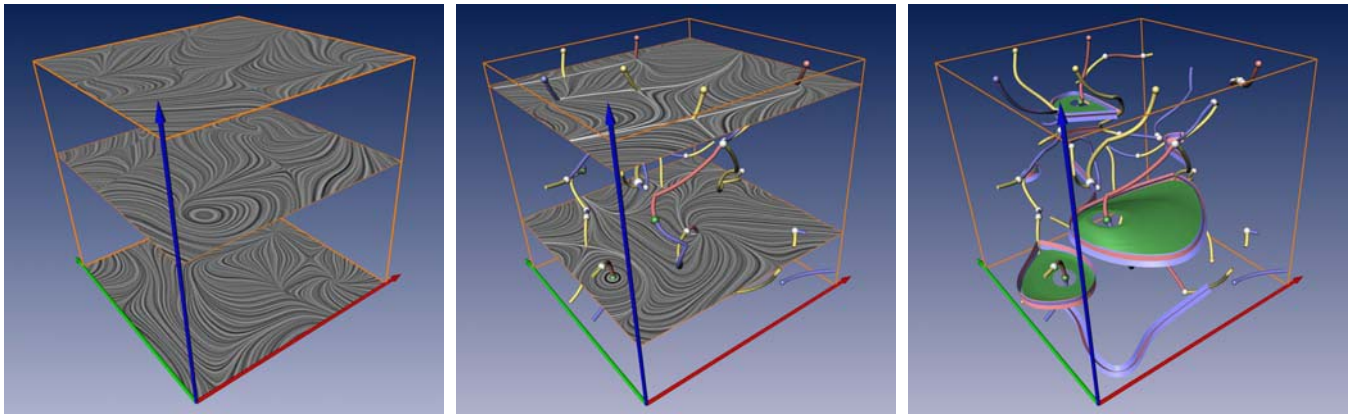
Stream Line and Path Line Oriented Topology for 2D Time-Dependent Vector Fields

Holger Theisel [*]
MPI Informatik Saarbrücken

Tino Weinkauf [†]
Zuse Institute Berlin

Hans-Christian Hege [‡]
Zuse Institute Berlin

Hans-Peter Seidel [§]
MPI Informatik Saarbrücken



(a) LIC images at 3 different time slices.

(b) Tracking the locations of critical points as stream lines (red/blue/yellow); local bifurcations: Hopf bifurcations (green spheres), fold bifurcations (gray spheres).

(c) Global bifurcations: saddle connections (red/blue flow ribbons), tracked closed stream lines (green surfaces).

Figure 1: Stream line oriented topology of a 2D time-dependent vector field.

Abstract

Topological methods aim at the segmentation of a vector field into areas of different flow behavior. For 2D time-dependent vector fields, two such segmentations are possible: either concerning the behavior of stream lines, or of path lines. While stream line oriented topology is well established, we introduce path line oriented topology as a new visualization approach in this paper. As a contribution to stream line oriented topology we introduce new methods to detect global bifurcations like saddle connections and cyclic fold bifurcations. To get the path line oriented topology we segment the vector field into areas of attracting, repelling and saddle-like behavior of the path lines. We compare both kinds of topologies and apply them to a number of data sets.

CR Categories: I.3.3 [Computer Graphics]: Line and Curve Generation I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

Keywords: flow visualization, vector field topology, bifurcations, stream lines, path lines

[*]e-mail: theisel@mpi-sb.mpg.de
[†]e-mail: weinkauf@zib.de
[‡]e-mail: hege@zib.de
[§]e-mail: hpseidel@mpi-sb.mpg.de

IEEE Visualization 2004
October 10-15, Austin, Texas, USA
0-7803-8788-0/04/$20.00 ©2004 IEEE

1 Introduction

Topological methods have become a standard tool in vector field Visualization. Initially introduced as a visualization tool in [9], topological methods have been extended to higher order critical points [15], boundary switch points [3], and closed separatrices [26]. In addition, topological methods have been applied to simplify [3] [4] [20] [21], smooth [25], compress [11] and design [17] vector fields. The topology of 3D vector fields is visualized in [6], [12], [13], [19], [24].

For 2D time-dependent vector fields there exists a number of extensions of topological concepts. [22] and [23] track the location of critical points over time and detect local bifurcations like fold bifurcations and Hopf bifurcations. This approach works on a piecewise linear vector field and computes and connects the critical points on the faces of a prism cell structure, which is constructed from the underlying triangular grid. [26] introduce a method to detect closed stream lines in 2D steady vector fields. This method also relies on the underlying triangular grid of the piecewise linear vector field. [27] extends this to 2D time-dependent vector fields by applying an approach similar to contouring & connecting of isosurfaces: closed stream lines are extracted for every time step, then corresponding lines in adjacent time steps are connected. As every contouring & connecting approach, special attention has to be paid to events like appearance, disappearance or collapsing of closed stream lines. [18] introduces a method to track critical points in a time-dependent vector field which does not depend on a particular underling grid: the paths of the critical points are tracked as the stream lines of a new vector field called feature flow field which can be extracted from the original vector field.

The main motivation behind topological methods is to segment a vector field into areas of similar flow behavior which is determined

by observing the behavior of certain characteristic curves. For 2D time-dependent vector fields, two important classes of curves exist: stream lines and path lines. Hence, two different kinds of topologies can be considered: a stream line oriented topology where areas are segmented which show a similar behavior of stream lines, and a path line oriented topology which does so for path lines. All above-mentioned 2D time-dependent topological methods are stream line oriented.

Based on this distinction of topologies, this paper makes two major contributions: for stream line oriented methods, we propose new approaches to detect global bifurcations like saddle connections and cyclic fold bifurcations. In addition we propose a method to detect and track closed stream lines which does not depend on an underlying grid structure and which does not have to explicitly solve the correspondence between adjacent time steps. The second major contribution is the consideration of path line oriented topology. This kind of topology has not been considered in the visualization community yet. For this, we divide the vector field into areas where the path lines show attracting, repelling, or saddle-like behavior respectively.

The rest of the paper is organized as follows: section 2 recalls the concepts of stream lines and path lines and gives a setup to distinguish and analyze them. Section 3 treats the streamline oriented topology: after reviewing the most important concepts and previous work (section 3.1), we introduce a new method to detect saddle connections (section 3.2). Section 3.3 presents a new method to track closed stream lines which is independent of an underlying grid and robust against cyclic fold bifurcations. Section 4 presents the approach to a path line oriented topology by segmenting the vector field into areas of different path line behavior. In section 5 we apply both concepts of topology to two data sets. Conclusions are drawn in section 6.

2 Stream Lines and Path Lines

Given a 2D time-dependent vector field

$$\mathbf{v}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix}, \tag{1}$$

where (x,y) describe the 2D domain and t is the temporal component, stream and path lines are generally different classes of curves. Stream lines are the tangent curves of $\mathbf{v}$ for a fixed time t, while path lines describe the paths of massless particles in $\mathbf{v}$ over time. To capture both types of lines, we define two 3D vector fields and consider their topological behavior.

To treat stream lines of $\mathbf{v}$, we consider

$$\mathbf{s}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \\ 0 \end{pmatrix}. \tag{2}$$

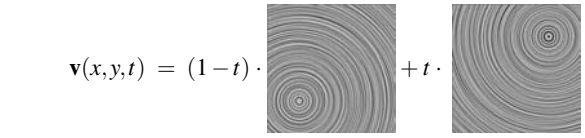
$\mathbf{s}$ can be seen as a steady 3D vector field. Its stream lines coincide with the stream lines of $\mathbf{v}$, since any integration step in $\mathbf{s}$ keeps the time component unchanged. Any (x,y)-slice through $\mathbf{s}$ represents $\mathbf{v}$ at a constant time.

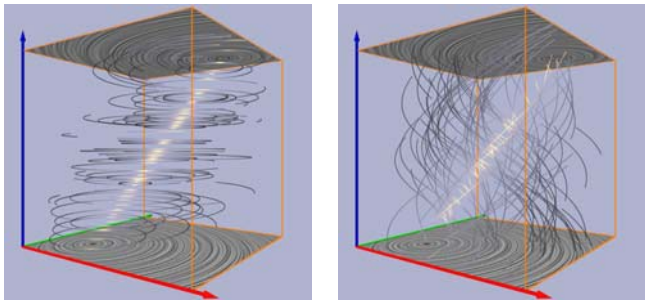
To capture the path lines of $\mathbf{v}$, we consider the 3D vector field

$$\mathbf{p}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \\ 1 \end{pmatrix} \tag{3}$$

where the stream lines of $\mathbf{p}$ coincide with the path lines of $\mathbf{v}$: given a starting point (x_0,y_0,t_0), one step of a simple Euler approximation of $\mathbf{p}$ would be

$$\begin{pmatrix} x_1 \\ y_1 \\ t_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ t_0 \end{pmatrix} + d\ \mathbf{p}(x_0,y_0,t_0) = \begin{pmatrix} x_0 + d\ u(x_0,y_0,t_0) \\ y_0 + d\ v(x_0,y_0,t_0) \\ t_0 + d \end{pmatrix} \tag{4}$$





(a) Stream lines of $\mathbf{s}$ correspond to the stream lines in $\mathbf{v}$.

(b) Stream lines of $\mathbf{p}$ correspond to the path lines in $\mathbf{v}$.

Figure 2: Characteristic curves of a simple 2D time-dependent vector field shown as illuminated field lines.

which does not only change the location but also goes forward in time.[1] Figure 2 illustrates $\mathbf{s}$ and $\mathbf{p}$ for a simple example vector field $\mathbf{v}$. It is obtained by a linear interpolation over time of two bilinear vector fields. Note that in all figures throughout this paper the coordinate system is shown as follows: red/green coordinate axes denote the (x,y)-domain, the blue axis shows the time component.

Now the problem of finding a stream line and path line oriented topology is simply reduced to finding the topological skeletons of $\mathbf{s}$ and $\mathbf{p}$. Unfortunately, neither for $\mathbf{s}$ nor for $\mathbf{p}$ the classical vector field topology extraction techniques for 3D vector fields are applicable: $\mathbf{s}$ consists of critical lines while $\mathbf{p}$ does not have any critical points at all. Sections 3 and 4 show how to treat the topological skeleton

For 2D time-dependent vector fields there exist also other classes of characteristic curves, namely streak lines and time lines. However, they cannot be transformed to a 3D vector field similar to $\mathbf{s}$ and $\mathbf{p}$ because they are not locally unique: for a particular location and time, there is more than one streak line and time line passing through. Because of this we restrict ourselves to stream line and path line oriented topology.

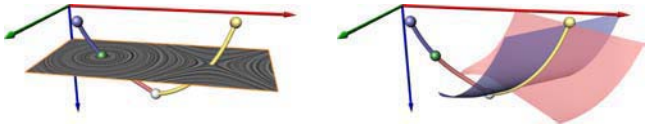
3 Stream Line Oriented Topology

Stream line oriented topology is well-understood in the visualization community ([10], [1], [2]). In addition to track the topological features over time, bifurcations have to be extracted. Bifurcations are the events of structural changes of the flow behavior at a certain time. We first review the most important concepts and approaches for their visualization (section 3.1) before we introduce new methods to detect certain global bifurcations: saddle connections (3.2) and cyclic fold bifurcations (3.3).

3.1 Concepts and previous work

Critical points are important topological features of steady vector fields. Tracking their location over time is necessary for capturing the topological behavior of $\mathbf{v}$. This is equivalent to extracting the zero lines of $\mathbf{s}$, where all points on these lines are zero points of $\mathbf{v}$ at a certain time. To do so, one can either extract and connect the zeros on the faces of an underlying prism cell grid ([22]), or a feature flow field integration from a start zero point of $\mathbf{s}$ is applied.

[1] For all algorithms in this paper we used a fourth-order Runge-Kutta integration in our implementation.



(a) Critical lines of **s**, LIC plane through Hopf bifurcation.

(b) Separation surfaces created by the moving saddle.

Figure 3: Topological visualization of a simple 2D time-dependent vector field consisting of sink, source, saddle, fold and Hopf bifurcation – one of each type.

The feature flow field for tracking critical points is a 3D vector field **f** which points into the direction where all components of **s** remain unchanged. [18] shows that

$$\mathbf{f}(x,y,t) = \begin{pmatrix} \det(\mathbf{v}_y, \mathbf{v}_t) \\ \det(\mathbf{v}_t, \mathbf{v}_x) \\ \det(\mathbf{v}_x, \mathbf{v}_y) \end{pmatrix}. \tag{5}$$

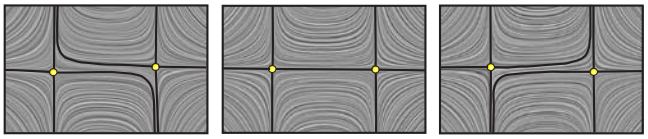
Starting a stream line integration of **f** from a point $\mathbf{x}_0$ with $\mathbf{s}(\mathbf{x}_0) = (0,0,0)^T$, all points **x** on this stream line fulfill $\mathbf{s}(\mathbf{x}) = (0,0,0)^T$ as well. Here we prefer the feature flow field approach to extract the critical lines of **s** since it does not depend on an underlying grid.

To extract all critical lines of **s**, an appropriate number of start points is needed. We get them by considering all critical points at the boundaries of the domain of **s** (which can easily be obtained as critical points of 2D vector fields) and by additionally considering all *fold bifurcations* of **v**. A fold bifurcation appears if at a certain time t a critical point appears and in the same moment splits up to a saddle and source/sink/center.[2] Fold bifurcations can be found as the zeros of the following system of equations: $[u = 0,\ v = 0,\ \det(\mathbf{v}_x, \mathbf{v}_y) = 0]$. This refers to local extrema of the t-values on the critical lines. To solve this system for isolated (x,y,t), we use a simple subdivision approach similar to detecting isolated critical points in 3D vector fields: a cell C in the domain is checked whether one of the components (u, v, or $\det(\mathbf{v}_x, \mathbf{v}_y)$) is positive at all 8 vertices of C (or whether one component is negative at all vertices). If so, no fold bifurcation is found in C. Otherwise we recursively subdivide C into 8 subcells until their size is smaller than a certain threshold. The results are clusters of small cells which represent the isolated fold bifurcations.

Another important class of local bifurcations are *Hopf bifurcations* denoting locations where a sink becomes a source or vice versa. Thus, this denotes the location of a center, i.e. a critical point with a vanishing divergence and a positive Jacobian. Hopf bifurcations can be extracted similar to fold bifurcations by numerically solving the system $[u = 0,\ v = 0,\ \mathrm{div}(\mathbf{v}) = u_x + v_y = 0]$ for (x,y,t) and selecting all isolated solutions with positive Jacobian.

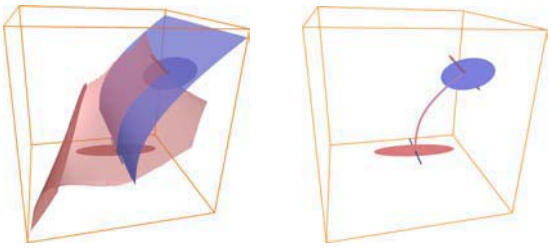
Another part of the topological skeleton of **v** are the separation curves starting from saddle points. It is a well-known fact that a saddle of a 2D vector field creates 4 separation curves by starting the integration into the directions of the eigenvectors of the Jacobian matrix. While the saddle moves over time in **v**, their sweepings form 4 stream surfaces dividing **s** into areas of different flow behavior. Figure 3 gives an illustration of a simple vector field containing all topological feature mentioned above. In this figure (as well as in the following figures) we use the following color coding: the critical lines of **s** are color coded according to the inflow/outflow behavior of the represented critical points in **v**: a red/blue/green/yellow line segment represents a source/sink/center/saddle critical point respectively. The same color coding is used for particular critical points which are visualized as small spheres. This means that a Hopf bifurcation is shown as a small green sphere. Furthermore, fold bifurcations are shown as gray sphere, while particular stream lines of **s** are shown as gray lines. For integrated separation surfaces we color code according to the integration direction as red (forward integration) or blue (backward integration) surfaces.

[2] Or the other way around: a saddle and a source/sink/center collapse and disappear.



(a) Shortly before. (b) The event. (c) Shortly after.

Figure 4: Saddle connection bifurcation.



(a) Separation surfaces of the saddles.

(b) Their intersection curve is the saddle connector.

Figure 5: Definition of saddle connectors for 3D vector fields.

3.2 Detecting saddle connections

Saddle connections are global bifurcations which appear when two separatrices starting from saddle points collapse, i.e. when a separatrix of one saddle ends in another saddle. Figure 4 illustrates an example. We are not aware of pre-existing solutions to extracting all saddle connections of **v** for visualization purposes.

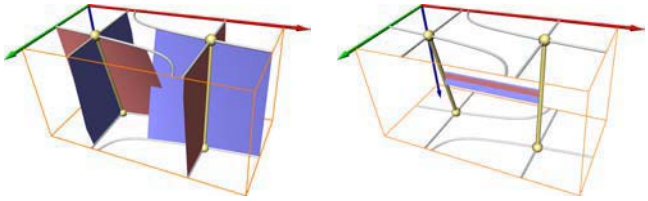
The solution we propose here is an adaption of the saddle connectors approach [19] which recently has been introduced to visualize topological skeletons of 3D steady vector fields.[3] Saddle connectors are the intersection curves of the separation surfaces of a 3D vector field starting in the outflow and inflow planes of the saddle points. The basic idea is to numerically integrate two separation surfaces until an intersection is found. After some refinement, a stream line is integrated from the intersection point both in forward and backward direction. Figure 5 illustrates the idea of saddle connectors, details about implementation, accuracy and speed are in [19].

Now the idea of saddle connectors can be modified to detect saddle connections in **s**: instead of starting the integration of one separation surface at each saddle of a 3D vector field, we start in the critical lines of **s** which represent a moving saddle. In fact, we start four stream surface integrations[4] at the critical lines of **s** into the directions of the eigenvectors of the Jacobian matrix. The rest of the algorithm is similar to saddle connectors and yields all saddle connections in **v**. Figure 6 gives an illustration.

A special case of saddle connections is the so-called *periodic blue sky bifurcation* ([1]) where two separatrices of the same saddle

[3] Note the nice coincidence of concepts: saddle connections (of 2D time-dependent vector fields) are extracted using an adaption of saddle connectors (for 3D steady vector fields). Though both concepts are not the same: the 3D counterpart to a saddle connection bifurcation is the collapsing of two separation surfaces and this contradicts the definition of 3D saddle connectors as curves (see [19]).

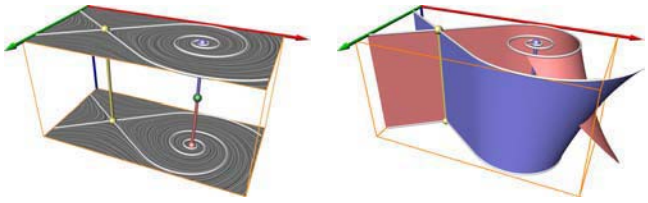
[4] Two forwards and two backwards.



(a) Separation surfaces starting from critical lines of **s**.

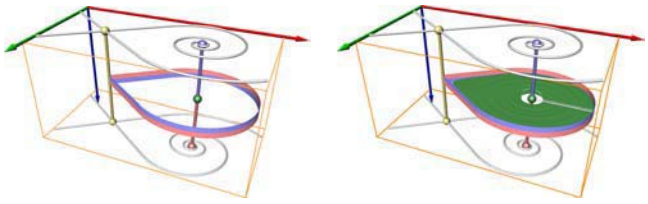
(b) Saddle connection as the intersection of these surfaces.

Figure 6: Extracting saddle connections.



(a) Critical lines of **s** and two LIC planes.

(b) Separation surfaces shortly after their intersection.



(c) Two separation curves of the same saddle collapse.

(d) Tracked closed stream line starting from Hopf bifurcation.

Figure 7: Periodic blue sky bifurcation.

collapse. Our algorithm to extract saddle connections automatically extracts these bifurcations as well. Figure 7 illustrates this.

3.3 Tracking closed streamlines

Closed stream lines are global topological features which evolve over time in **v**. Doing so, several bifurcations can occur: a closed stream line may appear or disappear, or two closed stream lines may collapse and disappear. The last case is called *cyclic fold bifurcation* and is illustrated in figures 8 and 10.

A first approach to track closed stream lines was proposed in [22]: closed stream lines are extracted in different time levels, and corresponding stream lines in adjacent time levels are connected. The results are tube-shaped surfaces starting/ending in Hopf bifurcations, periodic blue sky bifurcations, or at the boundaries of the domain. This approach depends on the underlying grid structure and does not consider cyclic fold bifurcations.

The new solution we propose here follows the general idea of feature flow fields [18]. Suppose we have a closed stream line $\mathbf{c}_i$. We want to construct a 3D vector field $\mathbf{g}$ such that the evolution of $\mathbf{c}_i$ over time can simply be obtained by a stream surface integration of $\mathbf{g}$ starting at $\mathbf{c}_i$. Unfortunately, such a feature flow field $\mathbf{g}$ cannot locally be derived from **s**, since the evolution of a closed stream line is a global process. To solve this problem, we apply an adaption of the concept of saddle connectors [19] again: given $\mathbf{c}_i$, we want to find an adjacent closed stream line $\mathbf{c}_{i+1}$ within a certain distance d from $\mathbf{c}_i$. Note that d refers to the (x,y,t)-domain, i.e. we do not make any assumption whether the t-value of $\mathbf{c}_{i+1}$ is before, after, or at the same t-value as $\mathbf{c}_i$. Also note that a closed stream line is uniquely defined by a single point on it. Hence, we only have to

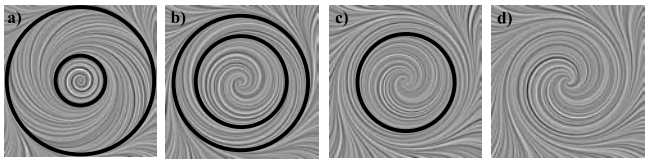


Figure 8: Cyclic fold bifurcation: two closed stream lines move towards each other (a and b), merge (c) and disappear (d).

construct a point $\mathbf{x}_{i+1}$ on $\mathbf{c}_{i+1}$ with a certain distance from a point $\mathbf{x}_i$ on $\mathbf{c}_i$.

We describe one step of our algorithm now (Figures 9b-d). Given two adjacent closed stream lines $\mathbf{c}_{i-1}$ and $\mathbf{c}_i$ together with their defining points $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ (Figure 9b), we want to find a point $\mathbf{x}_{i+1}$ which defines the next closed stream line $\mathbf{c}_{i+1}$. To do so, we consider a plane through $\mathbf{x}_i$ perpendicular to $\mathbf{s}(\mathbf{x}_i)$ and place a circle $\mathbf{k}$ around $\mathbf{x}_i$ with the radius d into this plane: $\mathbf{x}_{i+1}$ is searched on $\mathbf{k}$. Since we assume the closed stream line to evolve continuously, we search $\mathbf{x}_{i+1}$ only on a circular arc $\hat{\mathbf{k}}$ of $\mathbf{k}$ consisting of all points $\mathbf{x} \in \mathbf{k}$ with $(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}_i-\mathbf{x}_{i-1}) > 0$ (Figure 9b). This ensures, that the algorithm does not run back to $\mathbf{c}_{i-1}$. $\hat{\mathbf{k}}$ is further subdivided in t-direction into two arcs $\mathbf{h}_1$ and $\mathbf{h}_2$. [5] They act as seeding curves of a stream surface integration of **s** in both forward and backward direction (Figure 9c). Similar to the saddle connector approach, we detect the intersection of these stream surfaces: it describes a closed stream line (Figure 9d).

Assuming $\text{length}(\mathbf{h}_1) \geq \text{length}(\mathbf{h}_2)$, we first apply a stream surface integration of **s** starting in $\mathbf{h}_1$ both in forward and backward direction, and check them for an intersection curve. If an intersection curve is found, this is the new closed stream line $\mathbf{c}_{i+1}$; its intersection with $\mathbf{h}_1$ gives $\mathbf{x}_{i+1}$. If no intersection curve is found, we check $\mathbf{h}_2$ in a similar way. If this gives no intersection neither, the algorithm stops.

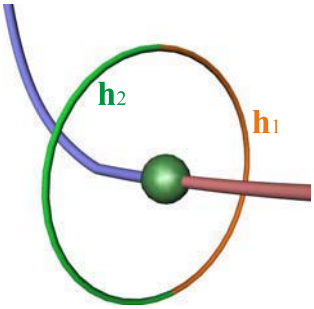
After extracting a sequence of closed stream lines, there are two ways for extracting the stream surface describing them. One is to use the curve $[\mathbf{x}_0,\ldots,\mathbf{x}_n]$ as a seeding line for another stream surface integration. However, since we already extracted a number of closed stream lines, we use them to triangulate the strips between each adjacent $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$. The result is a triangular mesh representing a stream surface describing the evolution of a closed stream line over time.

To complete the algorithm, we have to describe how to start the integration at a Hopf or periodic blue sky bifurcation. At a Hopf bifurcation $\mathbf{x}_0$, we start the integration in a semi-circle around $\mathbf{x}_0$ which lies in the plane defined by $(0,0,1)^T$ and $(0,0,1)^T \times \mathbf{f}(\mathbf{x}_0)$, where $\mathbf{f}$ is the feature flow field for tracking critical points (5). Figure 9a illustrates this. At a periodic blue sky bifurcation, we take any point on the stream line as starting point $\mathbf{x}_0$. Setting $\hat{\mathbf{k}}$ as the full circle $\mathbf{k}$, the rest of the integration step is as described above.

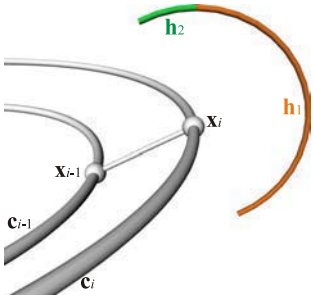
Figure 10 shows an example of tracking a closed stream line starting in a Hopf bifurcation. We see the sequence of defining points $\mathbf{x}_i$ as well as the searching arcs $\mathbf{h}_1$ and $\mathbf{h}_2$ in every step. This example shows that the algorithm can deal with cyclic fold bifurcations: it appears at the closed stream line $\mathbf{c}_i$ if $(t_i - t_{i-1})(t_{i+1} - t_i) \leq 0$ (where t_i is the t-component of the points on $\mathbf{c}_i$). The example in Figure 10 has one cyclic fold bifurcation (gray line).

Another example of a tracked closed stream line is in figure 7d. Here, a Hopf bifurcation and a periodic blue sky bifurcation have the same t-value. It turns out that the closed stream line completely

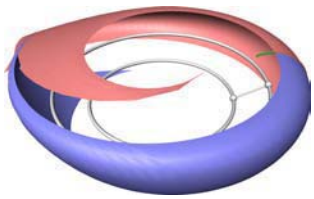
[5] The splitting of $\hat{\mathbf{k}}$ into $\mathbf{h}_1$ and $\mathbf{h}_2$ is necessary to ensure that each found connector is a closed stream line. If we would start the integration on $\hat{\mathbf{k}}$, false connectors starting and ending in different points are possible, since two points on $\hat{\mathbf{k}}$ may have the same t-value.



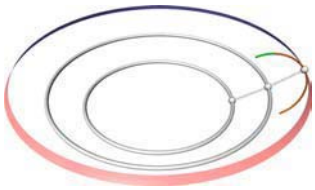
(a) Setup for starting at Hopf bifurcation.



(b) Seeding curves $\mathbf{h}_1$ and $\mathbf{h}_2$ for finding the closed stream line $\mathbf{c}_{i+1}$.

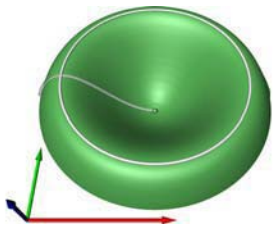


(c) Intersecting stream surfaces evolving from $\mathbf{h}_1$.

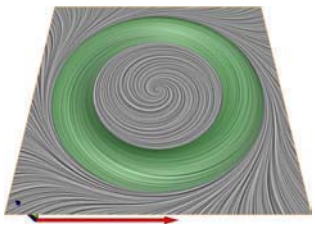


(d) Detected connector is $\mathbf{c}_{i+1}$.

Figure 9: Tracking closed stream lines.



(a) Stream surface describing closed stream lines. Cyclic fold bifurcation is detected on top of this surface (gray line).



(b) Intersection of stream surface with LIC plane. Compare with Figure 8b.

Figure 10: Detecting a cyclic fold bifurcation.

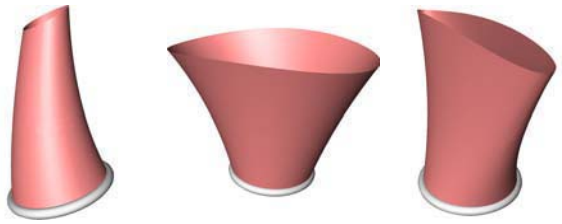
evolves in the same time-slice. This is an example where tracking approaches fail which are based on extracting and connecting closed stream lines in fixed time steps.

The numerical integration of the seeding line $[\mathbf{x}_0,\ldots,\mathbf{x}_n]$ has the advantage that numerical errors are not inherited: if due to numerical errors $\mathbf{x}_i$ is located a little bit away from a closed stream line, $\mathbf{x}_{i+1}$ can still be found correctly.

4 Path Line Oriented Topology

Constructing a path line oriented topology means to consider the stream lines of $\mathbf{p}$ (cf. equation (3)) and segment $\mathbf{p}$ into regions of different flow behavior of them. Although an extensive research about extracting characteristic structures of path lines ([7], [8]) and local fluid particle motion ([14]) has been done in the fluid dynamics community, topological approaches of path lines have not been used as a visualization tool yet.

Given a 3D vector field $\mathbf{w}$, it is a well-known fact that a number of local properties (like acceleration, shear, curvature, convergence and others) can be computed by a local analysis of the Jacobian matrix $\mathbf{J}_\mathbf{w}$ ([14]). The usual approach to compute them is a decomposition of $\mathbf{J}_\mathbf{w}$ together with introducing a local coordinate system.



(a) Attracting. (b) Repelling. (c) Saddle-like.

Figure 11: Behavior of stream lines of $\mathbf{p}$ starting on a circle $\mathbf{k}$ around $\mathbf{x}_0$.

These properties also have been used in icon-based visualization tools for 3D vector fields [5]. The local properties of $\mathbf{w}$ can be divided into two groups: the first group consist of pure directional properties of the stream lines of $\mathbf{w}$, i.e. they are invariant under scalings of $\mathbf{w}$. The second group consists of properties which depend both on the length and the direction of $\mathbf{w}$.

The basic approach we use here is to exploit local properties of a 3D vector field to get a segmentation of $\mathbf{p}$. Since a topological skeleton should only depend on the direction of the stream lines, we focus here on properties of the first group. In fact, we aim at segmenting $\mathbf{p}$ into areas of attracting, repelling, or saddle like behavior of the stream lines of $\mathbf{p}$.

Consider a point $\mathbf{x}_0$ with its corresponding vector $\mathbf{p}(\mathbf{x}_0)$ defining a stream line $\mathbf{l}_0$ in $\mathbf{p}$ starting at $\mathbf{x}_0$. We consider a small circle $\mathbf{k}_0$ around $\mathbf{x}_0$ in the plane perpendicular to $\mathbf{p}(\mathbf{x}_0)$. Considering all stream lines $\mathbf{l}$ in $\mathbf{p}$ starting in $\mathbf{k}_0$, three stable cases are possible concerning their convergence/divergence behavior towards $\mathbf{l}_0$:

- All $\mathbf{l}$ converge towards $\mathbf{l}_0$ under forward integration.
- All $\mathbf{l}$ move away from $\mathbf{l}_0$ under forward integration.
- Both stream lines with converging and diverging behavior exist (saddle-like behavior).

Figure 11 illustrates these cases.

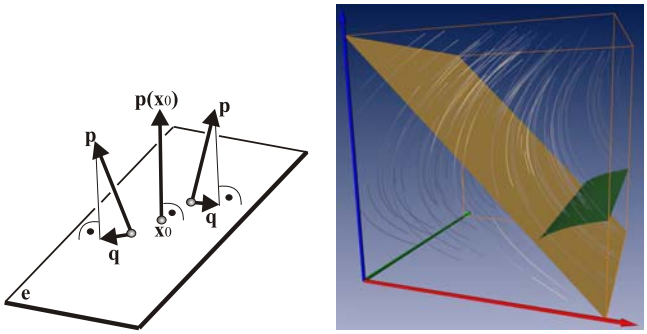
To compute this classification as a local property of $\mathbf{x}_0$, we transform $\mathbf{p}$ into a local coordinate system such that the new origin is $\mathbf{x}_0$ and the new base vectors $\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3$ are

$$\mathbf{b}_3 = \frac{\mathbf{p}(\mathbf{x}_0)}{\|\mathbf{p}(\mathbf{x}_0)\|} \quad , \quad \mathbf{b}_1 = \frac{(0,-1,v)^T}{\sqrt{1+v^2}} \quad , \quad \mathbf{b}_2 = \mathbf{b}_3 \times \mathbf{b}_1. \tag{6}$$

Note that $\mathbf{b}_1$ is chosen to be non-zero and perpendicular to $\mathbf{b}_3$. Any other choice of $\mathbf{b}_1$ fulfilling these demands is possible as well. Since we are only interested in local properties perpendicular to the flow direction, we consider the plane $\mathbf{e}$ through $\mathbf{x}_0$ defined by $\mathbf{b}_1$ and $\mathbf{b}_2$, and project the vectors $\mathbf{p}(\mathbf{e})$ into $\mathbf{e}$. Doing so we get a 2D vector field $\mathbf{q}$ in $\mathbf{e}$ which has a critical point in $\mathbf{x}_0$. Figure 12a illustrates the definition of $\mathbf{q}$. To get the flow behavior of $\mathbf{p}$ in $\mathbf{x}_0$, we can classify $\mathbf{x}_0$ in $\mathbf{q}$ by an eigenvalue analysis of the Jacobian matrix $\mathbf{J}_\mathbf{q}$ of $\mathbf{q}$ at $\mathbf{x}_0$. Computing the eigenvalues e_1,e_2 of $\mathbf{J}_\mathbf{q}(\mathbf{x}_0)$ we get by a straightforward exercise in algebra

$$e_1\,e_2 = \frac{\det(\mathbf{p},\mathrm{grad}(u),\mathrm{grad}(v))}{\mathbf{p}^2} \tag{7}$$

$$e_1+e_2 = u_x+v_y-\frac{u\cdot(\mathbf{p}\cdot\mathrm{grad}(u))+v\cdot(\mathbf{p}\cdot\mathrm{grad}(v))}{\mathbf{p}^2} = \|\mathbf{p}\|\cdot\mathrm{div}\left(\frac{\mathbf{p}}{\|\mathbf{p}\|}\right) \tag{8}$$



(a) Finding the projected vector field **q**.

(b) Example of **p** consisting of three sectors.

Figure 12: Computing path line oriented topology.

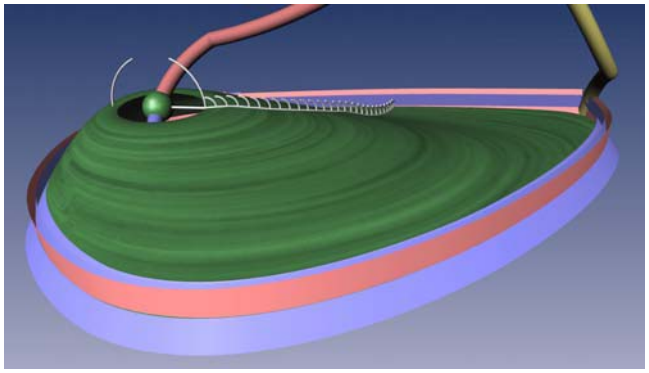


Figure 13: Test data set: Closeup of a tracked closed stream line.

where $\mathrm{grad}(u) = (u_x, u_y, u_t)^T$ and $\mathrm{grad}(v) = (v_x, v_y, v_t)^T$ are the gradients of the u- and v-components respectively. [6]

To separate regions where **p** has a saddle-like behavior, we extract the isosurface $e_1 e_2 = 0$: areas with $e_1 e_2 < 0$ reveal a saddle-like behavior. In areas with $e_1 e_2 > 0$ we further check whether a complete attracting or repelling behavior is present. We do so by extracting the isosurface $e_1 + e_2 = 0$: $e_1 + e_2 > 0$ denotes repelling behavior while $e_1 + e_2 < 0$ gives attracting behavior of **p**. Figure 12b shows a simple example of **p** consisting of one attracting, one repelling and one saddle-like sector.

Note that this segmentation of **p** can also be achieved by considering the Gaussian and Mean curvature of a surface **z** through $\mathbf{x}_0$ with all its normals in the direction of **p**: a saddle-like behavior of **p** corresponds to a negative Gaussian curvature of **z**, a repelling/attracting behavior corresponds to a positive/negative Mean curvature. Also note that the isosurfaces $e_1 e_2 = 0$ and $e_1 + e_2 = 0$ separating different sectors in **p** are not stream surfaces of **p**. This means that a stream line in **p** can travel through sectors of different flow behavior.

Contrary to the stream line oriented topology, the path line oriented topology is not invariant under scalings of **v**. In fact, the path lines of two vector fields $\mathbf{v}(x,y,t)$ and $c \cdot \mathbf{v}(x,y,t)$ with $c > 0$ and $c \neq 1$ differ. The factor c influences the impact of the temporal changes with respect to the spatial changes. For large c, the flow is dominated by the spatial changes. For $c \to +\infty$, the stream lines of **p** converge to the stream lines of **s**.

5 Application and Results

We applied stream line oriented and path line line oriented methods to a number of test data sets. Not surprisingly, not all topological features appear in all data sets, and different topological features turned out to be important for different data sets.

Figures 1 and 13 show a stream line oriented topological visualization of a random 2D time-dependent data set on a $5 \times 5 \times 5$ grid. Random vector fields are useful tools for a proof-of-concept of topological methods, since they contain a maximal amount of topological information. Figure 1a shows LIC images of the vector field at three different time slices which already indicates a high topological complexity. Figure 1b shows parts of the stream line oriented topological skeleton. We detected 18 critical lines of **s** (shown in red/blue/yellow, according to their outflow/inflow/saddle behavior), 32 fold bifurcations (gray spheres), and 4 Hopf bifurcations (green spheres). In this figure we also included two LIC planes to illustrate the relation between the critical lines of **s** and the critical points of **v**. In addition, figure 1c shows 8 detected saddle connections, among them 4 periodic blue sky bifurcations. We visualized them as red/blue double flow ribbons describing the orientation of the intersecting separation surfaces which create them. Starting from the Hopf bifurcations, we tracked the closed stream lines of **s**: each closed stream line starting in a Hopf bifurcation turned out to end in a periodic blue sky bifurcation. The resulting surfaces are shown in green. Figure 13 shows a detail of figure 1c to illustrate the tracking of closed stream lines: also shown are the seeding arcs for each step of the integration.

The computing time for extracting the saddle connections in this example was 20 seconds on a Pentium 4 1.7 GHz. For this, 42 attracting and 42 repelling stream surfaces had to be integrated and checked for intersections. For tracking the closed stream lines, our algorithm took 14 seconds on the same hardware. For this, 52 steps of the described algorithm had been carried out.

Figure 14 shows the visualization of a 2D time-dependent flow behind a circular cylinder. The cylinder is in the (x,y) plane around the origin of the underlying coordinates system. This data set was kindly provided by Gerd Mutschke (FZ Rossendorf) and Bernd R. Noack (TU Berlin). Figures 14a and 14b show the stream lines of **s** and **p** as illuminated stream lines [28]. As we can see in figure 14a, this non-compressible flow does not contain critical points (except for a center and a saddle directly behind the cylinder). This means that stream line oriented techniques fail for this data set. However, path line oriented techniques are still applicable. Figure 14e shows the path line oriented skeleton which reveals quasi-periodic structures behind the cylinder. These structures move slowly away from the cylinder over time. They correspond to the well-known von Kármán vortex street. To make this data set applicable to stream line oriented techniques, we subtract a constant vector field[7] which leads to the appearance of critical points. Figure 14c shows the stream line oriented skeleton of this modified data set which consists of 47 critical lines and 13 fold bifurcations. Note that the critical lines appear only in green and yellow, indicating that only moving centers and saddles are present. This corresponds to the fact that the vector field describes an incompressible flow. Figure 14c also shows that the critical points slowly move away from the cylinder over time: the critical lines of **s** are in general not parallel to the time axis. Figure 14d shows a close-up, where the separation surfaces emanating from the moving saddles are visualized. As we can see here, a number of separation surfaces tend to collapse making it impossible to extract isolated saddle connections. This is also due

[6] Note that instead of (6) we can also use the local Frenet frame as a local coordinates system. However, since we apply only an eigenvalue analysis of the Jacobian matrix perpendicular to the flow direction, the results (7) and (8) are the same.

[7] This trick is well-known in the fluid dynamics community. It is motivated by the idea that the observer is moving with the flow.

to the fact that the vector field is non-compressible.

Figure 14 shows that for our particular data set stream line and path line oriented topology reveal rather similar structures. We see the reason of this in the fact that the temporal changes in this data set are marginal in comparison to the spatial changes which dominate this flow.

6 Conclusions

In this paper we made the following contributions:

- We introduced an approach to extracting all saddle connections of a 2D time-dependent vector field.
- We introduced an approach to track closed stream lines which is robust against cyclic fold bifurcations.
- We introduced an approach to a path line oriented topology by distinguishing sectors of attracting, repelling and saddle-like behavior of the path lines.

The application to a number of test data sets shows that stream line and path line oriented topology reveal different structural properties of vector fields. While the stream line oriented topology is dominated by the presence and movements of critical points, path line oriented topology can be interpreted as focusing on mixing properties of a flow: areas with a saddle-like behavior of the path lines indicate a good mixing property of the flow over time. In particular, path line oriented topology also gives a segmentation in areas where a stream line oriented topology fails due to the absence of critical points.

We consider this paper as a starting point for further research on visualizing path line oriented topological features. Since our approach is based on an isosurface segmentation, we did not consider events in the path line flow yet.

7 Acknowledgements

We thank Bernd R. Noack for the fruitful discussions and supply of simulation data which was kindly provided by Gerd Mutschke.

All visualizations in this paper have been created using AMIRA – a system for advanced 3D visualization and volume modeling [16] (see `http://amira.zib.de/`).

References

[1] L. Abraham and K. Shaw. *Dynamics, The Geometry of Behaviour*. Addison-Wesley, 1992.

[2] P. G. Bakker. *Bifurcations in Flow Patterns (Theory and Applications of Transport in Porous Media)*. Kluwer Academic Publishers, 1991.

[3] W. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In *Proc. IEEE Visualization '99*, pages 149–354, 1999.

[4] W. de Leeuw and R. van Liere. Visualization of global flow structures using multiple levels of topology. In *Data Visualization 1999. Proc. VisSym 99*, pages 45–52, 1999.

[5] W. de Leeuw and J. van Wijk. A probe for local flow field visualization. In *Proc. IEEE Visualization '93*, pages 39–45, Los Alamitos, 1993. IEEE Computer Society Press.

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

[7] G. Haller. Finding finite-time invariant manifolds in two-dimensional velocity fields. *Chaos*, 10(1):99–108, 2000.

[8] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D*, 149:248–277, 2001.

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

[10] J. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11:36–46, May 1991.

[11] S.K. Lodha, J.C. Renteria, and K.M. Roskin. Topology preserving compression of 2D vector fields. In *Proc. IEEE Visualization 2000*, pages 343–350, 2000.

[12] H. Löffelmann, H. Doleisch, and E. Gröller. Visualizing dynamical systems near critical points. In *Spring Conference on Computer Graphics and its Applications*, pages 175–184, Budmerice, Slovakia, 1998.

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

[14] R.W. Panton. *Incompressible Flow*. John Wiley & Sons, New York, etc., 1984.

[15] G. Scheuermann, H. Krüger, M. Menzel, and A. Rockwood. Visualizing non-linear vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109–116, 1998.

[16] D. Stalling, H.-C. Hege, and M. Westerhoff. Amira – a highly interactive system for visual data analysis. In Christopher R. Johnson and Charles D. Hansen, editors, *Visualization Handbook*. Academic Press, 2004.

[17] H. Theisel. Designing 2D vector fields of arbitrary topology. *Computer Graphics Forum (Eurographics 2002)*, 21(3):595–604, 2002.

[18] H. Theisel and H.-P. Seidel. Feature flow fields. In *Data Visualization 2003. Proc. VisSym 03*, pages 141–148, 2003.

[19] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Saddle connectors - an approach to visualizing the topological skeleton of complex 3D vector fields. In *Proc. IEEE Visualization 2003*, pages 225–232, 2003.

[20] X. Tricoche, G. Scheuermann, and H. Hagen. A topology simplification method for 2D vector fields. In *Proc. IEEE Visualization 2000*, pages 359–366, 2000.

[21] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proc. Visualization 01*, pages 159 – 166, 2001.

[22] X. Tricoche, G. Scheuermann, and H.Hagen. Topology-based visualization of time-dependent 2D vector fields. In *Data Visualization 2001. Proc. VisSym 01*, pages 117–126, 2001.

[23] X. Tricoche, T. Wischgoll, G. Scheuermann, and H.Hagen. Topology tracking for the visualization of time-dependent two-dimensional flows. *Computers & Graphics*, 26:249–257, 2002.

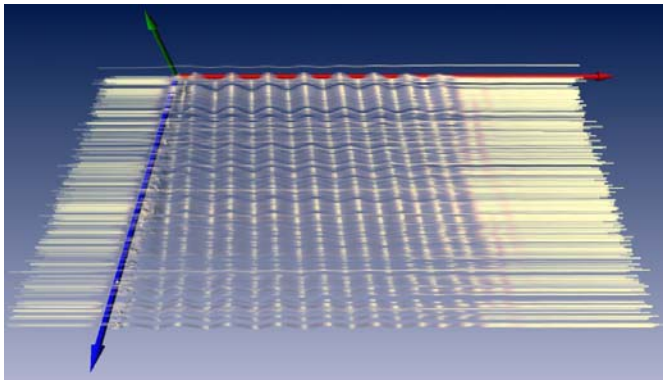
[24] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel. Boundary switch connectors for topological visualization of complex 3d vector fields. In *Data Visualization 2004. Proc. VisSym 04*, 2004.

[25] R. Westermann, C. Johnson, and T. Ertl. Topology-preserving smoothing of vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):222–229, 2001.

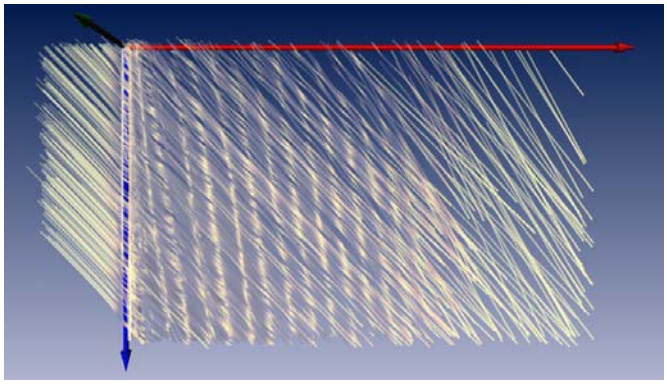
[26] T. Wischgoll and G. Scheuermann. Detection and visualization of closed streamlines in planar flows. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165–172, 2001.

[27] T. Wischgoll, G. Scheuermann, and H. Hagen. Tracking closed stream lines in time-dependent planar flows. In *Proc. Vision, Modeling and Visualization 2001*, pages 447–454, 2001.

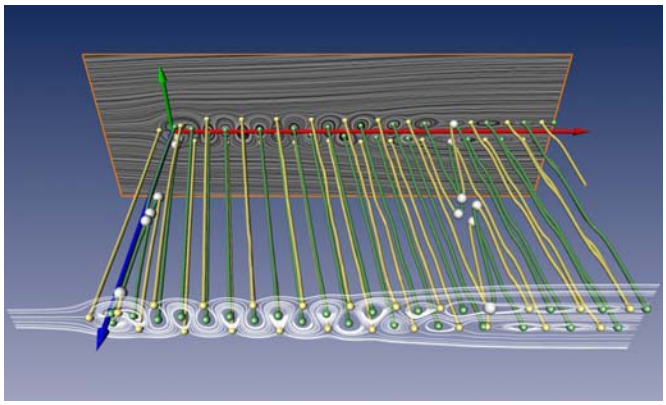
[28] M. Zöckler, D. Stalling, and H.C. Hege. Interactive visualization of 3D-vector fields using illuminated stream lines. In *Proc. IEEE Visualization '96*, pages 107–113, 1996.



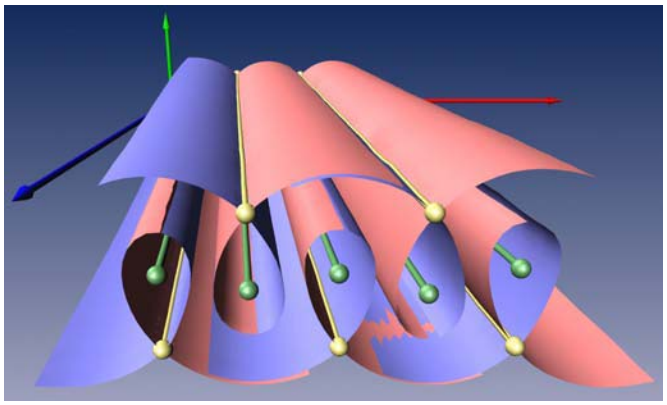
(a) Stream lines of $\mathbf{s}$ correspond to the stream lines in $\mathbf{v}$.



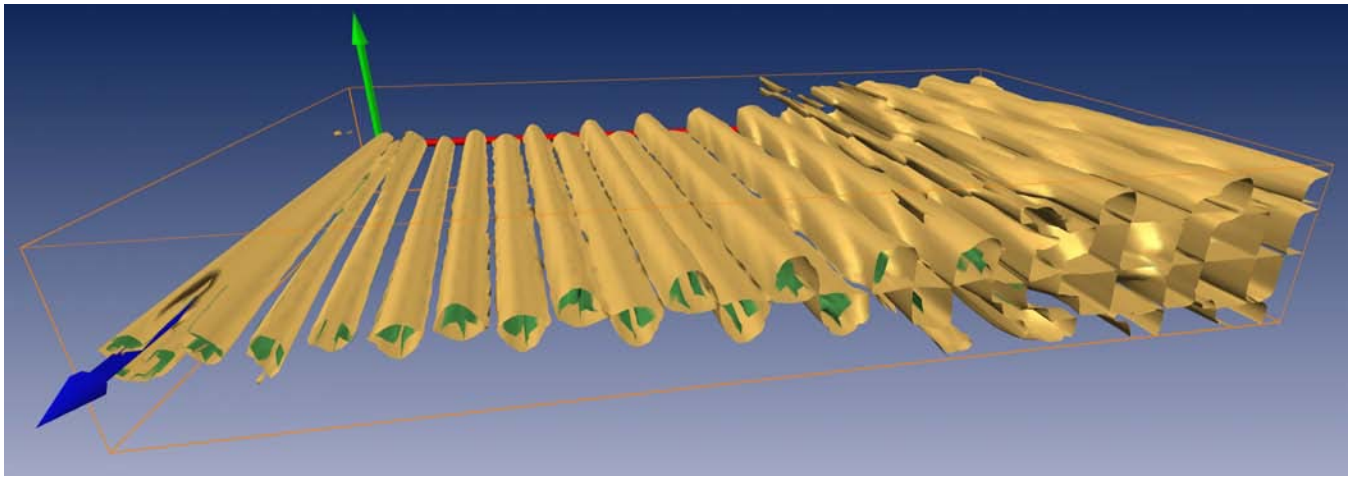
(b) Stream lines of $\mathbf{p}$ correspond to the path lines in $\mathbf{v}$.



(c) Stream line oriented topology after subtracting a constant vector field.



(d) Stream line oriented topology with separation surfaces (close-up).



(e) Path line oriented topology.

Figure 14: 2D Flow behind a circular cylinder.