Extraction of Parallel Vector Surfaces in 3D Time-Dependent Fields and Application to Vortex Core Line Tracking

Holger Theisel [*] MPI Saarbrücken
Jan Sahner [†] ZIB Berlin
Tino Weinkauf [‡] ZIB Berlin
Hans-Christian Hege [§] ZIB Berlin
Hans-Peter Seidel [¶] MPI Saarbrücken

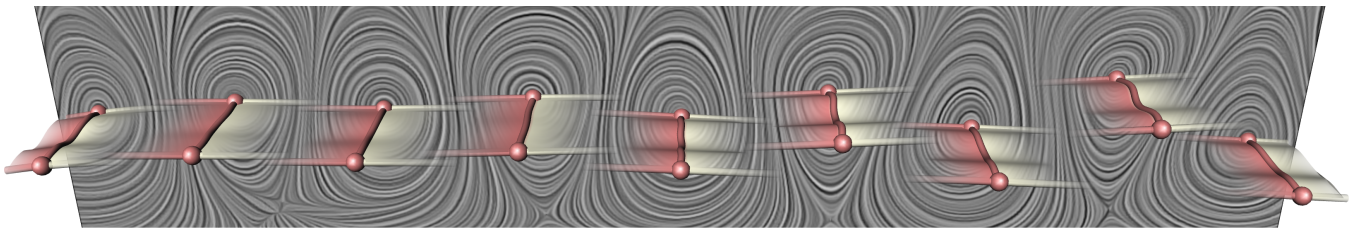


Figure 1: Flow behind a circular cylinder. Shown are vortex core lines in a certain frame of reference. Their evolution over time is tracked by our algorithm and depicted using transparent surfaces. Red color encodes the past while gray shows the future.

ABSTRACT

We introduce an approach to tracking vortex core lines in time-dependent 3D flow fields which are defined by the parallel vectors approach. They build surface structures in the 4D space-time domain. To extract them, we introduce two 4D vector fields which act as feature flow fields, i.e., their integration gives the vortex core structures. As part of this approach, we extract and classify local bifurcations of vortex core lines in space-time. Based on a 4D stream surface integration, we provide an algorithm to extract the complete vortex core structure. We apply our technique to a number of test 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, vortex core lines, bifurcations

1 INTRODUCTION

Flow fields play a vital role in many research areas. Examples are burning chambers, turbomachinery and aircraft design in industry as well as blood flow in medicine. As the resolution of numerical simulations as well as experimental measurements like PIV have evolved significantly in the last years, the challenge of understanding the intricate flow structures within their massive result data sets has made automatic feature extraction schemes necessary.

Among the features of interest, vortices are the most prominent. They play a major role in many research areas due to their wanted or unwanted effects on the flow. In turbomachinery design, vortices reduce efficiency, whereas in burning chambers, vortices have to be controlled to achieve optimal mixing of oxygen and fuel. In aircraft design, vortices can both increase and decrease lift.

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

Thorough overviews of algorithms for the treatment of vortical structures in time-independent flow fields can be found in the literature [13, 12]. We give a short introduction here. Vortex detection schemes can be classified in two major categories:

- *Vortex region detection* is based on scalar *vortex region quantities* that are used to define a vortex as a spatial region where the quantity exhibits a certain value range. Examples are regions of high magnitude of vorticity or negative λ_2-criterion [8]. Isosurfaces or volume rendering are common approaches for visualizing these quantities.

- *Vortex core line extraction* aims at finding line type features that are regarded as centers of vortices. Different approaches exist. Sujudi and Haimes consider lines where the flow exhibits a swirling motion around it [19, 12]. Banks and Singer extract vorticity lines seeded at critical points and corrected towards pressure minima [2]. Roth and Peikert consider stream lines of zero torsion [14]. While those methods depend on the reference frame, recently a vortex core line extraction method was proposed that is Galilean invariant by extracting extremum lines of scalar region quantities [15], i.e., this method is invariant under adding constant flow fields.

All vortex core line methods mentioned above can be implemented using the parallel vectors (PV) operator, a popular line feature extraction approach for static flow fields [12]. The idea of the PV approach is to derive two vector fields $\mathbf{w}_1$, $\mathbf{w}_2$ out of a given 3D vector field $\mathbf{v}$, such that the desired vortex core lines are the locations where $\mathbf{w}_1$ and $\mathbf{w}_2$ are parallel. Several ways of extracting these lines exist [12], either based on extracting and intersecting isosurfaces [9], Newtonian iterations on grid faces, analytic solutions for triangular faces, or curve following schemes [1]. All these solutions (except [1]) have in common that they are based on an underlying grid and a simple (linear or trilinear) interpolation inside the grid cells. In fact, these solutions find intersection points of the vortex core lines on grid faces and connect them afterward by local connection strategies. Also [1] computes the intersection of the vortex core lines with certain faces which are locally set by a predictor step. Then the actual intersections with the vortex core lines are computed in a corrector step.

Flow dynamicists are interested in tracking vortex core lines over time for several reasons:

- When the impact of a feature on the flow is measured by certain criteria (rotation strength [16], pressure [10], Okubo-Weiss-criterion [7]), feature tracking becomes necessary to answer the question, whether the impact of a feature on the flow increases or decreases, when time evolves, as the correspondence problem of features in different time steps is not a trivial task. With the full feature surface of our method at hand, the problem can be solved by simply checking, if two features at different times lie in the same connected component of the surface.

- Also, the spatial evolution over time is interesting e.g. in burning chambers, where the location and extent of vortices are the key ingredient for a complete burning process.

A first approach to tracking PV-based vortex core lines over time was given in [3] which focused on scale-space as the additional dimension. There, a marching-cubes-like algorithm is performed to extract 4D triangular structures in regular 4D hypercubes building the cells of the space-time domain.

In this paper we introduce a new method to extract and track vortex core lines which are based on a PV formulation. This method is based on the concept of feature flow fields (FFF) [21]: we derive appropriate vector fields such that the searched vortex core lines are stream lines on them. This way, the extraction/tracking of vortex core lines is reduced to a simple stream line/surface integration of vector fields. We choose this approach because of the following reasons:

- Numerical stream line/surface integration is well-understood in the Visualization community. A variety of fast and stable algorithms exist for this purpose. [6, 24, 4, 17]

- The stream surface integration approach is independent of an underlying grid, giving a subcell accuracy and relieving us of finding appropriate local connection strategies.

- Bifurcations (i.e. events of sudden changes of the behavior of vortex core lines over time) play an important role in the understanding of the dynamical behavior of vortex core lines. Contrary to pre-existing methods, the FFF approach permits to localize, characterize and classify these bifurcations. To the best of our knowledge, this has not been done in the Visualization community before.

The rest of the paper is organized as follows: sections 2–5 describe our FFF based approach of extracting and tracking vortex core lines. Since the approach is exclusively based on the PV formulation, we describe the approach independently of the vortex core background. We call the solutions of the PV operator in the static case *PV lines*, while their sweeping over time in time-dependent fields are called *PV surfaces*. Section 2 explains the FFF approach to extract PV lines in static fields. Based on this, section 3 introduces the feature flow fields to track PV lines over time. Section 4 gives a complete classification of local bifurcations of PV lines over time. Section 5 describes the final algorithms to tracking PV lines. Section 6 shows the application of our technique to a number of test data sets, among them the particular PV realization for vortex core lines defined by Sujudi and Haimes [19] in a Galerkin model of a flow behind a circular cylinder.

Notation: In this paper we consider points, vectors and vector fields both in 3D and 4D. To make a clear distinction between them, we write a 4D structure as $\widetilde{\mathbf{p}}$, $\widetilde{\mathbf{v}}$, ..., while for 3D structures we simply write $\mathbf{p}$, $\mathbf{v}$.

2 Extracting PV lines in static fields

In this section we shortly describe the parallel vectors (PV) operator [12] and explain how to use the concept of feature flow fields (FFF) in order to extract PV lines in a 3D static field. This will later be the foundation of an FFF-based algorithm for tracking PV lines in unsteady fields.

Given two continuous 3D vector fields $\mathbf{w}_1$ and $\mathbf{w}_2$, the PV operator extracts all points in the domain where the vectors of $\mathbf{w}_1$ and $\mathbf{w}_2$ point in the same direction, i.e. $\mathbf{w}_2 = \lambda \mathbf{w}_1$ for some real λ[1], or $\mathbf{w}_1 \parallel \mathbf{w}_2$ in shorthand notation.

Peikert and Roth [12] list examples of reasonable choices of $\mathbf{w}_1$ and $\mathbf{w}_2$ to extract line type features like vortex core lines in flow fields or extremum lines in scalar fields. As the theory presented here can be formulated completely in terms of the PV operator, we keep the derivation as general as possible and just choose particular vector fields $\mathbf{w}_1$ and $\mathbf{w}_2$ in section 6.

Aiming at extracting PV lines of ($\mathbf{w}_1$, $\mathbf{w}_2$) in the domain $D = [x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$, we define the vector field $\mathbf{s}$ as

$$\mathbf{s}(x,y,z) = \begin{pmatrix} k(x,y,z) \\ m(x,y,z) \\ n(x,y,z) \end{pmatrix} = \mathbf{w}_1 \times \mathbf{w}_2. \tag{1}$$

Then the PV lines consist of all locations (x,y,z) with $\mathbf{s}(x,y,z) = (0,0,0)^T$. If $\mathbf{w}_1$ and $\mathbf{w}_2$ are continuous, then the PV lines are indeed continuous line structures, i.e. point sets of dimensionality 1 [12][2]. PV structures of dimensionality 0 or dimensionality 2 are structurally unstable in 3D, i.e. they disappear by adding noise to the data. For this reason we do not consider them here.

The FFF approach [21] (originally introduced to track critical points of time-dependent vector fields) was already used for extracting particular Galilean invariant vortex core lines without using the PV approach [15]. To apply the FFF concept to PV lines, the following steps are necessary:

1. A vector field $\mathbf{f}$ is defined which fulfills the FFF property. This means that given a point $\mathbf{x}_0 = (x_0, y_0, z_0)$ with $\mathbf{s}(\mathbf{x}_0) = (0,0,0)^T$, each point $\mathbf{x}$ on the stream line of $\mathbf{f}$ starting from $\mathbf{x}_0$ fulfills $\mathbf{s}(\mathbf{x}) = (0,0,0)^T$ as well. In other words: PV lines of ($\mathbf{w}_1$, $\mathbf{w}_2$) are stream lines of $\mathbf{f}$.

2. A set of starting points is defined which guarantees that the stream line integration of $\mathbf{f}$ starting from them covers all PV lines.

Then all PV lines of ($\mathbf{w}_1$, $\mathbf{w}_2$) can simply be extracted by applying a stream line integration of $\mathbf{f}$. We treat the two parts of the approach in the following subsections.

2.1 Obtaining the feature flow field f

In this section we show that $\mathbf{f}$ essentially consists of an appropriate combination of the first order partials $\mathbf{s}_x$, $\mathbf{s}_y$, $\mathbf{s}_z$ of $\mathbf{s}$. We denote the gradients of the components of $\mathbf{s}$ in (1) by

$$\nabla k = \begin{pmatrix} k_x \\ k_y \\ k_z \end{pmatrix}, \quad \nabla m = \begin{pmatrix} m_x \\ m_y \\ m_z \end{pmatrix}, \quad \nabla n = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}.$$

Then $\mathbf{f}$ should point into a direction where the components of $\mathbf{s}$ remain constant. This means that $\mathbf{f}$ has to be perpendicular to the

[1] $\lambda = \pm\infty$ is also allowed, i.e., $\mathbf{w}_1 \parallel \mathbf{w}_2$ holds if $\mathbf{w}_1$ or $\mathbf{w}_2$ vanishes.

[2] Note that this statement gives that these line structures cannot be obtained by replacing $\mathbf{s}(x,y,z) = (0,0,0)^T$ by $\|\mathbf{s}(x,y,z)\| = 0$ and applying a simple scalar field analysis of $\|\mathbf{s}\|$, since the zeros of general scalar fields are structures of dimensionality 2.

gradients of the components of $\mathbf{s}$. We define

$$\mathbf{f}_1 = \nabla m \times \nabla n = \begin{pmatrix} \det(\mathbf{s}_y, \mathbf{s}_z, (1,0,0)^T) \\ \det(\mathbf{s}_z, \mathbf{s}_x, (1,0,0)^T) \\ \det(\mathbf{s}_x, \mathbf{s}_y, (1,0,0)^T) \end{pmatrix}. \quad (2)$$

which is perpendicular to ∇m and ∇n. Hence, all points on a stream line of $\mathbf{f}_1$ have constant components m and n. In a similar way we define $\mathbf{f}_2$ and $\mathbf{f}_3$ as

$$\mathbf{f}_2 = \nabla n \times \nabla k = \begin{pmatrix} \det(\mathbf{s}_y, \mathbf{s}_z, (0,1,0)^T) \\ \det(\mathbf{s}_z, \mathbf{s}_x, (0,1,0)^T) \\ \det(\mathbf{s}_x, \mathbf{s}_y, (0,1,0)^T) \end{pmatrix} \quad (3)$$

$$\mathbf{f}_3 = \nabla k \times \nabla m = \begin{pmatrix} \det(\mathbf{s}_y, \mathbf{s}_z, (0,0,1)^T) \\ \det(\mathbf{s}_z, \mathbf{s}_x, (0,0,1)^T) \\ \det(\mathbf{s}_x, \mathbf{s}_y, (0,0,1)^T) \end{pmatrix}. \quad (4)$$

In general, $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$ describe different directions. However, if we are on a PV line, we can write $\mathbf{w}_2 = \lambda \mathbf{w}_1$ for a certain λ. Inserting this into the computation of the derivatives of (1), we get

$$\begin{aligned} \mathbf{s}_x &= (\mathbf{w}_{1x} \times \mathbf{w}_2) + (\mathbf{w}_1 \times \mathbf{w}_{2x}) = \mathbf{w}_1 \times (\mathbf{w}_{2x} - \lambda \mathbf{w}_{1x}) \\ \mathbf{s}_y &= (\mathbf{w}_{1y} \times \mathbf{w}_2) + (\mathbf{w}_1 \times \mathbf{w}_{2y}) = \mathbf{w}_1 \times (\mathbf{w}_{2y} - \lambda \mathbf{w}_{1y}) \quad (5) \\ \mathbf{s}_z &= (\mathbf{w}_{1z} \times \mathbf{w}_2) + (\mathbf{w}_1 \times \mathbf{w}_{2z}) = \mathbf{w}_1 \times (\mathbf{w}_{2z} - \lambda \mathbf{w}_{1z}) \end{aligned}$$

which shows that $\mathbf{s}_x$, $\mathbf{s}_y$, $\mathbf{s}_z$ are coplanar perpendicular to $\mathbf{w}_1$ and $\mathbf{w}_2$. (5) and (2)–(4) give that $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$ are parallel on a PV line. Thus, almost every linear combination of $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$ can act as feature flow field. Setting $\mathbf{f} = \alpha\,\mathbf{f}_1 + \beta\,\mathbf{f}_2 + \gamma\,\mathbf{f}_3$ and $\mathbf{a} = (\alpha, \beta, \gamma)^T$, we obtain

$$\mathbf{f} = \begin{pmatrix} e \\ f \\ g \end{pmatrix} = \begin{pmatrix} \det(\mathbf{s}_y, \mathbf{s}_z, \mathbf{a}) \\ \det(\mathbf{s}_z, \mathbf{s}_x, \mathbf{a}) \\ \det(\mathbf{s}_x, \mathbf{s}_y, \mathbf{a}) \end{pmatrix}. \quad (6)$$

Choosing the vector field a:
In order to choose a suitable vector field $\mathbf{a}$, we rewrite $\mathbf{f}$ as

$$\mathbf{f} = \begin{pmatrix} \mathbf{b}_1 \cdot \mathbf{a} \\ \mathbf{b}_2 \cdot \mathbf{a} \\ \mathbf{b}_3 \cdot \mathbf{a} \end{pmatrix} \quad (7)$$

with

$$\mathbf{b}_1 = \mathbf{s}_y \times \mathbf{s}_z \;,\; \mathbf{b}_2 = \mathbf{s}_z \times \mathbf{s}_x \;,\; \mathbf{b}_3 = \mathbf{s}_x \times \mathbf{s}_y. \quad (8)$$

On a PV line, the coplanarity of $\mathbf{s}_x$, $\mathbf{s}_y$, $\mathbf{s}_z$ together with (8) gives $\mathbf{w}_1 \parallel \mathbf{w}_2 \parallel \mathbf{b}_1 \parallel \mathbf{b}_2 \parallel \mathbf{b}_3$. Consequently, the only condition we have to put on $\mathbf{a}$ is that it must not be perpendicular to $\mathbf{w}_1$ and $\mathbf{w}_2$ respectively on a PV line. If we know that $\mathbf{w}_1$ never vanishes on a PV line, the simple choice $\mathbf{a} = \mathbf{w}_1$ does the job. A similar statement holds for $\mathbf{w}_2$. In case that both $\mathbf{w}_1$ and $\mathbf{w}_2$ may vanish on a PV line, we choose

$$\mathbf{a} = \begin{cases} \mathbf{w}_1 & \text{if } \|\mathbf{w}_1\| \geq \|\mathbf{w}_2\| \\ \mathbf{w}_2 & \text{otherwise,} \end{cases}$$

which guarantees $\mathbf{a}$ to be continuous in direction but not in orientation. Thus, $\mathbf{f}$ has to be integrated as an orientation-free vector field (similar e.g. to an eigenvector field of a tensor field), i.e., the local orientation has to be obtained from the information where the integration of the line has come from.

A number of vortex core line extraction concepts based on parallel vectors use $\mathbf{w}_1 = \mathbf{v}$ and $\mathbf{w}_2 = \mathbf{M}\,\mathbf{v}$ (see for example [19] with $\mathbf{M} = \nabla \mathbf{v}$ or [20] with $\mathbf{M} = (\nabla \mathbf{v})^T$). For these approaches, both $\mathbf{w}_1$ and $\mathbf{w}_2$ vanish at critical points of $\mathbf{v}$ causing both $\mathbf{s}(\mathbf{x}) = (0,0,0)^T$ and $\mathbf{f}(\mathbf{x}) = (0,0,0)^T$ there. To deal with this problem, we equivalently reformulate $\mathbf{w}_2$ as an appropriate eigenvector of $\mathbf{M}$. This eigenvector does not vanish along the vortex core line, but since eigenvectors have no orientation, an orientation-free integration of $\mathbf{f}$ is necessary here as well.

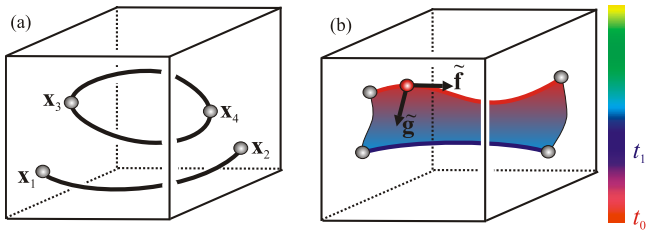


Figure 2: (a) two kinds of PV lines: ending in the boundary points $\mathbf{x}_1$, $\mathbf{x}_2$, or closed: in this case we extract two starting points $\mathbf{x}_3$,$\mathbf{x}_4$; (b) PV structures are surfaces in 4D.

2.2 Starting points for integrating f

Given the definitions above, we first analyze whether an integration of $\mathbf{f}$ along a PV line may get stuck in a critical point of $\mathbf{f}$. This happens at a location $\mathbf{x} \in D$ with $\mathbf{s}(\mathbf{x}) = (0,0,0)^T$ and $\mathbf{f}(\mathbf{x}) = (0,0,0)^T$. Both conditions independently build structures of dimensionality 1 in D, i.e., they are line structures[3]. If these line structures intersect, the intersection points are structurally unstable, i.e., they disappear by adding noise to the data. Because of this we assume that such intersections do not exist in D. However, we mention that $[\mathbf{s}(\mathbf{x}) = (0,0,0)^T\,,\,\mathbf{f}(\mathbf{x}) = (0,0,0)^T]$ gives stable solutions in a time-depending setting. We treat this in section 4.

If $[\mathbf{s}(\mathbf{x}) = (0,0,0)^T\,,\,\mathbf{f}(\mathbf{x}) = (0,0,0)^T]$ does not have solutions in D, every PV line ends either on the boundary faces of D (for both forward and backward integration), or builds a closed stream line in $\mathbf{f}$ [12][4]. To get the starting points for the first case, we search for the intersections of the PV lines with the boundary of D: for the boundary face $x = x_{min}$, we search for all points (y,z) with $\mathbf{s}(x_{min}, y, z) = (0,0,0)^T$. To do so, different numerical solvers can be applied. We use a simple subdivision approach in the (y,z)-domain: a rectangular cell C is checked whether one of the components k,m,n is positive/negative at *all* 4 corners of C. If so, no PV line intersection is found inside C. Otherwise, we recursively subdivide C into 4 subcells until their size is smaller than a certain threshold. In a similar way we compute the intersections of the PV lines with the remaining 5 faces.

To find a starting point on a closed PV line, it is sufficient to identify an arbitrary point on the line. We have chosen to extract points $\mathbf{x}$ with

$$[\;\mathbf{s}(\mathbf{x}) = (0,0,0)^T \quad,\quad e(\mathbf{x}) = 0\;] \quad (9)$$

with e from (6). To do so, we apply a similar 3D subdivision approach as described above for the 2D case. Since a closed PV line must consist of points with both positive and negative e-components, each closed PV line must consist of at least two points fulfilling (9). Figure 2a illustrates an example. Clearly, (9) may also deliver solutions on open PV lines, but it guarantees to find at least two solutions for each closed PV line[5]. Finally, we do an integration of $\mathbf{f}$ starting from all detected points and remove multiply obtained curves.

[3] This has been shown for $\mathbf{s}$, as it defines PV lines [12]. To show that $\mathbf{f}(\mathbf{x}) = (0,0,0)^T$ builds line structures as well, we have to show that $\mathbf{f}$ can also be formulated using the PV operator. (2)–(4) give that we can rewrite $\mathbf{f}$ as $\mathbf{f} = (\alpha\,\nabla m - \beta\,\nabla k) \times (\nabla n - \frac{\gamma}{\beta}\,\nabla m)$. Hence, $\mathbf{f} = (0,0,0)^T$ corresponds to $(\alpha\,\nabla m - \beta\,\nabla k) \parallel (\nabla n - \frac{\gamma}{\beta}\,\nabla m)$.

[4] This statement implies that our approach does not have to incorporate algorithms to detecting closed stream lines in flow fields [26], since we know in advance that our stream lines of interest in $\mathbf{f}$ are closed.

[5] If a closed PV line completely lies in the $y-z$ plane by chance, (9) gives many solutions. In this case, $e(\mathbf{x}) = 0$ can simply be replaced by $f(\mathbf{x}) = 0$ in (9) to reduce the number of solutions.

3 Feature flow fields for tracking

Now we consider PV structures in time-dependent 3D vector fields $(\mathbf{w}_1(x,y,z,t),\mathbf{w}_2(x,y,z,t))$. To do so, we first note that all the 3D static vector fields $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{s}$, ∇k, ∇m, ∇n, $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$, $\mathbf{f}$, $\mathbf{a}$ which were introduced in section 2 can be defined in a similar way for the time-dependent case as well. In the following we consider these vector fields to be time-dependent, i.e. they are maps for the 4D domain $\widetilde{D} = D \times [t_{min}, t_{max}]$ to $\mathbb{R}^3$.

PV structures in $(\mathbf{w}_1(x,y,z,t),\mathbf{w}_2(x,y,z,t))$ can be considered as lines in D sweeping over time while smoothly changing their shape and location. In addition, certain bifurcations may occur. Hence, the PV structures in $\widetilde{D}$ have the dimensionality 2, i.e. they are surfaces in $\widetilde{D}$. Figure 2b gives an illustration. Here a PV line at time t_0 (red) moves to the blue line at time t_1. Each point on the swept surface between the two lines is actually a 4D point: in addition to the spatial values it is provided with a t-value. In figure 2b (as well as in the following figures) we color code the t-values of points, lines and surfaces.

In order to extract the PV surfaces in $\widetilde{D}$, we need to define two 4D feature flow fields $\widetilde{\mathbf{f}}$ and $\widetilde{\mathbf{g}}$. The first one can easily be defined as

$$\widetilde{\mathbf{f}}(x,y,z,t) = \begin{pmatrix} \mathbf{f}(x,y,z,t) \\ 0 \end{pmatrix} \tag{10}$$

where $\mathbf{f}$ is defined in (6). It gives a PV line at a certain time level, i.e. all points on a stream line of $\widetilde{\mathbf{f}}$ have the same t-value. The evolution in time of a PV line should be covered by the 4D feature flow field $\widetilde{\mathbf{g}}$. Keeping mind that PV structures in $\widetilde{D}$ are surfaces, a family of different $\widetilde{\mathbf{g}}$ could be chosen such that each linear combination of $\widetilde{\mathbf{f}}$ and $\widetilde{\mathbf{g}}$ is a FFF. Among them, we choose the $\widetilde{\mathbf{g}}$ with $\widetilde{\mathbf{f}} \perp \widetilde{\mathbf{g}}$. This gives a unique $\widetilde{\mathbf{g}}$ (except for scaling). We obtain

$$\widetilde{\mathbf{g}}(x,y,z,t) = \begin{pmatrix} \mathbf{h} \times \mathbf{f} \\ \|\mathbf{f}\|^2 \end{pmatrix} = \begin{pmatrix} \mathbf{h} \times \mathbf{f} \\ e^2 + f^2 + g^2 \end{pmatrix} \tag{11}$$

with

$$\mathbf{h}(x,y,z,t) = \begin{pmatrix} \det(\mathbf{s}_x, \mathbf{s}_t, \mathbf{a}) \\ \det(\mathbf{s}_y, \mathbf{s}_t, \mathbf{a}) \\ \det(\mathbf{s}_z, \mathbf{s}_t, \mathbf{a}) \end{pmatrix} \tag{12}$$

and $\mathbf{f}$ defined in (6). Figure 2b illustrates $\widetilde{\mathbf{f}}$ and $\widetilde{\mathbf{g}}$ at a certain point (red) on the PV surface.

To prove that $\widetilde{\mathbf{g}}$ is indeed the desired feature flow field, we consider the gradients in $\widetilde{D}$ of the components of $\mathbf{s}$:

$$\widetilde{\nabla k} = \begin{pmatrix} k_x \\ k_y \\ k_z \\ k_t \end{pmatrix}, \quad \widetilde{\nabla m} = \begin{pmatrix} m_x \\ m_y \\ m_z \\ m_t \end{pmatrix}, \quad \widetilde{\nabla n} = \begin{pmatrix} n_x \\ n_y \\ n_z \\ n_t \end{pmatrix}.$$

Then we have to show that from $\mathbf{w}_1 \parallel \mathbf{w}_2$ (i.e. for $\mathbf{s} = (0,0,0)^T$) the following four properties can be deduced:

$$\widetilde{\nabla k} \cdot \widetilde{\mathbf{g}} \;=\; \widetilde{\nabla m} \cdot \widetilde{\mathbf{g}} \;=\; \widetilde{\nabla n} \cdot \widetilde{\mathbf{g}} \;=\; \widetilde{\mathbf{f}} \cdot \widetilde{\mathbf{g}} \;=\; 0$$

where $\cdot$ denotes the 4D dot product. This can be shown by a straightforward exercise in algebra.

We note that Theisel et al. already proposed a FFF for PV tracking [21] which appears not to work: the FFF proposed there is constantly vanishing on a PV line and therefore unable to track it.

4 Local bifurcations

Although in general PV lines change smoothly over time, there are certain points $\widetilde{D}$ in which the behavior of the PV lines changes abruptly. These bifurcation are a vital ingredient for assuring the complete extraction of PV surfaces. Furthermore, knowing what kind of bifurcations may occur contributes to understanding the parallel vectors operator. In this section we characterize local bifurcations and show how to extract them. In general, we can distinguish between two kinds of bifurcations: inner bifurcations and boundary bifurcations. We treat them separately in the following sections.

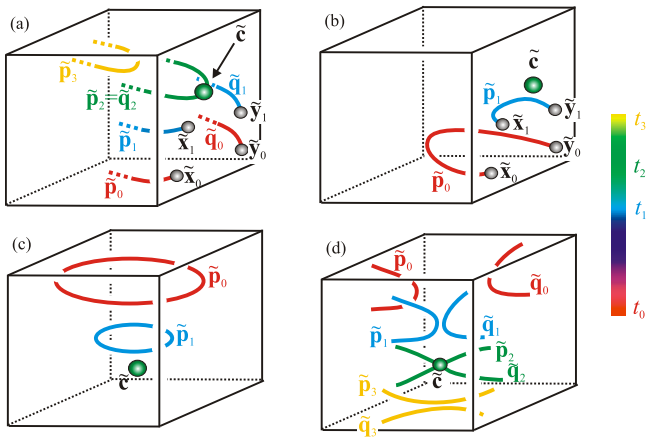
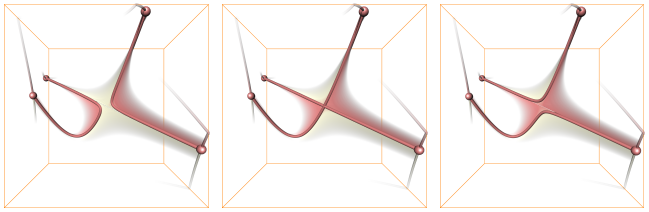


Figure 3: local bifurcations of PV lines: (a) inflow boundary bifurcation; (b) outflow boundary bifurcation; (c) closed collapse bifurcation; (d) saddle bifurcation.



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

Figure 4: Saddle bifurcation.

4.1 Inner bifurcations

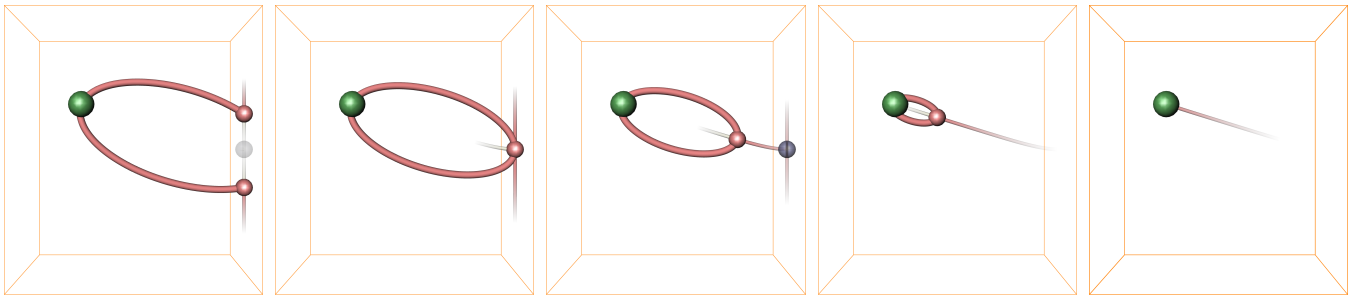
An inner bifurcation is characterized by the fact that the integration of $\widetilde{\mathbf{f}}$ on a PV surface in $\widetilde{D}$ ends in a critical point of $\widetilde{\mathbf{f}}$. This means that an inner bifurcation occurs at a location $\widetilde{\mathbf{c}} \in \widetilde{D}$ with

$$[\ \mathbf{s}(\widetilde{\mathbf{c}}) = (0,0,0)^T \quad , \quad \mathbf{f}(\widetilde{\mathbf{c}}) = (0,0,0)^T\ ]. \tag{13}$$

Since both $\mathbf{s}(\widetilde{\mathbf{c}}) = (0,0,0)^T$ and $\mathbf{f}(\widetilde{\mathbf{c}}) = (0,0,0)^T$ gives surfaces in $\widetilde{D}$ as solutions[6], their intersections are stable isolated points in $\widetilde{D}$. To get them, we use a subdivision approach in 4D similar to the one already explained in 2D and 3D.

In order to analyze the behavior of the PV lines around an inner bifurcation, we analyze the Jacobian matrix of $\widetilde{\mathbf{f}}$ in $\widetilde{\mathbf{c}}$ which is a common approach in the field of vector field topology to classify critical points of 2D [5] and 3D [25] vector fields. On $\widetilde{\mathbf{c}}$ fulfilling (13) we know from (7) that $\mathbf{b}_1 = \mathbf{b}_2 = \mathbf{b}_3 = (0,0,0)^T$. This and (8) gives that $\mathbf{s}_x \parallel \mathbf{s}_y \parallel \mathbf{s}_z$, i.e. we can set $\mathbf{s}_y = p\,\mathbf{s}_x$ and $\mathbf{s}_z = q\,\mathbf{s}_x$ for

[6] This is due to the fact that both $\mathbf{s}(\widetilde{\mathbf{c}}) = (0,0,0)^T$ and $\mathbf{f}(\widetilde{\mathbf{c}}) = (0,0,0)^T$ can be interpreted as sweeping lines over time.



(a) Shortly before inflow bifurcation. (b) The event of inflow bifurcation. (c) Shortly after inflow bifurcation. (d) Shortly before closed collapse bifurcation. (e) The event of closed collapse bifurcation.

Figure 5: Inflow bifurcation and closed collapse bifurcation.

certain p, q. Inserting this into the derivatives of (8), we get

$$\begin{aligned}
\mathbf{b}_{1x} &= (\mathbf{s}_{yx}\times\mathbf{s}_z)+(\mathbf{s}_y\times\mathbf{s}_{zx}) &&= \mathbf{s}_x\times(p\,\mathbf{s}_{zx}-q\,\mathbf{s}_{yx})\\
\mathbf{b}_{1y} &= (\mathbf{s}_{yy}\times\mathbf{s}_z)+(\mathbf{s}_y\times\mathbf{s}_{zy}) &&= \mathbf{s}_x\times(p\,\mathbf{s}_{zy}-q\,\mathbf{s}_{yy})\\
\mathbf{b}_{1z} &= (\mathbf{s}_{yz}\times\mathbf{s}_z)+(\mathbf{s}_y\times\mathbf{s}_{zz}) &&= \mathbf{s}_x\times(p\,\mathbf{s}_{zz}-q\,\mathbf{s}_{yz})\\
\mathbf{b}_{2x} &= (\mathbf{s}_{zx}\times\mathbf{s}_x)+(\mathbf{s}_z\times\mathbf{s}_{xx}) &&= \mathbf{s}_x\times(q\,\mathbf{s}_{xx}-\mathbf{s}_{zx})\\
\mathbf{b}_{2y} &= (\mathbf{s}_{zy}\times\mathbf{s}_x)+(\mathbf{s}_z\times\mathbf{s}_{xy}) &&= \mathbf{s}_x\times(q\,\mathbf{s}_{xy}-\mathbf{s}_{zy})\\
\mathbf{b}_{2z} &= (\mathbf{s}_{zz}\times\mathbf{s}_x)+(\mathbf{s}_z\times\mathbf{s}_{xz}) &&= \mathbf{s}_x\times(q\,\mathbf{s}_{xz}-\mathbf{s}_{zz})\\
\mathbf{b}_{3x} &= (\mathbf{s}_{xx}\times\mathbf{s}_y)+(\mathbf{s}_x\times\mathbf{s}_{yx}) &&= \mathbf{s}_x\times(\mathbf{s}_{yx}-p\,\mathbf{s}_{xx})\\
\mathbf{b}_{3y} &= (\mathbf{s}_{xy}\times\mathbf{s}_y)+(\mathbf{s}_x\times\mathbf{s}_{yy}) &&= \mathbf{s}_x\times(\mathbf{s}_{yy}-p\,\mathbf{s}_{xy})\\
\mathbf{b}_{3z} &= (\mathbf{s}_{xz}\times\mathbf{s}_y)+(\mathbf{s}_x\times\mathbf{s}_{yz}) &&= \mathbf{s}_x\times(\mathbf{s}_{yz}-p\,\mathbf{s}_{xz}).
\end{aligned} \tag{14}$$

This shows that the 9 vectors in (14) are coplanar perpendicular to $\mathbf{s}_x$, $\mathbf{s}_y$, $\mathbf{s}_z$. In addition, the following statements follow directly from (14):

$$\mathbf{b}_{1x}+p\,\mathbf{b}_{2x}+q\,\mathbf{b}_{2x} = (0,0,0)^T \tag{15}$$
$$\mathbf{b}_{1y}+p\,\mathbf{b}_{2y}+q\,\mathbf{b}_{2y} = (0,0,0)^T \tag{16}$$
$$\mathbf{b}_{1z}+p\,\mathbf{b}_{2z}+q\,\mathbf{b}_{2z} = (0,0,0)^T \tag{17}$$
$$\mathbf{b}_{1x}+\mathbf{b}_{2y}+\mathbf{b}_{3z} = (0,0,0)^T. \tag{18}$$

Keeping $\mathbf{b}_1=\mathbf{b}_2=\mathbf{b}_3=(0,0,0)^T$ in mind, we can write the Jacobian matrix of $\mathbf{f}(\widetilde{\mathbf{c}})$ as

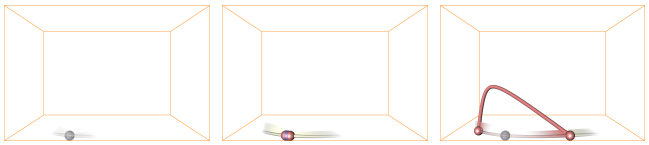
$$\mathbf{J}_{\mathbf{f}(\widetilde{\mathbf{c}})} = \begin{pmatrix} \mathbf{b}_{1x}\cdot\mathbf{a} & \mathbf{b}_{1y}\cdot\mathbf{a} & \mathbf{b}_{1z}\cdot\mathbf{a} \\ \mathbf{b}_{2x}\cdot\mathbf{a} & \mathbf{b}_{2y}\cdot\mathbf{a} & \mathbf{b}_{2z}\cdot\mathbf{a} \\ \mathbf{b}_{3x}\cdot\mathbf{a} & \mathbf{b}_{3y}\cdot\mathbf{a} & \mathbf{b}_{3z}\cdot\mathbf{a} \end{pmatrix}. \tag{19}$$

(15)–(17) show that the lines of $\mathbf{J}_{\mathbf{f}(\widetilde{\mathbf{c}})}$ are not independent, which gives $\det(\mathbf{J}_{\mathbf{f}(\widetilde{\mathbf{c}})})=0$. Hence, one eigenvalue of $\mathbf{J}_{\mathbf{f}(\widetilde{\mathbf{c}})}$ is zero. From (18) we infer that the trace of $\mathbf{J}_{\mathbf{f}(\widetilde{\mathbf{c}})}$ is zero. As the trace of a matrix equals the sum of its eigenvalues, we see that also the remaining eigenvalues of $\mathbf{J}_{\mathbf{f}(\widetilde{\mathbf{c}})}$ add to zero. So they can be written as

$$0\ ,\ -\sqrt{r}\ ,\ \sqrt{r}$$

for some real, possibly negative r. Hence we can classify three kinds of inner bifurcations. The first, $r=0$, is a generally unstable higher order inner bifurcation and not considered here. For $r\neq 0$, exactly two stable kinds of inner bifurcations are possible depending on the sign of r:

A *closed collapse bifurcation* appears if (13) and $r<0$ hold. In this case, the two non-zero eigenvalues of $\mathbf{J}_{\mathbf{f}(\widetilde{\mathbf{c}})}$ are purely imaginary indicating a rotational behavior of $\mathbf{f}$ around $\widetilde{\mathbf{c}}$. While figure 3c illustrates this, figure 5 depicts this bifurcation using a test data set, see section 6 for details on the used visualization scheme. Now imagine a closed PV line $\widetilde{\mathbf{p}_0}$ at the time t_0. While moving forward in time ($t=t_1$), the closed PV line $\widetilde{\mathbf{p}_1}$ becomes smaller until at a certain time t_2 it collapses to a point $\widetilde{\mathbf{c}}$ and disappears. Note that the inverse case of a closed collapse bifurcation exists as well indicating the birth of a (small) closed PV line.



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

Figure 6: Reversed outflow boundary bifurcation.

A *saddle bifurcation* appears if (13) and $r>0$ hold. See figure 3d for an illustration and figure 4 for an example from a test data set. Two PV lines $\widetilde{\mathbf{p}_0}$, $\widetilde{\mathbf{q}_0}$ at the time $t=t_0$ move towards each other ($t=t_1$), share a common point $\widetilde{\mathbf{c}}$ at the time $t=t_2$, and move away from each other ($t=t_3$). The directions of the PV lines out of $\widetilde{\mathbf{c}}$ are the directions of the two eigenvectors of $\mathbf{J}_{\mathbf{f}(\widetilde{\mathbf{c}})}$ corresponding to the non-zero eigenvalues.

4.2 Boundary bifurcations

A boundary bifurcation is characterized by the fact that $\widetilde{\mathbf{f}}$ on a PV surface is tangential to the boundary surface of $\widetilde{D}$. A boundary bifurcation $\widetilde{\mathbf{c}}=(x_{\widetilde{\mathbf{c}}},y_{\widetilde{\mathbf{c}}},z_{\widetilde{\mathbf{c}}},t_{\widetilde{\mathbf{c}}})$ on the boundary face $x=x_{max}$ is the solution of

$$[\ \mathbf{s}(\widetilde{\mathbf{c}})=(0,0,0)^T\ \ ,\ \ x_{\widetilde{\mathbf{c}}}=x_{max}\ \ ,\ \ e(\widetilde{\mathbf{c}})=0\ ] \tag{20}$$

which gives isolated points in the stable case. To get them, we may apply a 3D subdivision approach (in (x,y,t)-space) similar as described in section 2.2. However, there is a faster approach which will be explained later in section 5. In a similar way we compute the boundary bifurcations for the remaining boundary faces of D.

At a boundary bifurcation $\widetilde{\mathbf{c}}$, the integration of $\widetilde{\mathbf{f}}$ starting from $\widetilde{\mathbf{c}}$ (both in forward and backward direction) may enter $\widetilde{D}$, or it may leave $\widetilde{D}$ immediately after starting the integration. To distinguish these two kinds of behavior, we check whether directional derivative $\nabla\mathbf{f}\cdot\mathbf{f}$ of $\mathbf{f}$ points inside or outside D. In the first case, we have an *inflow boundary bifurcation*. See figure 3a for an illustrating example. Imagine two PV lines $\widetilde{\mathbf{p}_0}$ and $\widetilde{\mathbf{q}_0}$ at the time $t=t_0$ which leave $\widetilde{D}$ at the points $\widetilde{\mathbf{x}_0}$ and $\widetilde{\mathbf{y}_0}$ respectively. While moving forward in time ($t=t_1$), the exit points $\widetilde{\mathbf{x}_1}$, $\widetilde{\mathbf{y}_1}$ of the current PV lines $\widetilde{\mathbf{p}_1}$, $\widetilde{\mathbf{q}_1}$ move towards each other until at a certain time t_2 they collapse to

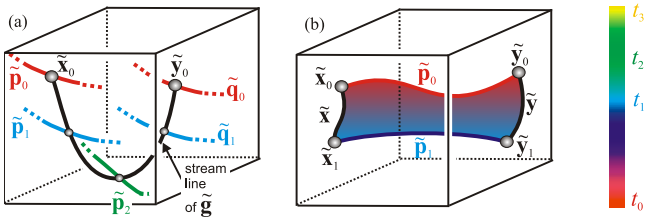


Figure 7: (a) PV line fold bifurcations do not exist! (b) tracking an open PV line.

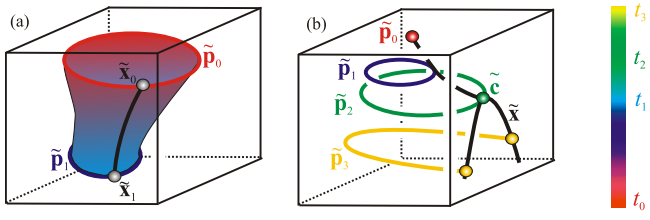


Figure 8: seeding lines (black) for simple examples.

a point $\tilde{\mathbf{c}}$. At $\tilde{\mathbf{c}}$ the current PV lines $\widetilde{\mathbf{p}_2}$, $\widetilde{\mathbf{q}_2}$ get smoothly connected and build a single PV line $\widetilde{\mathbf{p}_3}$ from this moment on. The point $\tilde{\mathbf{c}}$ denotes the inflow boundary collapse bifurcations.

An *outflow boundary bifurcation* illustrated in figure 3b and shown in figure 6 within a test data set. Here, the PV line $\widetilde{\mathbf{p}_0}$ at the time t_0 enters and leaves $\widetilde{D}$ at the points $\widetilde{\mathbf{x}_0}$ and $\widetilde{\mathbf{y}_0}$. While moving forward in time ($t = t_1$), the exit points $\widetilde{\mathbf{x}_1}$, $\widetilde{\mathbf{y}_1}$ move towards each other, until at a certain time t_2 they collapse in the point $\tilde{\mathbf{c}}$ making the PV line disappear.

Also for boundary bifurcations the reverse cases exist. At an inflow boundary bifurcation a PV line may split up into two lines, and at an inflow boundary bifurcation a PV line may appear.

4.3 Further bifurcations

After introducing the local bifurcations above, one may ask whether there are more bifurcations possible. In particular we check whether fold bifurcations of (open or closed) PV lines exist. A fold bifurcation occurs when two PV lines move toward each other, merge at a certain time and immediately disappear after that[7]. It turns out that such a bifurcation cannot exist for PV lines. To show this, imagine two PV lines $\widetilde{\mathbf{p}_0}$, $\widetilde{\mathbf{q}_0}$ at the time t_0 as illustrated in figure 7a. While moving forward in time ($t = t_1$), the current PV lines $\widetilde{\mathbf{p}_1}$, $\widetilde{\mathbf{q}_1}$ move towards each other until they merge in the line $\widetilde{\mathbf{p}_2}$ and disappear. If we pick a point $\widetilde{\mathbf{x}_0}$ on $\widetilde{\mathbf{p}_0}$ and start a stream line integration of $\tilde{\mathbf{g}}$ from $\widetilde{\mathbf{x}_0}$, we end in a point $\widetilde{\mathbf{y}_0}$ on $\widetilde{\mathbf{q}_0}$. Since $\widetilde{\mathbf{x}_0}$ and $\widetilde{\mathbf{y}_0}$ have the same t-value $t = t_0$, the integration of $\tilde{\mathbf{g}}$ must go both forward and backward in time. This is a contradiction to (11) which shows that the last component of $\tilde{\mathbf{g}}$ (specifying the evolution in time) is always non-negative. Therefore, PV fold bifurcations do not exist.

5 The Algorithms

Before we formulate the algorithms for PV surface extraction in $\widetilde{D}$, we explain the main ideas on a number of simple examples.

Consider figure 7b: suppose there is a PV line $\widetilde{\mathbf{p}_0}$ at the time $t = t_0$ which leaves $\widetilde{D}$ in the points $\widetilde{\mathbf{x}_0}$, $\widetilde{\mathbf{y}_0}$. While moving forward in time until $t = t_1$, $\widetilde{\mathbf{p}_0}$ sweeps to the line $\widetilde{\mathbf{p}_1}$ which leaves $\widetilde{D}$ in $\widetilde{\mathbf{x}_1}$, $\widetilde{\mathbf{y}_1}$. Doing this sweeping, the points where the PV lines leave $\widetilde{D}$ form two lines $\widetilde{\mathbf{x}}$, $\widetilde{\mathbf{y}}$ on the boundary of $\widetilde{D}$: $\widetilde{\mathbf{x}}$ connects $\widetilde{\mathbf{x}_0}$ and $\widetilde{\mathbf{x}_1}$, while $\widetilde{\mathbf{y}}$ connects $\widetilde{\mathbf{y}_0}$ and $\widetilde{\mathbf{y}_1}$. In order to extract the PV surfaces (i.e. the surface bounded by the curves $\widetilde{\mathbf{p}_0}$, $\widetilde{\mathbf{y}}$, $\widetilde{\mathbf{p}_1}$, $\widetilde{\mathbf{x}}$), we have the choice between two approaches: One approach is to start with an extraction of $\widetilde{\mathbf{p}_0}$ and using it as seeding curve for a stream surface integration of $\tilde{\mathbf{g}}$ until we reach $\widetilde{\mathbf{p}_1}$ (or reversely, integrate $\tilde{\mathbf{g}}$ backward from $\widetilde{\mathbf{p}_1}$ until we reach $\widetilde{\mathbf{p}_0}$). The second approach is to extract $\widetilde{\mathbf{x}}$ and use it as seeding curve of a stream surface integration of $\tilde{\mathbf{f}}$ until it reaches $\widetilde{\mathbf{y}}$ (or reversely, integrating $\tilde{\mathbf{f}}$ from $\widetilde{\mathbf{y}}$ to $\widetilde{\mathbf{x}}$). The first approach has two disadvantages over the second one: first, a stream surface integration of $\tilde{\mathbf{f}}$ is cheaper then a stream surface integration of $\tilde{\mathbf{g}}$ because $\tilde{\mathbf{f}}$ has a simpler structure (see section 3). Second, a stream surface integration of $\tilde{\mathbf{g}}$ starting from $\widetilde{\mathbf{p}_0}$ may partially leave $\widetilde{D}$ before reaching $\widetilde{\mathbf{p}_1}$. Hence we prefer the second approach. The extraction of $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ turns out to be simple: in the example, $\widetilde{\mathbf{x}}$ consists of all points (x_{min}, y, z, t) with $\mathbf{s}(x_{min}, y, z, t) = (0,0,0)^T$. Keeping x_{min} constant, this can be considered as finding PV lines in a steady 3D flow field in the (y,z,t)-domain. Thus, $\widetilde{\mathbf{x}}$ can be found by applying the algorithm of section 2 for the (y,z,t)-domain. Similarly we find $\widetilde{\mathbf{y}}$.

Another simple example is shown in figure 8a. Here we have a closed PV line $\widetilde{\mathbf{p}_0}$ at the time t_0 which moves over time to the line $\widetilde{\mathbf{p}_1}$ at $t = t_1$. To extract the PV surface, we might integrate $\tilde{\mathbf{g}}$ starting from $\widetilde{\mathbf{p}_0}$. In order to be consistent with the example before, we prefer to pick a point $\widetilde{\mathbf{x}_0}$ on $\widetilde{\mathbf{p}_0}$ [8] and apply a stream line integration of $\tilde{\mathbf{g}}$ starting from $\widetilde{\mathbf{x}_0}$ until it leaves $\widetilde{D}$ in a point $\widetilde{\mathbf{x}_1}$. This stream line is used as seeding structure for a stream surface integration of $\tilde{\mathbf{f}}$.

The last simple example is shown in figure 8b. Here a closed PV line appears at the time t_0 in the closed collapse bifurcation point $\widetilde{\mathbf{p}_0}$, grows over time ($t = t_1$) to $\widetilde{\mathbf{p}_1}$ until at $t = t_2$ it touches the boundary of $\widetilde{D}$ in the inflow boundary bifurcation point $\tilde{\mathbf{c}}$. From this moment on it is an open PV line $\widetilde{\mathbf{p}_3}$ which creates an intersection curve $\widetilde{\mathbf{x}}$ with the boundary of $\widetilde{D}$. In order to get a seeding structure for this example, we first extract $\widetilde{\mathbf{x}}$ similar to the example in figure 7b. Then we apply a stream line integration of $\tilde{\mathbf{g}}$ starting from $\tilde{\mathbf{c}}$ until it ends in $\widetilde{\mathbf{p}_0}$.

In order to extract PV surfaces in $\widetilde{D}$, we provide two algorithms. Algorithm 1 describes how to get a seeding structure, i.e., a set of lines in $\widetilde{D}$ such that a stream surface integration of $\tilde{\mathbf{f}}$ starting from them gives the complete PV surface. Based on this, algorithm 2 describes how to extract and visualize the PV surfaces for a particular time interval.

Algorithm 1 (getting the seeding lines):

1. Compute the intersection curves of the PV surface with the spatial boundaries of $\widetilde{D}$ (see figure 7b for an example).
2. Extract all local bifurcations introduced in section 4.
3. Extract closed PV lines at the times $t = t_{min}$ and $t = t_{max}$ respectively, pick a point on each extracted closed line, and apply a stream line integration of $\tilde{\mathbf{g}}$ starting from them until they leave $\widetilde{D}$ or end in a closed collapse bifurcation. Figure 8a shows an example.
4. Start a stream line integration of $\tilde{\mathbf{g}}$ from all inflow boundary bifurcations until it ends in a closed collapse bifurcation or leaves $\widetilde{D}$ (see figure 8b for an example).

[7] In the field of time-dependent vector field topology, we have similar fold bifurcations for critical points [23] and isolated closed stream lines (cyclic fold bifurcation [22]).

[8] We use one of the points which were necessary to extract $\widetilde{\mathbf{p}_0}$ – see section 2

Then the set of all lines obtained in steps 1–4 is the searched seeding structure. This algorithm needs some comments:
To 1: We have to find the PV lines at the faces of $\widetilde{D}$. This means to apply six times the algorithm for extracting PV lines of static vector fields as described in section 2.
To 2: All inner bifurcations can be found by applying a recursive subdivision approach as described earlier. Boundary bifurcations can be found by checking the lines from step 1 of algorithm 1 for local extremal values of the t-component.
To 3: We get the closed stream line by applying the static algorithm of section 2 to find closed PV lines in the first and last time step.

Algorithm 1 can be considered as a preprocess of the actual PV extraction algorithm described in the following

Algorithm 2 (extract and visualize the PV surface for a time interval $[t_0,t_1]$ with $t_{min} \le t_0 \le t_1 \le t_{max}$):

1. Load the seeding lines obtained from algorithm 1.
2. Identify all parts of the seeding lines with t-values between t_0 and t_1.
3. Starting from these seeding lines, apply a stream surface integration of $\widetilde{\mathbf{f}}$ until it leaves $\widetilde{D}$ or returns to its starting point.
4. Visualize the stream surfaces obtained in 3.

Note that although algorithm 2 guarantees that all PV surfaces are found, it does not guarantee that each surface is found only once. In fact, an open PV surface is extracted twice, by integrating from both exit curves of $\widetilde{D}$. In the current implementation we did not consider this problem and visualized parts of the PV surfaces twice.

Out-of-core considerations:
3D time-dependent fields tend to be larger than the main memory of high-end workstations. Thus, an out-of-core data handling is preferable. We show for our algorithm that only a certain part of the data has to be in memory at once, and that (in worst case) the whole data set has to be loaded only twice. We assume that certain time intervals of the data can be loaded into memory separately, e.g. the data may come as a sequence of static 3D vector fields, one for each time step: by loading the vector fields of two consecutive time steps t_i and t_{i+1} and applying a linear interpolation, we obtain the time-dependent vector field in that interval.

Algorithm 1 can be executed by treating time slices consecutively, but not in one sweep through the data since $\widetilde{\mathbf{g}}$ needs to be integrated in both directions. Thus, we make one forward sweep through the data collecting the local bifurcations and integrating $\widetilde{\mathbf{g}}$ in forward direction. While doing this, we build up 6 static 3D vector fields representing the spatial boundaries of the domain over time. They serve as input for step 1 of this algorithm. In a following backward sweep we integrate $\widetilde{\mathbf{g}}$ backwards starting from the already collected seeding points.

Since a stream line of $\widetilde{\mathbf{f}}$ always stays in the same time level, the stream surface integration of algorithm 2 can be applied to smaller subintervals independently if the data of the original time interval $[t_0,t_1]$ does not fit into main memory.

6 Applications

Figures 4–6 show examples of local bifurcations in constructed quadrilinear vector fields $\mathbf{w}_1$, $\mathbf{w}_2$. We show them both to illustrate the bifurcations again and to explain our visualization technique. As projecting the complete PV surface to space leads to selfintersections already in quite simple settings, we use the following approach to visualize the evolution of PV structures: at a given time we draw the PV lines as red tubes inside the PV surface that is displayed only for a certain time range for future and past. At the boundary the corresponding seeding lines from algorithm 1 are given for a larger time interval. Both the surfaces and the seeding lines fade out away from the current time. We use color coding to indicate past (red) and future (gray). Figure 4 shows the evolution of a saddle bifurcation. Note that the width of the surface in figures 4a and 4c confirms the intuition that the most drastic movements of the PV line over time takes place near the bifurcation points.

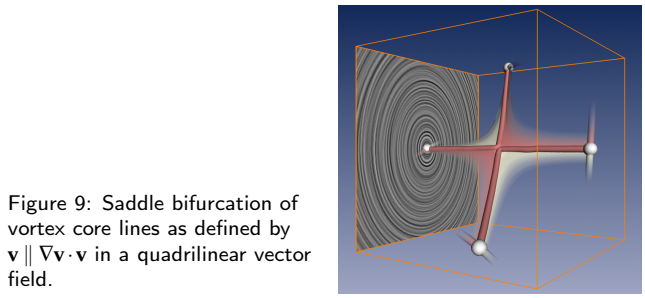
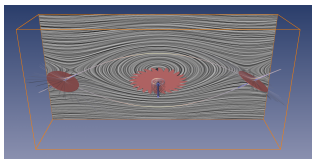
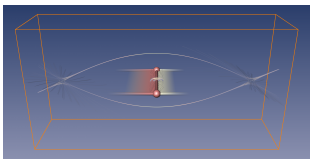


Figure 9: Saddle bifurcation of vortex core lines as defined by $\mathbf{v} \parallel \nabla\mathbf{v}\cdot\mathbf{v}$ in a quadrilinear vector field.



(a) Critical points and surrounding stream lines.



(b) Vortex core line together with its PV surface showing the evolution of the vortex over time.

Figure 10: Stuart Vortex moving over time from left to right.

Figure 6 shows a reversed outflow boundary bifurcation leading to the birth of a PV line. We omitted to display the PV surface for this and the following example. Figure 5 shows an inflow bifurcation and a subsequent closed collapse bifurcation in the green point. Note that in figure 5a, the location of the future inflow bifurcation is already shown by the grey semi-transparent point.

Now we proceed to applying our parallel vector based theory to vortex core line extraction. To do so, we consider the vortex core line concept defined by Sujudi and Haimes [19] searching for all locations with $\mathbf{v} \parallel \nabla\mathbf{v}\cdot\mathbf{v}$ in regions where $\nabla\mathbf{v}$ has a pair of complex eigenvalues. As already mentioned in section 2.1, we have equivalently chosen $\mathbf{w}_1 = \mathbf{v}$ and $\mathbf{w}_2$ as the eigenvector corresponding to the only real eigenvalue of $\nabla\mathbf{v}$ in the regions of interest. Before we apply the technique to a real data set, we analyze whether the bifurcations introduced in section 4 may appear for vortex core lines defined by [19] for piecewise low-degree vector fields. It turns out that the inner bifurcations do not exist inside a cell for a piecewise linear vector field in space-time[9]. For piecewise quadrilinear vector fields, all bifurcations can occur inside a cell. Figure 9 shows an example of a quadrilinear vector field containing a saddle bifurcation.

Consider Figure 10 that demonstrates our vortex core line tracking approach for visualizing a moving Stuart vortex. A Stuart vortex is a well-known vortical structure in fluid dynamics which can be described by a closed formula. Figure 10a shows that there is a critical point on the moving vortex core line. Figure 10b shows the vortex core line together with the PV surfaces indicating its past and future behavior. It shows that our FFF integration did not get stuck in the critical point.

Figures 1 and 11 demonstrate the results of our method applied to vortex core line tracking in a flow behind a circular cylinder.

[9] This is similar to the fact that e.g. fold bifurcations do not exist inside a cell for piecewise linear time-dependent vector fields.

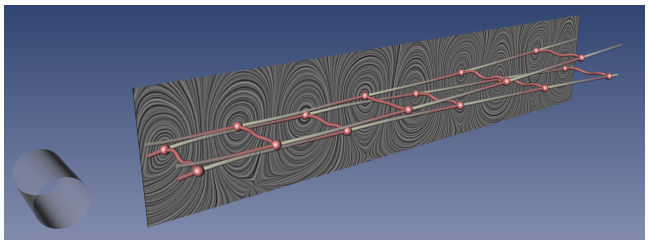


Figure 11: Flow behind a circular cylinder. The extracted seeding lines elucidate the alternating evolution of the vortical structures in transverse direction.

As above, we chose the vortex core line definition $\mathbf{v} \parallel \nabla\mathbf{v} \cdot \mathbf{v}$. The data set was derived by Bernd R. Noack (TU Berlin) from a direct numerical Navier Stokes simulation by Gerd Mutschke (FZ Rossendorf). It resolves the so called 'mode B' of the 3D cylinder wake at a Reynolds number of 300 and a spanwise wavelength of 1 diameter. The data is provided on a $265 \times 337 \times 65$ curvilinear grid as a low-dimensional Galerkin model [11, 27]. The examined time range is $[0, 2\pi]$. The flow exhibits periodic vortex shedding leading to the well known von Kármán vortex street. This phenomenon plays an important role in many industrial applications, like mixing in heat exchangers or mass flow measurements with vortex counters. However, this vortex shedding can lead to undesirable periodic forces on obstacles, like chimneys, buildings, bridges and submarine towers.

7 Conclusions

In this paper, we made the following contributions for 3D time-dependent fields:

- We presented feature flow fields which are equivalent to the PV operator.
- Based on the FFF's, we achieved a complete classification of stable local bifurcations of tracked PV lines in saddle bifurcations, closed collapse bifurcations, inflow and outflow boundary bifurcations.
- We presented a new algorithm to extract and track PV lines as a repeated stream line/surface integration of the FFF's. This way, the algorithm is independent of a particular underlying grid of the data. In fact, the accuracy of our method does not depend on the grid resolution but exclusively on the chosen technique and step size for the stream surface integration.

There is a number issues left for future research. First, algorithm 1 can be enhanced such that the set of seeding lines is minimal, i.e., each part of the PV surface is obtained only once. Second, algorithm 1 can be optimized such that the complete data set has to be loaded only once instead of twice.

Acknowledgments

We thank Jan Reininghaus for his great implementational efforts. We thank Bernd R. Noack for fruitful discussions and providing the cylinder data set. All visualizations in this paper have been created using AMIRA – a system for advanced visual data analysis [18] (see http://amira.zib.de/).

References

[1] D.C. Banks and B.A. Singer. Vortex tubes in turbolent flows: Identification, representation, reconstruction. In *Proc. IEEE Visualization 1994*, pages 132–139, 1994.

[2] D.C. Banks and B.A. Singer. A predictor-corrector technique for visualizing unsteady flow. *IEEE TVCG*, 1(2):151–163, 1995.

[3] D. Bauer and R. Peikert. Vortex tracking in scale space. In *Data Visualization 2002. Proc. VisSym 02*, pages 233–240, 2002.

[4] A. Van Gelder. Stream surface generation for fluid flow solutions on curvilinear grids. In *Data Visualization 2001. Proc. VisSym 01*, 2001.

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

[6] J. Hultquist. Constructing stream surfaces in steady 3D vector fields. In *Proc. IEEE Visualization '92*, pages 171–177, 1992.

[7] J.C.R Hunt. Vorticity and vortex dynamics in complex turbulent flows. *Proc CANCAM, Trans. Can. Soc. Mec. Engrs*, 11:21, 1987.

[8] J. Jeong and F. Hussain. On the identification of a vortex. *J. Fluid Mechanics*, 285:69–94, 1995.

[9] T. Minagawa and R. Rado. The 3D marching lines algorithm. *Graphical Models and Image Processing*, 58(6):503–509, 1996.

[10] H. Miura and S. Kida. Identification of tubular vortices in turbulence. *J. Physical Society of Japan*, 66(5):1331–1334, 1997.

[11] B.R. Noack and H. Eckelmann. A low-dimensional galerkin method for the three-dimensional flow around a circular cylinder. *Phys. Fluids*, 6:124–143, 1994.

[12] R. Peikert and M. Roth. The parallel vectors operator - a vector field visualization primitive. In *Proc. IEEE Visualization 99*, pages 263–270, 1999.

[13] F.H. Post, B. Vrolijk, H. Hauser, R.S. Laramee, and H. Doleisch. Feature extraction and visualisation of flow fields. In *Proc. Eurographics 2002, State of the Art Reports*, pages 69–100, 2002.

[14] M. Roth and R. Peikert. A higher-order method for finding vortex core lines. In *Proc. IEEE Visualization '98*, pages 143–150, 1998.

[15] J. Sahner, T. Weinkauf, and H.-C. Hege. Galilean invariant extraction and iconic representation of vortex core lines. In *Proc. EuroVis 2005*, 2005.

[16] M. Sato and R. Peikert. Core-line-based vortex hulls in turbomachinery flows. *J. Visualization Society of Japan*, 23(2):151–154, 2003.

[17] G. Scheuermann, T. Bobach, H. Hagen K. Mahrous, B. Hamann, K. Joy, and W. Kollmann. A tetrahedra-based stream surface algorithm. In *Proc. IEEE Visualization 01*, pages 151 – 158, 2001.

[18] D. Stalling, M. Westerhoff, and H.-C. Hege. Amira: A highly interactive system for visual data analysis. *The Visualization Handbook*, pages 749–767, 2005.

[19] D. Sujudi and R. Haimes. Identification of swirling flow in 3D vector fields. Technical report, Department of Aeronautics and Astronautics, MIT, 1995. AIAA Paper 95-1715.

[20] C.K. Tang and G. Medoni. Extremal feature extraction from 3-D vector and noisy scalar fields. In *Proc. IEEE Visualization 1998*, pages 95–102, 1998.

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

[22] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Stream line and path line oriented topology for 2D time-dependent vector fields. In *Proc. IEEE Visualization 2004*, pages 321–328, 2004.

[23] 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.

[24] J. van Wijk. Implicit stream surfaces. In *Proc. Visualization 93*, pages 245–252, 1993.

[25] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel. Boundary switch connectors for topological visualization of complex 3D vector fields. In *Proc. VisSym 04*, pages 183–192, 2004.

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

[27] H.-Q. Zhang, U. Fey, B.R. Noack, M. König, and H. Eckelmann. On the transition of the cylinder wake. *Phys. Fluids*, 7(4):779–795, 1995.