warren, and M. J. Tarr. Comparing 2d vector field visualization methods: A user study. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):59–70, Jan. 2005. doi: 10.1109/TVCG.2005.4
- [18] R. S. Laramée, C. Garth, H. Doleisch, J. Schneider, H. Hauser, and H. Hagen. Visual analysis and exploration of fluid flow in a cooling jacket. In *Proceedings IEEE Visualization 2005*, pp. 623–630, 2005.
- [19] R. S. Laramée, D. Weiskopf, J. Schneider, and H. Hauser. Investigating swirl and tumble flow with a comparison of visualization techniques. In *IEEE Visualization 2004*, pp. 51–58. IEEE, 2004.
- [20] R. S. Laramée, D. Weiskopf, J. Schneider, and H. Hauser. Investigating Swirl and Tumble Flow with a Comparison of Visualization Techniques. In *Proceedings IEEE Visualization 2004*, pp. 51–58, 2004.
- [21] Z. Lin, H. Yeh, R. S. Laramée, and E. Zhang. *2D Asymmetric Tensor Field Topology*, pp. 191–204. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi: 10.1007/978-3-642-23175-9_13
- [22] E. P. Rood. Interpreting vortex interactions with a free surface. *Journal of Fluids Engineering-transactions of The Asme - J FLUID ENG*, 116, 03 1994. doi: 10.1115/1.2910248
- [23] J. Palacios, L. Roy, P. Kumar, C.-Y. Hsu, W. Chen, C. Ma, L.-Y. Wei, and E. Zhang. Tensor field design in volumes. *ACM Trans. Graph.*, 36(6):188:1–188:15, Nov. 2017. doi: 10.1145/3130800.3130844
- [24] J. Palacios, H. Yeh, W. Wang, Y. Zhang, R. S. Laramée, R. Sharma, T. Schultz, and E. Zhang. Feature surfaces in symmetric tensor fields based on eigenvalue manifold. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1248–1260, Mar. 2016. doi: 10.1109/TVCG.2015.2484343
- [25] J. Palacios and E. Zhang. Rotational symmetry field design on surfaces. *ACM Trans. Graph.*, 26(3):55, 2007. doi: 10.1145/1276377.1276446
- [26] D. Palke, Z. Lin, G. Chen, H. Yeh, P. Vincent, R. Laramée, and E. Zhang. Asymmetric tensor field visualization for surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1979–1988, Dec. 2011. doi: 10.1109/TVCG.2011.170
- [27] T. Schultz and G. Kindlmann. Superquadric glyphs for symmetric second-order tensors. *IEEE transactions on visualization and computer graphics*, 16:1595–604, 01 2011. doi: 10.1109/TVCG.2010.199
- [28] N. Seltzer and G. Kindlmann. Glyphs for asymmetric second-order 2d tensors. *Comput. Graph. Forum*, 35(3):141–150, June 2016. doi: 10.1111/cgf.12890
- [29] P. Skraba, B. Wang, G. Chen, and P. Rosen. Robustness-based simplification of 2d steady and unsteady vector fields. *IEEE transactions on visualization and computer graphics*, 21(08):930–944, aug 2015. doi: 10.1109/TVCG.2015.2440250
- [30] X. Tricoche and G. Scheuermann. Continuous Topology Simplification of Planar Vector Fields. In *Proceedings IEEE Visualization 2001*, pp. 159–166, 2001.
- [31] X. Tricoche, G. Scheuermann, and H. Hagen. A Topology Simplification Method For 2D Vector Fields. In *Proceedings IEEE Visualization 2000*, 2000.
- [32] E. Zhang, J. Hays, and G. Turk. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):94–107, 2007.
- [33] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics*, 25(4):1294–1326, 2006.
- [34] E. Zhang and G. Turk. Visibility-guided simplification. In *Proceedings of the Conference on Visualization ’02, VIS ’02*, pp. 267–274. IEEE Computer Society, Washington, DC, USA, 2002.
- [35] E. Zhang, H. Yeh, Z. Lin, and R. S. Laramée. Asymmetric tensor analysis for flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):106–122, 2009.
- [36] X. Zheng and A. Pang. 2D asymmetric tensor analysis. *IEEE Proceedings on Visualization*, pp. 3–10, Oct 2005.
- [37] X. Zheng, B. N. Parlett, and A. Pang. Topological lines in 3d tensor fields and discriminant hessian factorization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):395–407, July 2005.