Topological Segmentation of 2D Vector Fields

Roxana Bujack*, Etienne Bresciani†, Jiajia Waters*, and Will Schroeder‡

*Los Alamos National Laboratory, Los Alamos, NM, USA
E-mail: bujack@lanl.gov, jwaters@lanl.gov

†Centro de Estudios Avanzados en Zonas Áridas, Santiago, Chile
E-mail: etienne.bresciani@ceaza.cl

‡Kitware, Clifton Park, NY, USA
E-mail: will.schroeder@kitware.com

***Abstract*—Vector field topology has a long tradition as a visualization tool. The separatrices segment the domain visually into canonical regions in which all streamlines behave qualitatively the same. But application scientists often need more than just a nice image for their data analysis, and, to best of our knowledge, so far no workflow has been proposed to extract the critical points, the associated separatrices, and then provide the induced segmentation on the data level.**

We present a workflow that computes the segmentation of the domain of a 2D vector field based on its separatrices. We show how it can be used for the extraction of quantitative information about each segment in two applications: groundwater flow and heat exchange.

***Index Terms*—vector field, topology, segmentation, groundwater, heat exchange, separatrix, transversal**

I. Introduction

Vector field analysis is critically important for many physics and environmental applications, such as combustion rate modeling, material sciences, climate research, or space science. In particular, vector field topology is one of the most popular visualization techniques for flow data [13], [15] because it breaks down even huge amounts of data into a compact, sparse, and easy to comprehend description with little information loss [4], [14], [19], [29].

The topological skeleton of a vector field contains critical points (zeros of the velocity, Fig. 1), periodic orbits (closed streamlines), boundary switch points (boundary points with flow tangential to the boundary), and invariant manifolds (streamlines going in and out of saddles or boundary switch points). In 2D they separate the flow into simple ("parallel") regions called canonical regions in which all streamlines have the same origin and destination points or zones [25], [27]. The result is a highly compressed representation of the vector field that still contains its important features (Fig. 2).

Even though a significant amount of research has been dedicated to the extraction of the separatrices, there is not much work on the actual segmentation of the underlying domain. In this paper, we provide a workflow and an open source implementation to extract the segmentation of the domain, which is available at https://github.com/etiennebresciani/flowtopolopy. The implementation makes use of the visualization toolkit (VTK) [17], [37], especially the existing filter vtkVectorFieldTopology to extract the separatrices, and a newly developed algorithm that takes the separatrices as input and generates the actual segmentation of the dataset. The latter was integrated into the existing filter vtkPolyDataEdgeConnectivityFilter, so that it is also available as part of the open source VTK library. Its output contains a dataset of the type vtkPolyData, whose underlying grid is in agreement with both the input vertices and the separatrices of the vector field topology. It comes with an array that assigns its connected canonical region to each cell. The region IDs are sorted by decreasing region size. The segmentation can be used to generate deeper insight into the application scientist's data because it provides a more meaningful visualization than the topological skeleton itself. Among others, it enables the computation of region transversals, which are topological line features that intersect each streamline in a canonical region exactly once, and a thorough quantitative analysis of the different canonical regions, including flow rate and residence time. We demonstrate the use of the workflow in two environmental science applications.

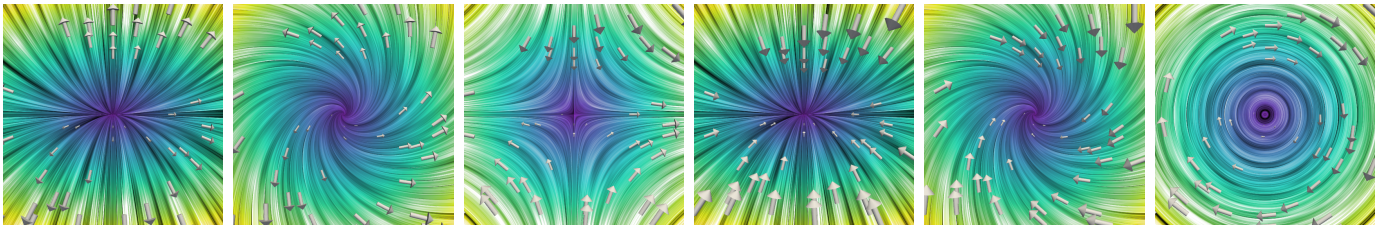
In a nutshell, the contributions of this paper are as follows:

- Implementation and open source integration of a segmentation algorithm respecting line boundaries.
- Workflow for 2D vector fields that generates:
  - The topological segmentation,
  - Transversals of each canonical region, and
  - Quantitative information, such as volume, flow rate, and duration of stay.
- Demonstration of the usefulness of the workflow in two applications in the environmental sciences, namely groundwater flow and heat exchange.

II. Related Work

Vector field topology is a very successful visualization tool [14], [19], [42]. The topological skeleton consisting of first order critical points and the invariant manifolds of saddles have been visualized for a long time [15]. Many extensions to 3D [12], higher order critical points [34], [36], [50], boundary switch points [10], [35], periodic orbits [51], [52], discrete topology [7], [20], [30], [40], and time-dependent flow [4], [29] have been suggested in the literature.

LEVIA'22: Leipzig Symposium on Visualization in Applications 2022



Rep. node $\Re(\lambda_i) > 0$, $\Im(\lambda_i) = 0$. Rep. focus $\Re(\lambda_i) > 0$, $\Im(\lambda_i) \neq 0$. Saddle $\Re(\lambda_1) > 0$, $\Re(\lambda_2) < 0$. Attr. node $\Re(\lambda_i) < 0$, $\Im(\lambda_i) = 0$. Attr. focus $\Re(\lambda_i) < 0$, $\Im(\lambda_i) \neq 0$. Center $\Re(\lambda_i) = 0$, $\Im(\lambda_i) \neq 0$.

Fig. 1: The different types of first order non-degenerate 2D critical points visualized with line integral convolution (LIC) [5] and arrow glyphs.

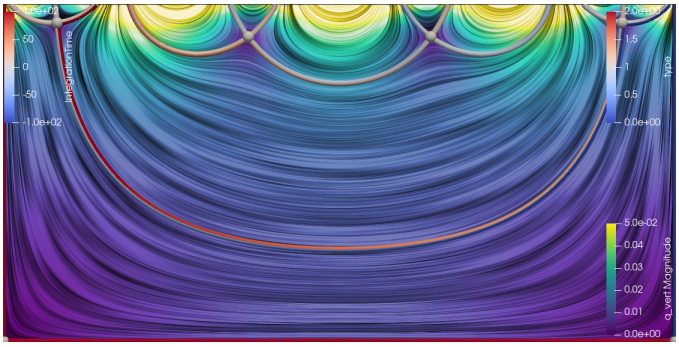


Fig. 2: Topological skeleton consisting of critical points and separatrices.

Only recently, an open source implementation has become broadly available to application scientists [3] through integration with VTK. So far, it only extracts 2D and 3D first order critical points and invariant manifolds of saddles [15], but extensions to periodic orbits and boundary switch points are in progress.

In contrast to the myriad of works dedicated to the extraction of the separatrices, there is not much in the literature about the actual segmentation of the domain underlying the vector field once they are extracted. There exists however work in which the process is the other way around, namely that a segmentation is produced through clustering of streamlines and then the separatrices are extracted as boundaries between the components. This process requires a dense cover of the whole domain with streamlines, which is significantly more computationally expensive than the one presented in this paper.

Mahrous et al. [24] state that 3D data sets often do not have critical points and detect a topological segmentation by extracting connected components in which all streamlines have the same destinations on the boundary. They make use of a material interface algorithm to assign each vertex to a component. Later [23] they improve their work including the origin of the streamlines, critical points, cell locking, and smart sampling.

Otto et al. [26] apply a similar method to uncertain 2D vector fields, where they trace the origin and destination of streamlines probabilistically.

Rössl and Theisel [32] segment the domain by applying standard clustering to a Euclidean approximation of the Hausdorff metric between streamlines. They claim that the resulting clusters coincide with the topological segmentation.

Analogously to Mahrous et al., Wang et al. [49] segment connected regions based on the origin and destination of streamlines. They encode origin and destination zones in one number, which they assign to the cells traversed by the streamlines. The separatrices are found using this scalar field. In contrast to Mahrous et al., they do not consider critical points as potential origins and destinations but only inflow and outflow region on the domain boundary.

The Topology toolkit (TTK) [44] offers segmentation techniques for scalar fields. The FTMTree filter assigns an index to each point in the dataset that indicates its correspondence to a branch in the contour tree. This simple assignment is sufficient, because the critical points of a scalar field coincide with its vertices. The analogous problem for vector fields is very different. Not only are the critical points located inside cells, but also do the separatrices follow arbitrary paths across the domain. Therefore pixel-based techniques cannot solve this problem.

Our contribution differs from works on topology preserving simplification [22], [41], which end up with grids that are not aligned with the separatrices.

III. Algorithm

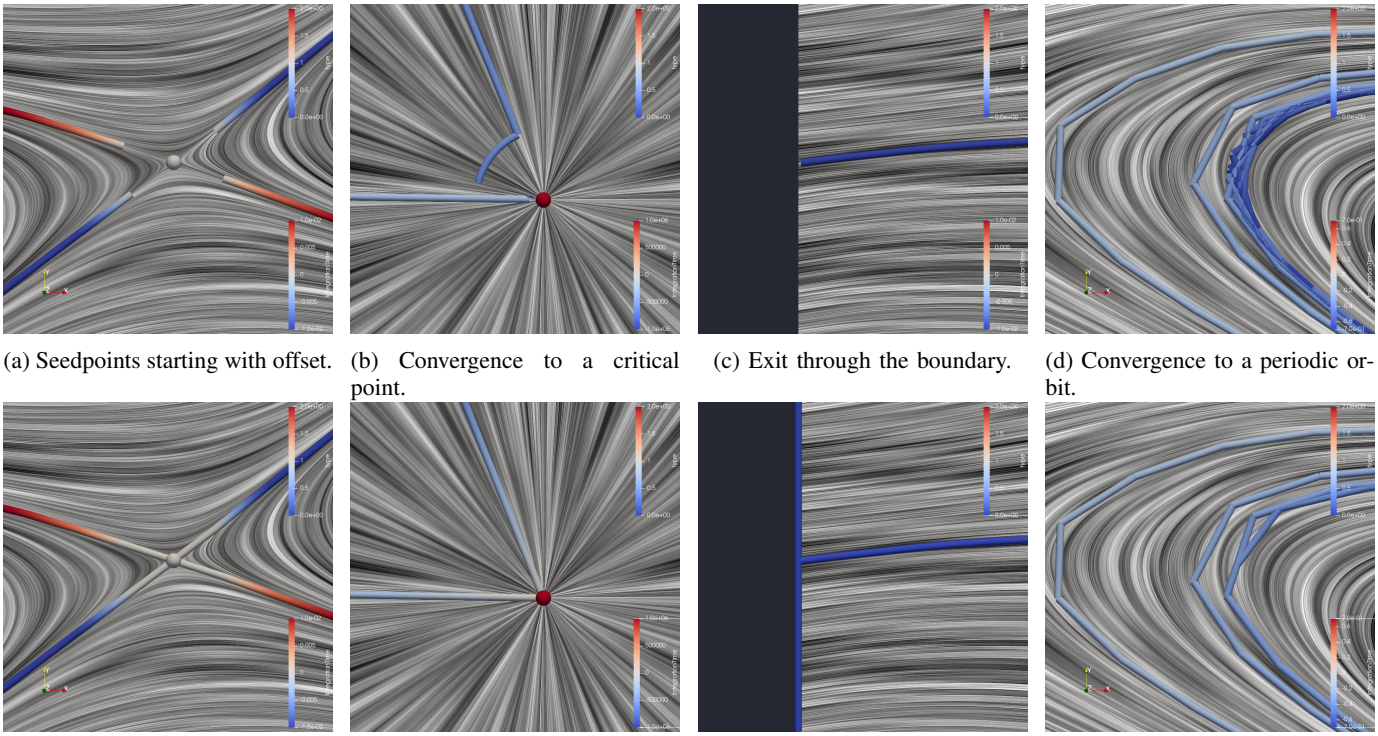
In a nutshell, the full workflow consists of the following steps:

1) Extraction of the separatrices
2) Closing gaps and removing intersections
3) Triangulation in agreement with the separatrices
4) Extracting and merging regions
5) Merge with the original grid

We will explain each step in detail in the following subsections.

A. Extraction of the Separatrices

We extract the critical points and the separatrices using the vtkVectorFieldTopology filter [3]. The topological skeleton of a d-dimensional vector field $v : \mathbb{R}^d \rightarrow \mathbb{R}^d$ consists of its critical points (i.e., the positions that have zero velocity



(a) Seedpoints starting with offset. (b) Convergence to a critical point. (c) Exit through the boundary. (d) Convergence to a periodic orbit.

(e) Seedpoints starting with offset. (f) Convergence to a critical point. (g) Exit through the boundary. (h) Convergence to a periodic orbit.

Fig. 3: Closing gaps and removing intersections. Top: before, bottom: after.

$v(x) = 0$) and separatrices (i.e., the co-dimension one invariant manifolds). The filter assumes linear interpolation within triangular cells and solve the locations of the critical points using that formula analytically. Then the critical points are categorized based on the eigenvalues of the vector field gradient $\nabla v : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ as sources (repelling nodes and foci), sinks (attracting nodes and foci), saddles, and centers, see Fig. 1. The filter computes the separatrices starting or ending at saddles through integration starting from seedpoints placed a small offset away from them along the principal axes of their Jacobian using a Runge-Kutta method in vtkStreamTracer.

B. Closing Gaps and Removing Intersections

The output of the vtkVectorFieldTopology filter is suitable for visualizations, but not for an actual segmentation in its off the shelf form. It needs to be adapted in three ways to induce a consistent segmentation.

First, we have to combine the separatrices with the boundary of the input dataset. Especially in unstructured data that contains non-rectangular boundaries or holes, it is crucial for the lines to be connected to closed loops using the actual boundary. We extract it with the vtkFeatureEdges filter and combine both using vtkAppendPolyData.

Second, we have to close gaps. The separatrices are seeded with a small offset from the saddles. This gap right around the saddles would result in a bleeding of separate regions into each other. We close these gaps by explicitly adding a line between the saddle and each seedpoint, Fig. 3(a),(e). Similarly, there are gaps between the end of a separatrix and its destination. If the separatrix converges to a critical point, the integration will abort at some point because the velocity drops to zero and the integration stagnates. We look for these types of ends of separatrices and explicitly add a line between the closest point of the trajectory to the nearest critical point while discarding the following tail, Fig. 3(b),(f). Finally, if a separatrix exits through the boundary, then the integration will stop right before leaving a gap between the line and the boundary. We explicitly add a line that continues the streamline with the boundary, Fig. 3(c),(g).

Third, we have to make the separatrices produce a consistent segmentation. The consistency is violated if separatrices intersect themselves or each other. Even though this cannot occur in theory, it very much does so in practice due to numerical error in the Runga-Kutta integrator even when using higher order methods with adaptive step size. Please note that this step is significantly different from methods detecting closed streamlines [21], [51]. While they focus on the detection of periodic orbits as meaningful topological features, we are only interested in removing artifacts that prevent an admissible segmentation. We follow the streamlines from the beginning, detect the first intersection using basic geometry, split the two involved line segments into four line segments by adding the intersection point and connecting it to all four involved points and remove the two original, intersecting line segments. Although this procedure does not resolve the theoretical in-

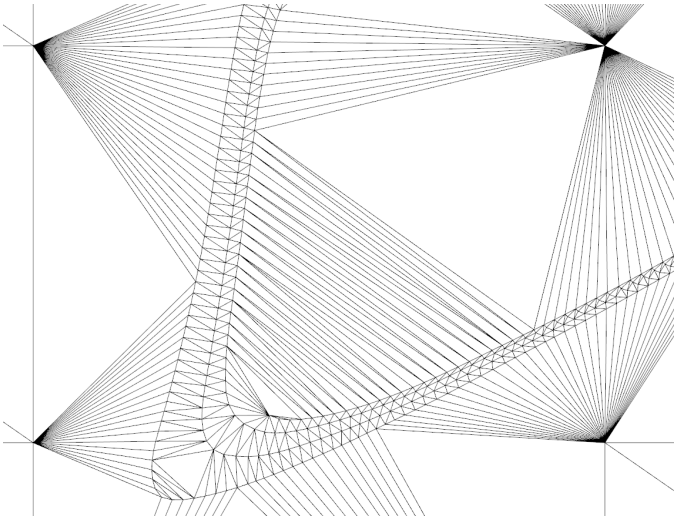


Fig. 4: Triangulation constrained by separatrices. Two additional points on the right have been added via regular sampling of background space to improve numerical stability.

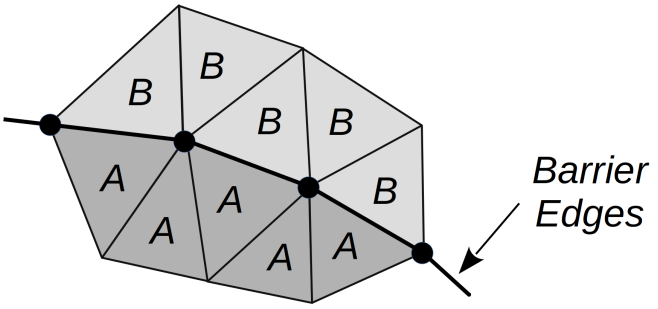


Fig. 5: Barrier edges can be used to separate regions. Here regions A & B are separated by edge barriers. These barriers may be based on explicit specification via separatrix edge segments; or via edge length.

consistency of intersecting streamlines, it does allow for a meaningful segmentation. We perform this step repeatedly until no more intersections are found to make sure that the newly inserted line segments are treated, too. For self-intersecting streamlines, we do an extra step that consists in traveling along the streamlines and removing the tail after the first return to itself. As streamlines tend to produce self intersections when they approach periodic orbits, this procedure results in a significant speed-up of subsequent treatments and a clearer visualization because every new revolution just produces more clutter and increasingly more intersections with previous ones, Fig. 3(d),(h).

C. Triangulation in Agreement with the Separatrices

Once the separatrices of the domain are formed, segmenting the topological regions of flow is in principal complete since each is bounded by curves that enclose a subregion of the domain. However, there are several practical concerns that require additional computational steps. For example, it is difficult to render complex, typically concave regions using standard graphics libraries and hardware. Instead, such regions must be triangulated to produce a tessellation that can be easily processed and visualized. Moreover, to avoid incompatible meshes (i.e., those with cracks or topological gaps between adjoining triangles), the tessellation must be performed to ensure mesh compatibility across subregion boundaries.

To perform the triangulation using VTK, we use the vtkDelaunay2D filter. Since the separating curves are represented by polylines, the points and (optionally) edge segments are used to construct a constrained Delaunay triangulation. The constraints on the triangulation are the line segments that compose the separatrix. To improve the overall shape of the resulting triangles, additional points can be added as well, Fig. 4. These additional points can be readily generated using a tessellated plane or even random point generation. The Delaunay triangulation is only well defined if all points are mutually different. Therefore, merging near coincident points is necessary to avoid numerical issues. The filter vtkCleanPolyData works well for this.

D. Extracting and Merging Regions

Once each region is triangulated, the triangles within each region are labeled and extracted. The filter vtkPolyDataEdgeConnectivityFilter (a new implementation to support this work) is used to perform this task. The filter simply traverses the mesh to identify and assign region labels to all edge-neighboring triangles. The triangle t_j is considered an edge neighbor to triangle t_i if t_i and t_j share a triangulation edge, and the edge is not specified as a barrier edge, Fig. 5. Barrier edges are defined by the line segments composing the separatrices and separate one region from another. Note that the traversal proceeds by propagating waves across edge-connected regions, with one connectivity traversal per region required to label all regions. Once the segmentation process completes, the regions are sorted by their total area and output in descending order. This facilitates further analysis and visualization, see below.

In practice, numerical issues are not uncommon, thus additional steps are taken to ensure the stability of the algorithm proposed here. This is typically due to the numerical sensitivity of the Delaunay triangulation, but also because of large differences in point spacing due to separatrix convergence near critical points - the end result being the generation of relatively small triangles. To address these numerical issues, we extend the concept of topological barrier edges (i.e., line segments defining the separatrix) and define *length-based* barrier edges that are defined as triangle edges of a length within a specified range of edge lengths. Using small-length barrier edges has the effect of segmenting out many small (in terms of area) regions, while enabling large regions to grow in a relatively unconstrained fashion. To improve the final result, we introduce a GrowSmallRegions() function which combines many small, edge-neighboring regions into single, larger regions. The final result is much improved, and prevents

unexpected connectivity traversal leakage from one separatrix region into another.

E. Merge with the Original Grid

Finally, we combine the vertices of the original grid with the new dataset. Each of the original vertices falls into one of the segments and gets the corresponding ID assigned through probing. The final dataset then contains all information of the input and allows visualization and following data analysis with full fidelity, e.g. streamlines or line integral convolution (LIC) [5] as in Fig. 9.

IV. Derivation of Quantitative Information

The segmentation allows for all sorts of quantitative measures that are useful to better characterize, understand, and communicate the underlying dataset.

To start with, we count the **number of regions** in the domain, which is a measure of the topological complexity of the flow [43]. To provide additional insight, counting can also be done distinguishing different **types of regions** such as closed regions (e.g., eddies) and open regions (i.e., regions where streamlines terminate on a domain boundary).

The **region area** is simply obtained by calculating and summing the areas of all the cells belonging to a region. Region area can be used for example to measure the regions and rank them, thus forming a rationale for focusing further description and analysis on the largest ones. It is also useful to calculate the mean residence time in each region, as explained below.

A fundamental property of the canonical regions is that they admit global transversals (i.e., lines that intersect every streamline of the region in exactly one point) [25]. This property is crucial for calculating the **flow rate** across each region by generating orthogonal transversals and integrating the velocity along them. Transversals are streamlines of the orthogonal flow field in 2D, visualized in Fig. 6 for the groundwater and in Fig. 9(b) for the heat exchange simulation. If the vector field is divergence free, the flow rate is the same across all the transversals of a region, except for transversals ending on a domain boundary, which could carry less flow. If the vector field is not divergence free, the flow rate can vary across the region. In both cases, we suggest to calculate the flow rate across a large number of transversals, seeding them along the separatrices (Fig. 6). Then, the maximum flow rate can be taken as representative of the region. If the vector field is divergence free, the mean **residence time** can also be calculated by dividing the region area by the flow rate [9]. Otherwise, the variations of the flow rate across the region can be analyzed.

Transversals can also be used to seed streamlines along them for an effective visualization of the flow patterns. Indeed, doing so for one transversal in each region will generate a physically meaningful distribution of streamlines throughout the domain, thereby providing a solution to the difficult streamlines seeding problem [33]. To this end, in each region we choose the transversal that is the longest among those that carry at least 99 % of the maximum flow rate (Fig. 6). It would be possible to seed the streamlines evenly along the transversals [21], [53], but the application scientists pointed out that they prefer a more physically meaningful placement in which the density represents the flow rate. Therefore along the selected transversals, streamlines are seeded so as to keep an equal flow rate between two consecutive streamlines. As a result, dense streamlines reflect high flow velocity, in line with the natural intuition from experiments. Furthermore, if many streamlines are generated in this way, the full distribution of residence times in each region can be calculated.

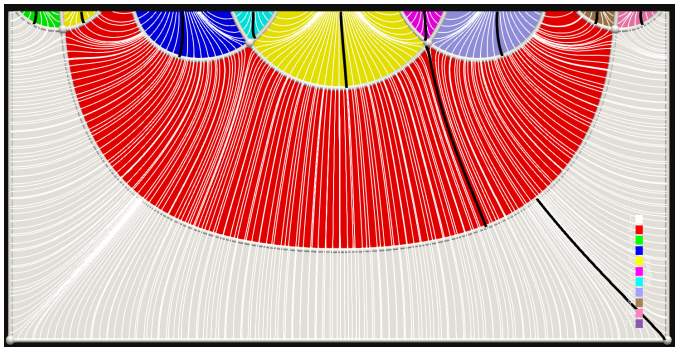


Fig. 6: White: transversals seeded on separatrices, black: the longest ones chosen for streamline seeding.

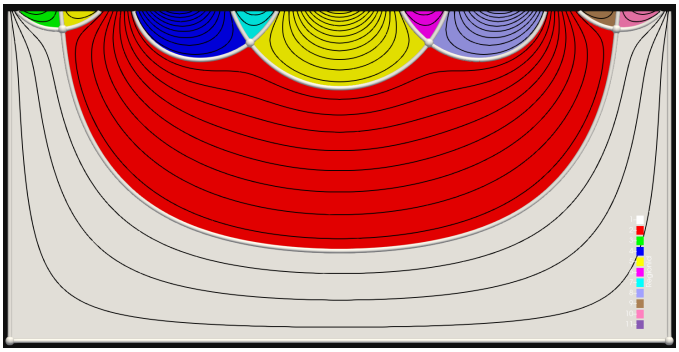
V. Applications

In this section, we showcase the usability of the workflow in two application cases of interest to the environmental scientific community.

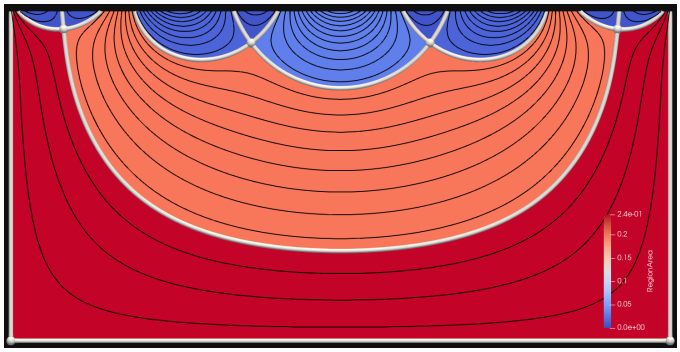
A. Groundwater Flow

Groundwater plays a critical role in the water cycle, and it is an essential source of water for humans. It typically recharges from rainfall and flows through several geological layers before reappearing on the surface in the form of springs, river baseflow, or wetlands. While flowing, groundwater carries along chemical elements that can react with the rocks through which it percolates. These reactions, which depend on the types of rocks met along the flow path, affect water chemistry and control various geological processes. Therefore, numerous scientific and environmental problems require a good understanding of groundwater flow paths and residence times.

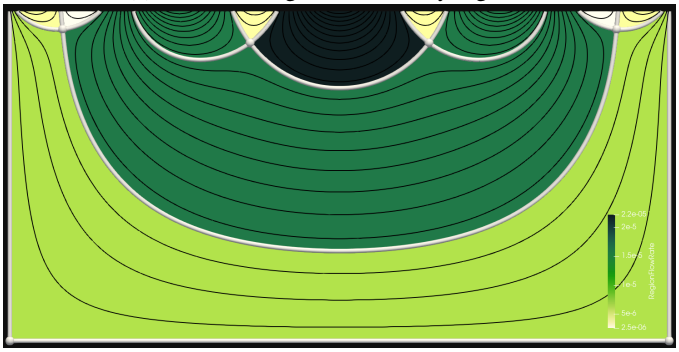
In a landmark study of regional-scale groundwater flow, Tóth [45] recognized "a certain grouping of the flow lines" in vertical 2D groundwater flow simulations. This observation prompted them to define the concept of flow system, which is essentially equivalent to the canonical regions defined herein. This concept provides an effective framework to describe and interpret groundwater flow and reactive transport at regional scale [6], [46], [54], making it one of the most fundamental concepts in modern hydrogeology [1], [2], [47]. However, until now, no quantitative analysis has been presented of the properties of these flow systems (i.e., volume, flow rate, etc.).



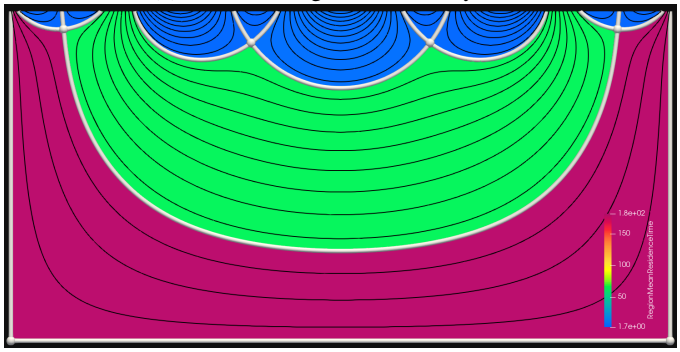
(a) Canonical regions colored by region id.



(b) Canonical regions colored by area.



(c) Canonical regions colored by flow rate.



(d) Canonical regions colored by mean residence time.

Fig. 7: Segmented groundwater flow field and quantitative information of each canonical region. Note: the mean residence time was calculated assuming a uniform porosity of 1.

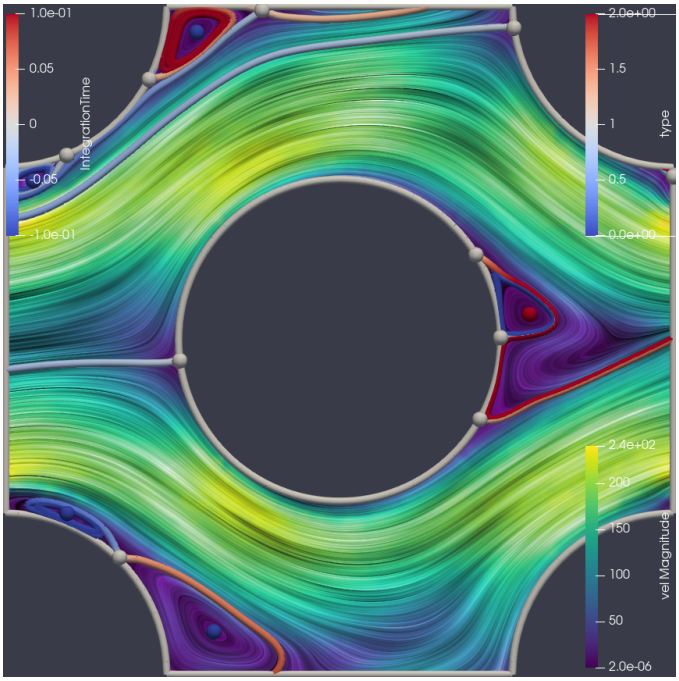
This knowledge gap is partly due to the lack of efficient and accessible tools to segment a groundwater flow field into its different flow systems (notwithstanding several research efforts in this direction [48], [49]).

We reproduced one of the groundwater flow fields analyzed in Tóth's landmark study and segmented it into its canonical regions (Fig. 7(a)). The result is a vivid display of the 11 flow regions that make up this particular example, in agreement with visual segmentations produced manually by previous authors [16], [45]. Calculation of the regions areas reveals that the shallower regions (i.e., those located closer to the top boundary) are smaller than the deeper ones (Fig. 7(b)). The flow rate generally follows an inverse logic (i.e., deeper regions carry less flow), except for the smallest shallow regions, which carry little flow due to their limited size (Fig. 7(c)). These two variables combine into the mean residence time, which is clearly larger in deeper regions (Fig. 7(d)).

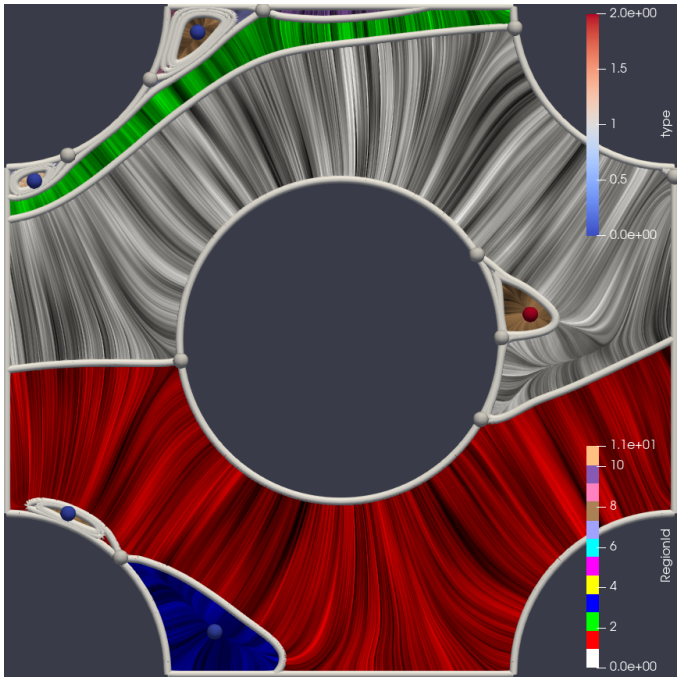
Etienne Bresciani is a researcher in hydrogeology at the Center for Advanced Studies in Arid Zones (CEAZA), Chile. He comments: "Groundwater flow is a complex process, but it is not chaotic (a priori). In this context, vector field topology is extremely relevant for characterizing flow patterns and associated transport properties. The hydrogeological community has long been waiting for a tool such as the one presented here. It will allow for the quantitative study of groundwater flow patterns, which, so far, have mostly been studied qualitatively".

B. Heat Exchange

Cross-flow in tube bundles has been extensively studied due to its frequent uses in many environmental applications. A tube bundle is a well-packed collection of pipes that is used as a radiation core in a heat exchanger device. Tube bundles are an ideal cooling solution for a wide variety of applications, such as offshore structures for the production and transmission of electricity, oil, gas [39], and other resources and cooling systems for nuclear plants [8]. Many experimental and numerical studies have been carried out to understand the physics of flow in arrays of cylinders. Here, we show the results of a flow simulation through a tube bundle that idealizes flow in the lower plenum. In this work, we are studying the flow of water through a staggered tube bundle [31], [38] by assigning proper boundary conditions to reproduce arrays of cylinders. The computation was undertaken for only one set of staggered tubes with periodic conditions modeling the repetitive configuration along the stream-wise (x) and span-wise (y) directions. A stream-wise periodic condition was applied at the inlet and outlet boundaries. This simulated constant mass flow was based on the average velocity of water flow $106\ cm/s$. The stream-wise periodic boundary condition is taken as fully developed flow, while the span-wise periodic lateral boundary is assumed to be repetitive in the y-direction. The Reynolds's number is 18000, hence a Large Eddy Simulation (LES) [11], [28] is applied in a finite element method to capture the eddies shown in Fig. 8.



(a) Separatrices colored by integration time. Red: forward, blue: backward. Background LIC of flow colored by velocity.



(b) LIC of orthogonal flow colored by region id of segmentation.

Fig. 8: Line integral convolution (LIC) [18] and topology of the heat exchange flow. Critical points colored by type. Red: repelling, white: saddle, blue: attracting.

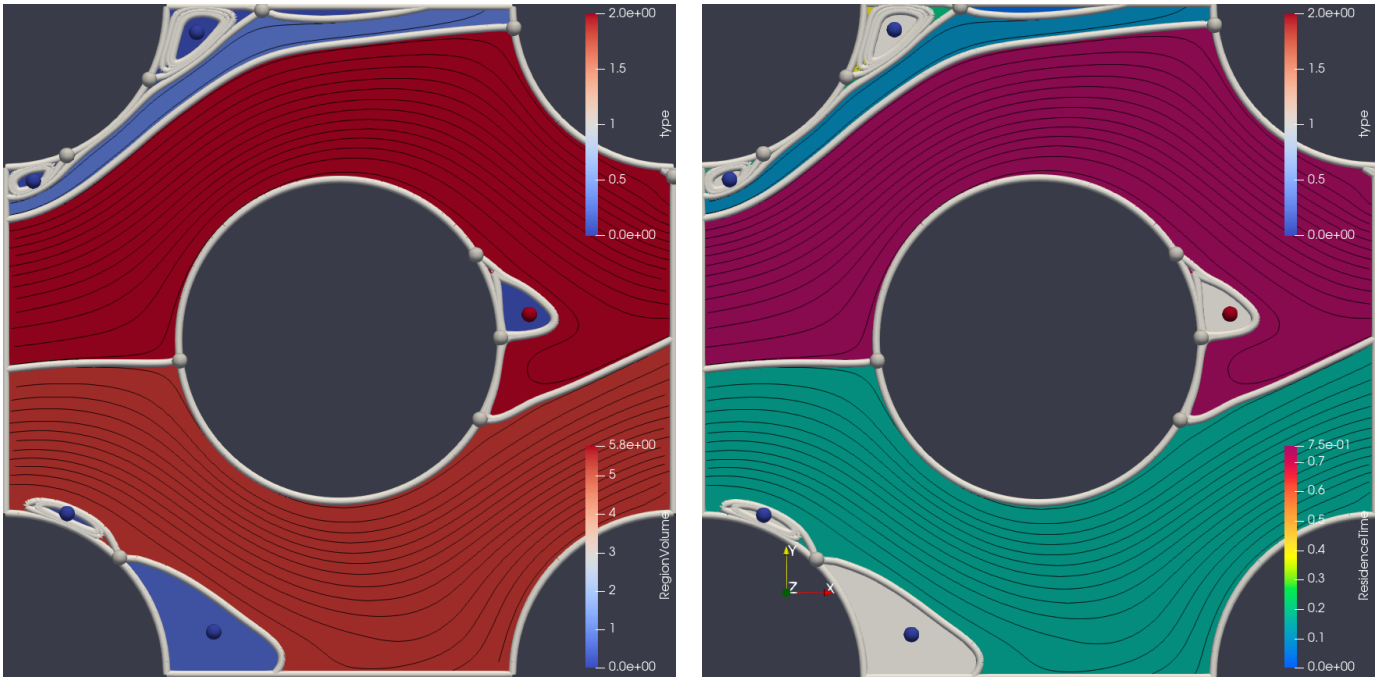
In order to guarantee an efficient heat transfer throughout the domain, the engineers want to analyze the effect of eddies on heat transfer as supposed to the laminar part of the fluid. Our segmentation allows to easily distinguish regions where water moves freely through the domain (efficient flow) from regions where water is stuck in recirculating zones (dead flow), Fig. 9(a),(b). Then one can compute the ratio between efficient flow and dead flow in terms of numbers and volume, which provides the application scientist with an immediate evaluation of the quality of their setup in one number. We were able to determine systematically which regions contain trapped water by examining the maximum integration time of particles seeded along the transversals. If it continues to grow with the maximum permitted number of steps in the Runge-Kutta integrator, then the water is trapped. We assign a "nan" value to the respective regions, which can then be assigned a distinct color (here white) when visualizing regions' mean residence time, Fig. 9(b). Adding up their area shows that they amount to only 0.5% in the currently investigated configuration.

This application turned out to be challenging for several reasons. First, the unstructured grid and especially the concave shape of the domain required an additional cropping step because the Delaunay algorithm naturally returns a triangulation of the domain's convex hull. Second, the strong spiralling behavior of the flow caused numerical instability during the Delaunay triangulation. The barrier edge length parameter allowed a successful segmentation despite this problem.

Jiajia Waters is an expert on fluid dynamics at Los Alamos National Laboratory studying the cross-flow of different tube bundle designs through numerical simulations. She collaborates with us applying the workflow to an application relevant to her research. She comments: "With this approach, we can easily analyze the results of our simulations. This saves a lot of time and is way more precise compared to the old analysis method, where we had to visually estimate the boundaries of the regions of different flow behaviors based on streamline plots."

VI. Runtimes

The computational requirement of the proposed segmentation algorithm can approximately be compared to the computational requirement of existing topological segmentation algorithms that involve computing forward and backward streamlines from all the cells of the grid [24], [49]. The comparison is not one to one because in contrast to the latter, our algorithm generates segmentations with subpixel accuracy. Since streamline computation is the most time-consuming part of those methods, we compared the runtime of the proposed segmentation algorithm with the runtime of computing forward and backward streamlines from all the cells of the grid in the groundwater flow example, which has 40,000 cells and the heat exchange, which has only 4,000 cells (Table I). Streamlines were computed with vtkStreamTracer, in which the runtime largely depends on the chosen integration step. Our segmentation algorithm also somewhat depends on the streamline integration step that is used for computing the



(a) Canonical regions colored by area.

(b) Residence time is an indicator of which regions are trapped.

Fig. 9: Quantitative analysis of the heat exchange simulation. The area of regions with a residence time of infinity (white), adds up to only 0.5% of the overall area.

separatrices. For the same integration step of 0.1 (in cell size unit), the computation time of our segmentation algorithm is 24 times smaller than that of computing forward and backward streamlines from all the cells for the groundwater example. For an integration step of 0.5, this number falls to 8, which is still a significant difference. Even for the very small heat exchange dataset, the proposed algorithm is significantly faster. The computation time of our algorithm is mostly spent in step E, i.e., for the merge with the underlying grid (Table II).

VII. Conclusions

We have presented a workflow (https://github.com/etiennebresciani/flowtopology) to extract the topological segmentation of 2D vector fields that is based on filters all publicly available through the open source library VTK. The overall workflow heavily relies on the three filters: vtkVectorFieldTopology, vtkDelaunay2D, and vtkPolyDataEdgeConnectivityFilter, which have been extended in their functionality to support the segmentation of complex, spiralling vector fields in a robust manner.

The resulting data level segmentation allows application scientists to access their complex data piece by piece. Each segment is a canonical region and can have only one out of three simple shapes, namely strip, annular, or spiral [25], [27]. Therefore even complex data can be treated as a composition of elementary building blocks. Each block comes with a transversal that permits the extraction of quantitative properties, like volume, flow rate, or residence time. That allows analysis of the underlying physical phenomena that goes far beyond the traditional visualization of the separatrices.

We have illustrated the use of the segmentation in two application cases of importance to the environmental sciences: groundwater flow and heat exchange. The results highlight the potential of the proposed approach for improving the understanding and/or engineering of various types of systems.

In future work, we plan on improving the numerical stability of the Delaunay triangulation implementation so that the currently employed parameter steering the lengths of the barrier edges can be omitted. That will improve the user-friendliness of the workflow. We also plan to extend the segmentation algorithm to three-dimensional vector fields.

Acknowledgements

We gratefully acknowledge the support of the U.S. Department of Energy through the LANL Laboratory Directed Research Development Program under project number 20190143ER for this work. This work is published under LA-UR-21-21785. The authors also acknowledge support from the Korea Research Fellowship program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (grant 2016H1D3A1908042). We would like to thank the Kitware team for their support with the integration of our changes into VTK.

References

[1] J. D. Bredehoeft. The Toth revolution. *Groundwater*, 56(1):157–159, 2018.

TABLE I: Runtimes (s) of the proposed algorithm and of the calculation of forward and backward streamlines from all the cells of the grid.

Method	Proposed algorithm (step = 0.1)	Proposed algorithm (step = 0.5)	Streamlines (step = 0.1)	Streamlines (step = 0.5)
Groundwater	3.7	2.5	89.9	20.0
Heat exchange	5.1	3.9	17.9	14.2

TABLE II: Runtimes (s) of the different parts of the proposed algorithm (step = 0.1).

Part	A+B	C	D	E
Groundwater	0.29	0.96	0.01	2.43
Heat exchange	2.43	0.40	0.03	2.28

[2] E. Bresciani, T. Gleeson, P. Goderniaux, J. R. de Dreuzy, A. D. Werner, A. Wörman, W. Zijl, and O. Batelaan. Groundwater flow systems theory: Research challenges beyond the specified-head top boundary condition. *Hydrogeology Journal*, 24(5):1087–1090, 2016.

[3] R. Bujack, K. Tsai, S. K. Morley, and E. Bresciani. Open source vector field topology. *SoftwareX*, 15:100787, 2021.

[4] R. Bujack, L. Yan, I. Hotz, C. Garth, and B. Wang. State of the Art in Time-Dependent Flow Topology: Interpreting Physical Meaningfulness Through Mathematical Properties. *Computer Graphics Forum*, 2020.

[5] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 263–270. ACM, 1993.

[6] M. B. Cardenas and X.-W. Jiang. Groundwater flow, transport, and residence times through topography-driven basins with exponentially decreasing permeability and porosity. *Water Resources Research*, 46(11):W11538, 2010.

[7] 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, 2008.

[8] Y.-J. Chung, H.-S. Park, W.-J. Lee, and K.-K. Kim. Heat transfer in a cooling water pool with tube bundles under natural circulation. *Annals of Nuclear Energy*, 77:402–407, 2015.

[9] P. V. Danckwerts. Continuous flow systems - Distribution of residence times. *Chemical Engineering Science*, 2(1):1–13, 1953.

[10] W. De Leeuw and R. Van Liere. Collapsing flow topology using area metrics. In *Proceedings Visualization'99 (Cat. No. 99CB37067)*, pages 349–542. IEEE, 1999.

[11] J. W. Deardorff. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics*, 41:453–480, Jan. 1970.

[12] A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on*, pages 33–40. IEEE, 1991.

[13] H. Hauser, H. Hagen, and H. Theisel, editors. *Topology-based Methods in Visualization*. Mathematics and Visualization. Springer, 2007.

[14] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. A survey of topology-based methods in visualization. In *Computer Graphics Forum*, volume 35, pages 643–667. Wiley Online Library, 2016.

[15] J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *Computer*, 22(8):27–36, Aug 1989.

[16] X.-W. Jiang, X.-S. Wang, L. Wan, and S. Ge. An analytical study on stagnation points in nested flow systems in basins with depth-decaying hydraulic conductivity. *Water Resources Research*, 47(1):W01512, 2011.

[17] Kitware, Inc. *The Visualization Toolkit User's Guide*, January 2003.

[18] M. Langbein and H. Hagen. A Generalization of Moment Invariants on 2D Vector Fields to Tensor Fields of Arbitrary Order and Dimension. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, M. Encarnação, C. Silva, and D. Coming, editors, *Advances in Visual Computing*, volume 5876 of *Lecture Notes in Computer Science*, pages 1151–1160. Springer Berlin Heidelberg, 2009.

[19] R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology-based flow visualization, the state of the art. In *Topology-based methods in visualization*, pages 1–19. Springer, 2007.

[20] J. A. Levine, S. Jadhav, H. Bhatia, V. Pascucci, and P.-T. Bremer. A quantized boundary representation of 2d flows. In *Computer Graphics Forum*, volume 31, pages 945–954. Wiley Online Library, 2012.

[21] Z. Liu and R. J. Moorhead II. Robust loop detection for interactively placing evenly placed streamlines. *Computing in Science & Engineering*, 9(4):86–91, 2007.

[22] S. K. Lodha, J. C. Renteria, and K. M. Roskin. Topology preserving compression of 2d vector fields. In *Proceedings Visualization 2000. VIS 2000 (Cat. No. 00CH37145)*, pages 343–350. IEEE, 2000.

[23] K. Mahrous, J. Bennett, B. Hamann, and K. I. Joy. Improving topological segmentation of three-dimensional vector fields. In *VisSym*, volume 3, pages 203–212, 2003.

[24] K. Mahrous, J. Bennett, G. Scheuermann, B. Hamann, and K. I. Joy. Topological segmentation in three-dimensional vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):198–205, 2004.

[25] L. Markus. Global structure of ordinary differential equations in the plane. *Transactions of the American Mathematical Society*, 76(1):127–148, 1954.

[26] M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2d vector field topology. In *Computer Graphics Forum*, volume 29, pages 347–356. Wiley Online Library, 2010.

[27] L. Perko. *Differential equations and dynamical systems*, volume 7. Springer Science & Business Media, 2013.

[28] H. Pitsch. Large-Eddy Simulation of Turbulent Combustion. *Annual Review of Fluid Mechanics*, 38(1):453–482, Jan. 2006.

[29] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matkovic, and H. Hauser. The State of the Art in Topology-based Visualization of Unsteady Flow. *Computer Graphics Forum*, 30(6):1789–1811, September 2011.

[30] J. Reininghaus and I. Hotz. Combinatorial 2d vector field topology extraction and simplification. *Topological Methods in Data Analysis and Visualization*, pages 103–114, 2011.

[31] A. Ridluan and A. Tokuhiro. Benchmark simulation of turbulent flow through a staggered tube bundle to support cfd as a reactor design tool. part ii: Urans cfd simulation. *Journal of nuclear science and technology*, 45(12):1305–1315, 2008.

[32] C. Rossl and H. Theisel. Streamline embedding for 3d vector field exploration. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):407–420, 2011.

[33] S. Sane, R. Bujack, C. Garth, and H. Childs. A survey of seed placement and streamline selection techniques. In *Computer Graphics Forum*, volume 39, pages 785–809. Wiley Online Library, 2020.

[34] G. Scheuermann, H. Hagen, H. Krüger, M. Menzel, and A. Rockwood. Visualization of higher order singularities in vector fields. In *Proceedings of the 8th conference on Visualization'97*, pages 67–74. IEEE Computer Society Press, 1997.

[35] G. Scheuermann, B. Hamann, K. I. Joy, and W. Kollmann. Visualizing local vector field topology. *Journal of Electronic Imaging*, 9(4):356–367, 2000.

[36] G. Scheuermann, H. Kruger, M. Menzel, and A. P. Rockwood. Visualizing nonlinear vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109–116, 1998.

[37] W. J. Schroeder, B. Lorensen, and K. Martin. *The visualization toolkit: an object-oriented approach to 3D graphics*. Kitware, 2004.

[38] O. Simonin and M. Barcouda. Measurements and prediction of turbulent entering a staggered tube bundle. *4th Int. Symp. Applications of Laser Anemometry to Fluid Mechanics*, 5(23), 1988.

[39] C. Sun, Y. Li, H. Han, J. Zhu, S. Wang, and L. Liu. Experimental and numerical simulation study on the offshore adaptability of spiral wound heat exchanger in lng-fpso dmr natural gas liquefaction process. *Energy*, 189:116178, 2019.

[40] A. Szymczak and E. Zhang. Robust morse decompositions of piecewise constant vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):938–951, 2012.

[41] H. Theisel, C. Rössl, and H.-P. Seidel. Compression of 2d vector fields under guaranteed topology preservation. In *Computer Graphics Forum*, volume 22, pages 333–342. Wiley Online Library, 2003.

[42] H. Theisel, C. Rössl, and T. Weinkauf. Topological representations of vector fields. In *Shape Analysis and Structuring*, pages 215–240. Springer, 2008.

[43] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. On the Applicability of Topological Methods for Complex Flow Data. In H. Hauser, H. Hagen, and H. Theisel, editors, *Topology-based Methods in Visualization*, pages 105–120, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[44] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The topology toolkit. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):832–842, 2017.

[45] J. Tóth. A theoretical analysis of groundwater flow in small drainage basins. *Journal of Geophysical Research*, 68(16):4795–4812, 1963.

[46] J. Tóth. Groundwater as a geologic agent: An overview of the causes, processes, and manifestations. *Hydrogeology Journal*, 7(1):1–14, 1999.

[47] J. Tóth. *Gravitational systems of groundwater flow: theory, evaluation, utilization*. Cambridge, UK, 2009.

[48] J.-Z. Wang, A. Wörman, E. Bresciani, L. Wan, X.-S. Wang, and X.-W. Jiang. On the use of late-time peaks of residence time distributions for the characterization of hierarchically nested groundwater flow systems. *Journal of Hydrology*, 543, Part:47–58, 2016.

[49] X.-S. Wang, L. Wan, X.-W. Jiang, H. Li, Y. Zhou, J. Wang, and X. Ji. Identifying three-dimensional nested groundwater flow systems in a tóthian basin. *Advances in water resources*, 108:139–156, 2017.

[50] T. Weinkauf, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel. Extracting higher order critical points and topological simplification of 3D vector fields. In *VIS 05. IEEE Visualization, 2005.*, pages 559–566. IEEE, 2005.

[51] T. Wischgoll and G. Scheuermann. Detection and visualization of closed streamlines in planar flows. *Visualization and Computer Graphics, IEEE Transactions on*, 7(2):165–172, 2001.

[52] T. Wischgoll and G. Scheuermann. Locating closed streamlines in 3d vector fields. *methods*, 16:19, 2002.

[53] K. Wu, Z. Liu, S. Zhang, and R. J. Moorhead II. Topology-aware evenly spaced streamline placement. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):791–801, 2010.

[54] W. Zijl. Scale aspects of groundwater flow and transport systems. *Hydrogeology Journal*, 7(1):139–150, 1999.