

## Topological Methods for Visualizing Vortical Flows

Xavier Tricoche<sup>1</sup> and Christoph Garth<sup>2</sup>

<sup>1</sup> Computer Science Department, Purdue University, West Lafayette, Indiana, USA  
xmt@purdue.edu

<sup>2</sup> Department of Computer Science, University of California, Davis, USA  
cgarth@ucdavis.edu

**Summary.** The paper describes the application of topological methods to the visualization of vortical flow patterns that arise in simulations from Computational Fluid Dynamics. Two techniques are presented: the first is concerned with the exploration of complicated, instantaneous flow structures while the second one permits the visualization of their temporal evolution in large-scale transient simulations. In both cases the mathematical framework is derived from the notion of parametric topology. This yields a unified formalism that permits to efficiently address the challenges raised by typical flow problems. The benefits of this approach are demonstrated in the analysis and visualization of transient vortical flows that undergo the phenomenon of vortex breakdown.

### 1.1 Introduction

Scientific computing is an important tool for the development of new prototypes in the design of modern aircrafts. While the basic theoretical principles of aerodynamics are well established, they are applicable to large scale problems only and do not describe the increasingly important details on small scales. The quality of numerical models has risen to a point where simulations can fill this gap. As the demand for faster aircrafts and improved security is high, they have established themselves as an extremely valuable alternative to physical experiments. Aside from the validation of prototypes, simulations can help to increase our understanding of the dynamics of some of the more complex flow patterns that keep appearing in aviation-related problems.

A prominent example is vortex breakdown. This phenomenon has stood in the way of a wide application of delta-wing type aircrafts as it limits the controllability in critical flight situations and causes damage to the aircraft through the induced pressure differences. In order to understand the origin of this phenomenon and avoid its occurrence in future designs, it has been reproduced and is now investigated in numerical simulations. In this case like in Computational Fluid Dynamics problems in general, these simulations facilitate complicated flow experiments and provide accurate measurements of multiple quantities over the whole 3D domain considered.

However, this comes along with a hindrance for analysis at the post-processing stage. Since detailed models require fine resolutions, the amount of generated data is enormous which is especially true for time-dependent problems. This obstacle must be properly addressed by visualization methods, as they are essential to assist and improve the evaluation of the resulting numerical data sets.

For the analysis of planar flows, flow topology has proven valuable in distilling a complete structural picture of the prevalent structures by an analysis of the critical points and separatrices of the flow vector field. Parametric topology has extended this methodology to time-dependent flows. The resulting visualization is expressive, while the algorithms are efficient. Therefore, planar flow topology can be regarded as mostly complete as a flow analysis tool. Unfortunately, the extension to three dimensions is far from having achieved the same quality of visualization. This can be in part attributed to the fact that the elements involved (e.g. separating surfaces) are inherently 3D. A full display of non-trivial 3D topology is very complicated at best and suffers from mutual obstruction of the corresponding primitives. Approaches exist for simplified depictions of 3D topology, however, they remain rather unsatisfying in comparison to their 2D counterparts. Therefore the aim of this work is to provide a visualization approach for complex 3D flows that inherits the appealing properties of planar flow topology. In that way we are able to complement well established feature extraction methods in a unified framework built upon rigorous mathematical notions.

More precisely, the key idea behind the visualization methods introduced in this paper is the notion of parametric topology. Depending on the considered application the corresponding parameter can be interpreted as the time underlying a transient evolution or as the distance reached along a particular curve that traverses a region of interest. Practically, to obtain accurate and intuitive depictions of intricate flow structures we transform traditional cutting planes into flexible and powerful tools for exploring flow volumes in a continuous way. These moving cutting planes smoothly travel along trajectories that can be either obtained automatically by standard feature extraction schemes or directly provided by the user to explore a particular region. We accurately track the parametric vector field topology captured on the cutting planes. This allows us to dissect the 3D flow, detect and visualize essential properties of the flow, especially for recirculation bubbles which are key features of vortex breakdown. While understanding of this phenomenon is still incomplete, it is known that it is characterized by the appearance of stagnation points (critical points of the flow velocity field) on the vortical axis. To gain insight from the temporal behavior of the stagnation points, the critical point tracking from 2D parametric topology is extended to 3D vector fields defined over tetrahedral grids. For visualization, the four-dimensional trajectories are reduced to two dimensions by using the symmetry inherent to the vortical structures. To further enhance the understanding of the full 3D flow pattern, we also incorporate stream surfaces into the representation.

The paper is structured as follows. Section 1.2 summarizes previous and related work. In Section 1.3, we recall essential theoretical notions of steady and parametric flow topology. In this context we also provide a detailed discussion of the Poincaré index. The tracking of vector field critical points with respect to a parameter change is discussed in Section 1.4 along with the corresponding algorithm. Next, Section 1.5

introduces the moving cutting plane approach. We complete our presentation with our visualization results for two CFD data sets in Section 1.6 and conclude in Section 1.7.

## 1.2 Related Work

The importance of topology for depicting flow fields was first recognized by Helman and Hesselink [7] and resulted in a 2D visualization method. Complete 3D topology has not been attempted yet, however there are authors that examine subsets, such as Globus et al. [6] and Theisel et al. [17] using saddle connectors. Tricoche et al. [18] describe how the time-tracking of singularities and the corresponding topological variations can be investigated for instationary 2D vector fields. Theisel and Seidel also propose a method for the tracking of critical points in more general settings by integrating streamlines of the derived feature flow field [16]. However, the construction of this field is prohibitively expensive for large data sets.

Concerning the temporal variation of features, there are approaches that detect features in several time steps and perform a matching procedure to extract their evolution (e.g. Silver and Wang [14] and Samtaney et al. [12]). Making explicit use of the temporal interpolation, Weigle and Banks [19] extract features in the form of four-dimensional isosurfaces. A similar course is followed by Bauer and Peikert [2]. They incorporate a scale-space approach into their method for the tracking of vortex cores. As to the interrelations among multiple features over time, Silver et al. [3] have developed the *Feature Tree* that is remotely related to the much simpler structural graph we establish here.

In our development of a critical point tracking algorithm on tetrahedral grids, we make use of the Poincaré Index concept, which was described earlier by Mann and Rockwood [9]. They explain its basic premise and show how it can be applied to the study of critical points and other types of singularities. Their work is however limited to the study of analytical vector fields and is not directly applicable to our work.

From the viewpoint of fluid mechanics, vortex breakdown (or vortex burst) has concerned many authors due to its relevance for a large number of applications (see e.g. [10]). In the field of visualization, Kenwright and Haines [8] were among few to write about the detection and visualization of vortex breakdown. They already emphasized its importance in aeronautics. However, their interpretation of vortex breakdown is not in accordance with modern theories.

## 1.3 Theoretical Aspects of (Parametric) Topology

We introduce in this section basic notions of vector field topology both in the 2D and 3D settings as required by the visualization methods discussed in the paper. The emphasis is put on linear structures induced by piecewise linear interpolation over simplicial grids. This choice is justified by the fact that arbitrary grid types can be decomposed into simplices.

### 1.3.1 Phase Portrait, Limit Sets and Separatrices

The essential idea behind topology analysis in the steady case is to characterize the nature of a flow with respect to the asymptotic behavior of its streamlines. For that purpose, one associates the domain of definition of the flow with its *phase portrait* that consists of the set of all streamlines. This corresponds to introducing an equivalence relation that groups all the points located on the path of the same streamline into a single class.

The topological structure of the flow is essentially a subdivision of the phase portrait into regions where all streamlines are asymptotically equivalent, thus forming a uniform flow. More specifically, all streamlines belonging to such a region converge toward the same so-called *limit sets* both forward and backward. Limit sets have a general mathematical definition but for the needs of this presentation we are only interested in critical points and closed orbits. The boundaries of the different topological regions are called *separatrices* and can be either streamlines or *stream surfaces*.

### 1.3.2 Critical Points

The *critical points* (or singular points) of a vector field are the positions where the field magnitude is zero. These points play a fundamental role in the field structure because they are the only locations where streamlines can meet. In the linear case the classification of critical points is based on the eigenvalues of the Jacobian matrix.

In planar fields, depending on the real and imaginary parts of these eigenvalues, there exist several basic configurations, some of which are shown in Fig. 1.1. The *saddle points* are of particular interest since the *separatrices* start or end at their location along the eigenvectors. Note that for every other critical point type, the sign of the real parts of both eigenvalues is either positive or negative, corresponding to a repelling (source) or an attracting (sink) nature, respectively. Thus separatrices emanate from saddle points and end at sources or sinks.

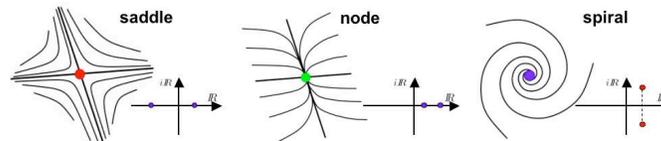


Fig. 1.1. Linear critical points in the plane

In the 3D case, the Jacobian matrix has three eigenvalues, and more combinations exist. If all eigenvalues have a positive (resp. negative) real part the corresponding critical point is a source (resp. a sink). Other cases correspond to different types of saddle points. As in the 2D setting, separatrices of the topology start originate 3D saddle points along their eigenvectors. However these separatrices are either

one-dimensional (streamlines) or two-dimensional (stream surfaces). The latter are spanned by both eigenvectors associated with the eigenvalues whose real parts have same sign. The various cases are illustrated in Fig. 1.2.

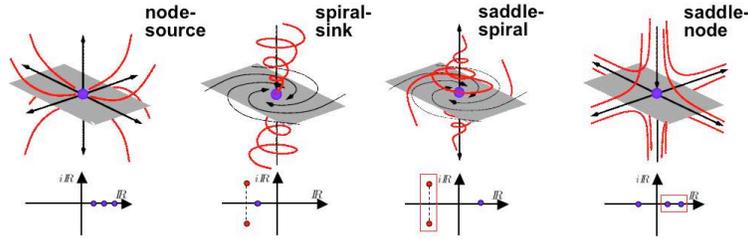


Fig. 1.2. Linear critical points in three dimensions

### 1.3.3 Poincaré Index

#### Definition and Properties

A fundamental concept in planar vector field topology is the so-called *Poincaré index* of a simple (i.e. non self-intersecting) closed curve. It measures the number of rotations of the vector field while traveling along the curve in positive direction (also called *winding number*). In a more mathematical way, one gets the following definition for the index of a simple curve  $\gamma$ :

$$\text{ind}_\gamma = \frac{1}{2\pi} \oint_\gamma d\phi, \text{ where } \phi = \arctan \frac{v_y}{v_x}.$$

( $\phi$  is the angle coordinate of the vector field  $\mathbf{v}(v_x, v_y)$ .) Remark that the index is always an integer.

Similarly one defines the index of a critical point as the index of a simple closed curve around the critical point enclosing no other singular point. For linear critical points, the possible index values are +1 and -1. A saddle point has index -1 whereas every other critical point has index +1. Following properties of the Poincaré index are essential in practice [1].

1. A simple closed curve that encloses no critical point has index 0.
2. The index of a simple closed curve that encloses several critical points is the sum of the respective indices of those critical points.

This notion can be generalized to 3D vector fields. First, one defines the winding number  $\#_x(S)$  of a closed surface  $S$  with respect to a point  $x$  as

$$\#_x(S) := \frac{1}{4\pi} \int_S \frac{y-x}{|y-x|^3} dS(y).$$

It is an integral value as in the planar case and corresponds to the number of times  $S$  wraps around  $x$ . For example, the  $x$ -centered sphere  $S_\varepsilon(x)$  of radius  $\varepsilon > 0$  has the canonical winding number 1. Now, to define the index of a closed surface  $S$  with respect to a three-dimensional vector field, one introduces the notion of *Gauss map*

$$\gamma : \mathbb{R}^3 \setminus \{0\} \rightarrow S^2, x \mapsto \frac{x}{\|x\|},$$

that maps any non-zero vector to its (normalized) direction. The index of a closed surface  $S$  is then defined as the number of times the vector field directions on  $S$  cover the origin as we move around all of  $S$ . In other words, it is the winding number of the Gauss map of  $\mathbf{v}$  restricted to  $S$  with respect to the origin. Mathematically speaking, we have

$$4\pi \text{ind}_S = \#_0(\gamma(\mathbf{v}|_S)) = \int_S \gamma(\mathbf{v}(x)) dS(\gamma(\mathbf{v}(x))). \quad (1.1)$$

Note that the winding number can be read as an oriented area integral of  $\gamma(\mathbf{v}|_S)$ . Hence, the sign of  $\text{ind}_S$  depends on the orientation of  $S$  relative to  $\mathbb{R}^3$ . We are able to define  $\text{ind}_z(\mathbf{v})$  of a singularity  $z$  via

$$\text{ind}_z(\mathbf{v}) := \lim_{\varepsilon \rightarrow 0} \#_0(\gamma(\mathbf{v}|_{S_\varepsilon(z)})). \quad (1.2)$$

The properties mentioned previously for the planar case hold in the three-dimensional case too. Let  $S$  be a closed surface that encloses the vector field singularities  $z_i$ . Then

$$\sum_i \text{ind}_{z_i}(\mathbf{v}) = \#_0(\gamma(\mathbf{v}|_S)).$$

As in the 2D case positive orientation is assumed for all closed surfaces under consideration. From the last equation, we find that the index vanishes if  $S$  does not enclose any singularity in its interior. Observe that the converse holds only in the linear case.

### Computation

In the piecewise linear setting the critical points that may be encountered in the interior domain of each linearly interpolated triangle or tetrahedron cell are of first order and have therefore either index +1 or -1. We consider 3D critical points first and show how the 2D setting can be seen as a special case. To compute the index of an isolated linear 3D critical point  $z$  we can use a simple approach that is based on the Jacobian  $J$  of the corresponding linear vector field. Indeed, assuming a non-degenerate case,  $J$  has full rank which implies that  $|\text{ind}_z(\mathbf{v})| = 1$ . Hence, the index is +1 if  $J$  is orientation-preserving and -1 otherwise. In other words, the index of a linear critical point is determined by the sign of its determinant. Therefore, if we consider the types of linear critical points mentioned previously, a source has index +1 and a sink has index -1. Concerning saddle points, their index depends on the particular type. If the dimension associated with the 1D separatrix corresponds to a source, the index is +1, otherwise -1. Refer to Figure 1.2. If we now consider two dimensional critical points, we easily see that a similar result applies: sink and sources correspond to two eigenvalues of same sign and both types have index +1 while saddle points have index -1, see Figure 1.1.

### 1.3.4 Parameter Dependent Topology

In the case of a parameter-dependent (e.g. instationary) flow, parameter changes entail transformations of the topology. Despite the unlimited variety of such transformations they always preserve qualitative consistency. In particular, the Poincaré index acts as a topological invariant.

For the needs of our method we only mention two types of local bifurcations in the 2D case and derive a similar example for the 3D case. The first local bifurcation in 2D involves pairs of critical points, namely a saddle point and a sink or a source. When both critical points become progressively closer, merge and eventually vanish the bifurcation is a pairwise annihilation. The reverse phenomenon is called pairwise creation. The common terminology for both evolutions is fold bifurcations. The second type of planar bifurcation affects a single spiral critical point, either a sink or a source, and is known as Hopf bifurcation. The corresponding evolution for a planar field is pictured in Figure 1.3. The corresponding critical point (a center) is an

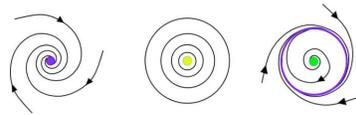


Fig. 1.3. Hopf bifurcation

unstable configuration and any change in the parameter value will transform it into a source surrounded by a closed orbit that behaves as a sink. The reverse evolution is possible too, as well as swapped roles for sinks and sources. Similar transformations occur in the 3D case. A simple example can be obtained by adding a one-dimensional source behavior to a saddle point and a source involved in a fold bifurcation. This creates two 3D saddle points that merge and vanish in the very same way. An illustration is proposed in Figure 1.4. Observe that as in the 2D case the basic ingredient of

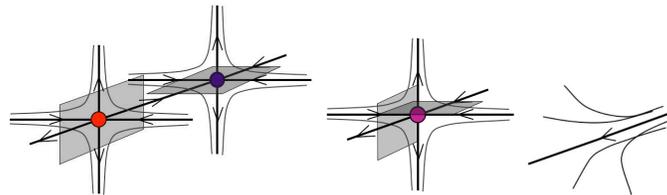


Fig. 1.4. Three-dimensional fold bifurcation

this fold bifurcation is the fact that the overall index of both critical points is 0 (two

saddle points of opposite indices) which corresponds to a neighborhood without singularity. Consequently, the local value of the Poincaré index is preserved throughout the corresponding transformation. This remark also explains why a fold bifurcation in 3D can concern a source (index +1) and a sink (index -1).

## 1.4 Topology Tracking

In the following we describe a simple algorithmic solution to track the *continuous* evolution of the topology and detect the associated bifurcations. The method was originally designed for time-dependent 2D vector fields [18] and has been recently extended to the 3D case [5]. We focus the description hereafter on the latter case, which is in essence very similar to the planar case. In particular, bifurcations on the common boundary of two neighboring cells, tracking through successive entry and exit points, as well as the ability to filter out insignificant details or interpolation artifacts in post-processing are common features of both the 2D and 3D implementation. Section 1.5.1 explains how the 2D method can be applied to steady 3D vector fields. Observe that in both cases, and in contrast to the original method, we do not explicitly track the separatrices of the topology.

### Setting

The objective is to determine the paths of isolated critical points of a time-dependent piecewise-linear vector field, given on a tetrahedral grid. Let  $p_i \in \mathbb{R}^3$  be a set of points and  $v_i^j$  the vector values associated with the  $p_i$  at discrete times  $t_j \in \mathbb{R}$ . Let  $T_k$  be a set of tetrahedra defined on the points  $p_i$ . Then every tetrahedron  $T_k$  gives rise to a vector field  $v(x, t)$  that is linear in both space and time: if  $x \in T_k$  and  $t \in [t_j, t_{j+1}]$ , then set

$$v(x, t) = \sum_{l=0}^3 \beta_l(x) \left( \frac{t - t_j}{t_{j+1} - t_j} v_l^{j+1} + \frac{t_{j+1} - t}{t_{j+1} - t_j} v_l^j \right),$$

where  $\beta_l$  are the barycentric coordinates w.r.t.  $T_k$  and  $l$  refers to the vertices of  $T_k$ . We will next examine the paths of singularities in a single tetrahedron  $T_k$ .

### Bifurcations

Due to the inherent limitations imposed on the singularities by the piecewise linear nature of the vector field, we can conclude that fold bifurcations that involve two critical points can only occur on the common face of two neighboring tetrahedra. Bifurcations on an edge or a vertex are special cases that are numerically highly unstable. Therefore there are not addressed here.

Assume we have two tetrahedra  $T_1$  and  $T_2$  that share a common face on which we find a bifurcation at some time  $t$ . Since the field is linear in both tetrahedra, from the two singularities involved, one is located in  $T_1$  and the other in  $T_2$ . Moreover, due to index conservation, the overall index must be zero. Hence the indices of the critical points must be +1 and -1, respectively.

**Paths in a single cell**

We first consider a single tetrahedron  $T$  and determine what possibilities exist for the path of a singularity  $z$ . To simplify the notations, we assume that the vector field in  $T$  is given in the form

$$v(x) = \sum_{i=0}^3 \beta_i(x) ((1-t)u_i + tv_i), \quad x \in T, t \in [0, 1]$$

and that  $v$  is non-degenerate, i.e. it contains exactly one isolated zero at all times. For fixed  $t$  we can solve for the position of the singularity of this field in barycentric coordinates. For example, with  $w_i(t) = (1-t)u_i + tv_i$  we write (omitting the parameters)

$$v = w_0 + \beta_1(w_1 - w_0) + \beta_2(w_2 - w_0) + \beta_3(w_3 - w_0)$$

and apply Cramer's rule to find

$$\beta_1(t) = \frac{\det(-w_0, w_2 - w_0, w_3 - w_0)}{\det(w_1 - w_0, w_2 - w_0, w_3 - w_0)} =: \frac{b_1(t)}{q(t)}.$$

The same can be done for all  $\beta_i$ . Brief computation shows that the resulting  $b_i(t)$  and  $q(t)$  are polynomials of degree 3 in  $t$ . We required that  $v$  be non-degenerate, this reflects in  $q(t) \neq 0$  for all  $t \in [0, 1]$ . Naturally, if  $\beta_i(t) < 0$  for some  $i$ , the singularity of  $v$  is outside the tetrahedron for this specific  $t$ . In other words, we have found an explicit representation for the location of  $z$ . Taking a closer look at  $b_i$ , we find that the zeros of these polynomials allow us to determine when  $z$  crosses one of  $T$ 's faces. If for  $\hat{t} \in [0, 1]$  we find  $\beta_i(\hat{t}) = 0$  and  $\beta_j(\hat{t}) \geq 0$  for  $j \neq i$ , then the singularity is located on the face of  $T$  opposite the vertex  $p_i$  (its barycentric coordinate is zero). For this case, by evaluating the sign of the derivative

$$\beta'_i(\hat{t}) = \left( \frac{b_i}{q} \right)'(\hat{t}) = \frac{b'_i(\hat{t})}{q(\hat{t})} \quad (\text{since } b_i(\hat{t}) = 0)$$

we can tell if the singularity enters or leaves the tetrahedron at  $\hat{t}$ . We will say that  $T$  has an entrance/exit on face  $F$  at  $\hat{t}$ . This information is important to determine in which neighboring tetrahedron (if one exists for  $F$ ) the singularity path continues.

For fixed  $t \in [0, 1]$  there can be at most one singularity inside  $T$ , hence we can conclude that if there is a singularity in  $T$  at some  $t \in (0, 1)$ , it must either have entered  $T$  at an earlier time  $0 < \hat{t} < t$  or remained in  $T$  since  $t = 0$  (in this case we will say that  $z$  enters at  $t = 0$ ). In complete analogy, it must either exit  $T$  at  $t < \hat{t} < 1$  or remain in  $T$  until  $t = 1$  (read  $z$  exits at  $t = 1$ ). In other words, a singularity path always connects an entrance to an exit, and exits and entrances always come in pairs. Moreover an entrance is always connected to the closest exit (in time).

When  $z$  passes from  $T$  to a neighbor  $T'$  through the face  $F$  at  $\hat{t}$ , there is a singularity on  $F$  in both  $T$  and  $T'$  at  $\hat{t}$ . Two possibilities exist: either we find an exit/entrance combination in  $T$  and  $T'$ , in which case the path continues in  $T'$ , or we find an exit/exit or entrance/entrance combination. In the latter case, the vector field has a fold bifurcation on  $F$  at  $\hat{t}$ , and the paths of both singularities involved start or end on  $F$ .

### 1.4.1 Tracking algorithm

Using previous results, we give in the following a simple scheme for tracking a singularity between two time steps  $t = 0$  and  $t = 1$ . It works by simply connecting entrance/exit path segments over tetrahedron boundaries. Observe that the iterative nature of our scheme allows to restrict to two consecutive time steps the amount of data that is needed at once for processing.

Assume that a singularity  $z$  is present in  $T$  at  $t \in (0, 1)$ . Then, to compute the path forward in time

1. compute the  $b_i$  and  $q$  for  $T$ , and determine entrances and exits
2. if there is no exit later than  $t$ ,  $z$  exits  $T$  at  $t = 1$ ; the path is complete
3. if there are exits in  $T$ , then  $z$  leaves  $T$  at the earliest exit later than  $t$ ; determine the neighbor tetrahedron  $T'$  corresponding to the exit face  $F$  and compute  $b'_i, q'$  for  $T'$
4. if  $T'$  has an exit on  $F$  corresponding to the exit on  $T$  ( $\rightarrow$  bifurcation), the path of  $z$  ends on  $F$
5. otherwise,  $T'$  has an entrance on  $F$  corresponding to the exit on  $T$ ;  $z$  is now in  $T'$ . Set  $T = T'$  and restart at 1.

Following the path of  $z$  backwards in time can be achieved in a completely analogous manner. Both directions are completely equivalent. We use this procedure as a building block for computing the paths of all singularities present in two given time steps between  $t = 0$  and  $t = 1$ :

1. find the sets of tetrahedra  $S_0$  and  $S_1$  that contain a singularity at  $t = 0$  and  $t = 1$  respectively. Let  $B = \{\}$  be the set of bifurcations encountered in between  $t = 0$  and  $t = 1$ .
2. for every  $T \in S_0$ : follow the path of  $z$  forward in time
  - a) if it ends in  $T'$  at  $t = 1$ , eliminate  $T'$  from  $S_1$ .
  - b) if it ends at a bifurcation, add it to  $B$ .
3. for every  $T \in S_1$  (singularities not reached by paths from  $t = 0$ ): follow the path of  $z$  backward in time
  - a) it must end at a bifurcation; add it to  $B$
4. for all bifurcations in  $B$ : check if  $B$  has two paths connecting to it; if it does not, there must be another singularity involved. Follow its path forward or backward in time depending on whether the bifurcation is a creation of singularities or an annihilation.
  - a) the path must end at a bifurcation; add it to  $B$ ; goto 4.

The algorithm avoids multiple tracing of the same path by using the equivalence between forward and backward tracing (i.e. if a path extends from  $t = 0$  to  $t = 1$ , we only need to trace it forward). The test in step 4 is required because non-intuitive situations can occur (see Figure 1.5). The final result is a set of paths that completely describe the continuous structural variation of the vector field between the two time steps. Going to several time steps from here is easy as it only involves connecting the paths from different time steps according to which singularity they start/end at.

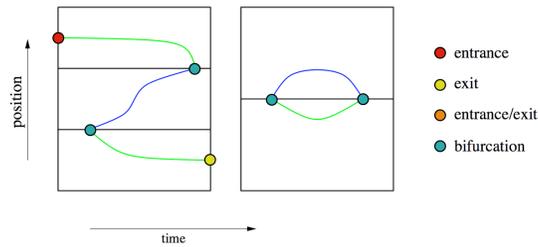


Fig. 1.5. Special cases of cell-wise singularity tracking

Observe that some cases are not covered by the given algorithm. If two bifurcations that create and annihilate a pair of singularities lie between two time steps, neither of the singularities will show up in either time step, and hence their paths will not be discovered by the algorithm (see Figure 1.5). However they do not interact with other singularities and therefore they do not play an important role in understanding the structural changes in between the time steps. Moreover, it is often desirable to ignore small-scale local behavior.

#### 1.4.2 Structural Graph Filtering

To obtain a complete picture of the structural evolution of a given dataset, the interaction of the various singularities form a *structural graph* with bifurcations as vertices and paths as edges (see Fig. 1.7 for an example). We describe here how this graph can be manipulated at the post-processing stage.

The method described above is restricted to tetrahedra which implies that arbitrary input data sets must be tetrahedrized before application. Although the tracking algorithm could be extended to non-tetrahedral grid cells, this would result in a number of special cases that complicate the simple structure of the algorithm. In its present form, implementation is straightforward and fast. However, the price to pay is that tetrahedrization of arbitrary grids can result in the creation of singularities that are not in the original dataset. It is possible that a cell of index 0 is split up such that the resulting tetrahedra have non-zero indices. These artificial singularities are not an issue since they are always created pair-wise and typically only last for a short period of time.

Numerical data sets are often subject to noise, especially if the computations involve some kind of differentiation. It is common practice to apply smoothing operators to data sets in order to account for this limitation. Numerical noise usually reflects in short-lived pairs of artificial singularities that exist in isolation and are not part of the data sets structural evolution over time. It can also occur that a path is interrupted by a pair of artificial bifurcations that enclose a path segment of very short duration (Figure 1.5 (left) gives an example).

What seems a drawback at first can be turned into an advantage: instead of smoothing the dataset we filter the resulting set of singularity paths by removing paths that last less than e.g. one time step. Filtering can be applied on the structural

graph directly and can be implemented in an efficient way by first removing edges that represent paths with short duration and successively removing all isolated vertices. In our experiments, we found this method to be very effective in treating noisy data sets. It must be mentioned here that conventional smoothing does not significantly reduce the number of artificial singularities. Moreover it affects the structure of the dataset in such a way that the structural evolution is obscured or changed (this is especially true for minimum/maximum tracking as described in the next paragraph). Consequently a filtering based directly on the topological structure of the flow offers a much more accurate control over the complexity of the structural information.

### 1.4.3 Algorithm performance

The tracking algorithm itself is of linear complexity in both the number of singularities and the number of time steps. The most time-intensive part is the pre-computation of all singularities in a time step, for which each cell has to be considered individually. This is not a drawback of our algorithm but rather a limitation inherent in this class of tracking algorithms (cf. [18, 16]) If this information is assumed given, the running times for our examples are on the order of very few seconds. Since the algorithm only needs two successive time steps to do its work, it is possible to integrate it directly into the CFD simulation. The structural graph for all time steps can then be completed in post-processing. This would also allow for online supervision of simulations that are still in progress.

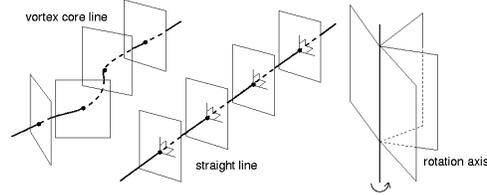
## 1.5 Planar Topology Tracking for Volume Exploration

As mentioned previously the approach developed in this section consists in using the framework of planar topology tracking to explore steady 3D flow structures. More precisely, the 3D (steady) flow is now investigated through the parametric topology of its 2D projection onto a plane that is swept along a given curve across the volume of interest. In other words, the curve provides the third dimension and is interpreted as the parameter space for topology tracking. Essential algorithmic aspects of this method are discussed next.

### 1.5.1 Moving Cutting Planes

#### Trajectories

By definition, the choice of a particular trajectory to explore a flow volume is essential to ensure the quality and the usefulness of the extracted topology and must therefore be carefully chosen with respect to the considered application. The general idea followed in our implementation is to use any inherent symmetry of the data set to yield a natural way to split the physical space. Since the application of our technique is focused on vortical flows and vortex breakdown we selected the curves described next. Refer to Figure 1.6 for an illustration.



**Fig. 1.6.** Different types of moving cutting planes

- *Vortex Core Lines* are a natural choice to explore vortical regions. We extract them using an implementation of the standard method by Sujudi and Haines [15] based on the Parallel Operator [11]. A smoothing step applied in preprocessing permits to improve the results.
- *Straight Line* across the grid are a straightforward alternative. They are mainly used to explore large structures whose overall orientation is known.
- *Recirculation Bubble Axis*: to explore the recirculation bubbles induced by vortex breakdown, the medium axis usually exhibited by these structures can be used. More specifically, as mentioned previously a recirculation bubble is delimited by two stagnation points and the line connecting them plays the role of a rotation axis in our method.

### Cutting Plane Orientation

A robust computation of the cutting plane orientation is mandatory for our flow exploration technique. It can be seen on Figure 1.6 that choosing the recirculation bubble axis as exploratory curve fully determines the plane orientation. Similarly we also used the straight line as plane normal when it is selected to capture large-scale features. On the contrary, when dealing with a vortex core line the inaccuracy of the extraction method results in an approximated position of the actual vortex core which can have a negative impact on the resulting normal value. The same holds true when approximating the curved, possibly complex path of a vortex by a straight line segment. In both cases we need an automatic way to compute a suitable normal at each point along the discrete path according to the local flow orientation. Practically, the quality of a normal is evaluated with respect to the amount of flow crossing the plane over a small region around the considered point. We use a simple iterative scheme to maximize this quantity which consists in assigning in every step the mean direction of crossing flow to the current approximation of the normal.

### Planar Resampling

The remaining task consists in resampling the 3D vector field on the cutting plane while ensuring consistency of the coordinate frames between consecutive positions along the followed curve. This is mandatory to obtain meaningful results during the topology tracking procedure. To do so, it is sufficient to assign a single basis vector to

each plane, the second one being readily obtained by cross product with the normal. Practically we select an arbitrary vector in the first plane and we iteratively transport this vector from one plane to the next by successive projections and normalization, similar to e.g. [13]. Once provided with the grid resolution (i.e. sampling rate) we only need to control the spatial extent of the sampled region around the curve on the plane. This is to ensure that the sampling will not include data points corresponding to positions lying outside the structure of interest. Practically we either assume a constant size of the feature or we apply the technique described in a recent paper [4] to determine the boundary of the vortex core region.

### 1.5.2 Topology Tracking

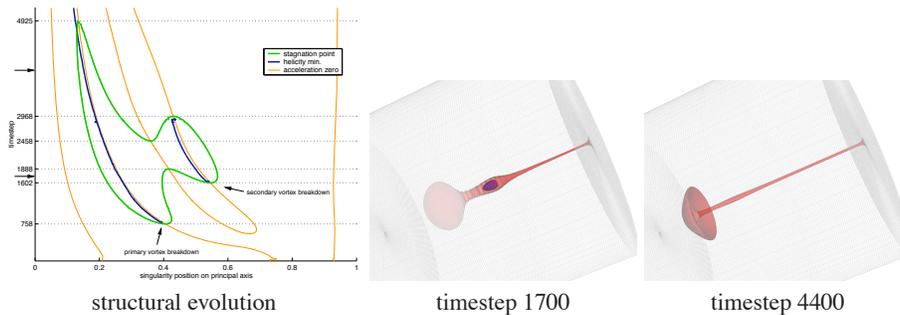
The previous step collects the successive values of the projected vector field as the cutting plane moves through the volume. As mentioned previously we can now abstract these data from their original embedding in three-space and treat them as the successive states of a parameter-dependent planar vector field. In that way we can apply the two-dimensional tracking scheme proposed in [18] and whose extension to three dimensions was discussed in the previous section. In essence the setting of the original method corresponds here to the computational space. One difference is that the results obtained (singularity paths, bifurcations) must be mapped back into physical space after tracking for visualization and interpretation. Moreover we need to account for the lack of smoothness of the vector field projected on the moving cutting plane along a curve. Specifically this may cause spiraling critical points to oscillate between sink and source behavior, creating numerous Hopf bifurcations. We correct this effect by filtering out small-scale features like pairs of critical points vanishing shortly after their creation or type swap between sources and sinks. The latter is handled by assigning the type *center* to the critical point. Although this is an unstable structure in planar topology, this may be monitored in cutting plane topology when inspecting a vortex whose spiraling flow neither converges nor diverges with respect to its core line.

## 1.6 Results

We now show the results of both topology tracking methods applied to two CFD simulations specifically designed to investigate the impact of vortex breakdown on a vortical flow.

### 1.6.1 Data sets

**Delta wing:** This simulation describes a sharp-edged delta wing at subsonic speed (0.2 mach) with the characteristic vortical systems above the wing. The angle of attack increases over time, eventually leading to vortex breakdown in later time steps. The viscous simulation of the full configuration was performed without the assumption of symmetry and was carried out using the DLR Tau Code solver. The grid

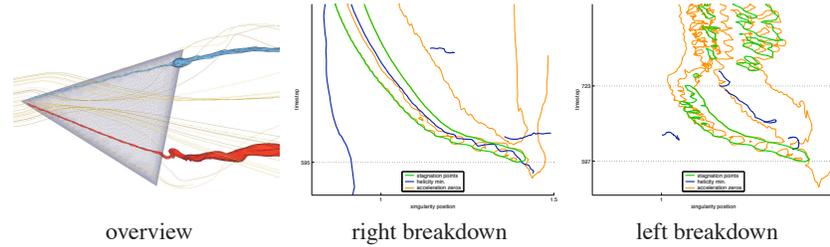


**Fig. 1.7.** *Left:* Structural graph of the can dataset. The green paths represent the stagnation points in the velocity field. Primary and secondary breakdown each create a pair of stagnation points. Around time step 1888, the two phenomena join, only to re-split at time step 2458 and successively decay. The blue and orange paths belong to helicity minima and acceleration zeros. Note the strong interrelation between the three quantities. **Middle and right:** Two snapshots from the can dataset. Separation stream surfaces are started at the singularity positions. Time step 1700 shows both breakdowns, whereas the second breakdown has already vanished in time step 4000 and the first breakdown shows the typical “mushroom” structure.

consists of 11.1 million unstructured grid cells based on about 3 million vertices. At these, a number of variables is given (velocity, pressure, kinetic energy, etc.) for each of the 90 time steps. Among the significant physical features are secondary and tertiary vortices on the wing and corresponding separation and attachment structures. **Can data set:** The aim behind this simulation of a cylindrical container filled with an incompressible and highly viscous liquid was to study vortex breakdown under ideal conditions, created by the viscosity of the fluid and the high symmetry in the problem that lead to numerically very accurate and smooth data. The bottom cylinder cap rotates, creating a vortex on the symmetry axis of the cylinder. A variation in the rotation speed leads to the appearance and successive vanishing of vortex breakdown during the 500 time steps. The dataset is given in the same form as the delta wing dataset, with the grid containing approx. 750.000 elements.

### 1.6.2 3D Critical point tracking

We have employed the critical-point tracking described in Section 1.4.1 for the analysis of both data sets. It is already known that vortex breakdown is associated with the occurrence of (pairwise) stagnation points, therefore we have applied the tracking algorithm to the velocity fields first. Furthermore, there are speculations that both acceleration and signed helicity (i.e. dot product of velocity and vorticity) play an important role in this context. We have computed these fields for those data sets and applied tracking to them as well, in the case of signed helicity minimum tracking was performed. Since these computations involve derivative computation, we observe strong numerical noise in both helicity and acceleration yielding many artificial



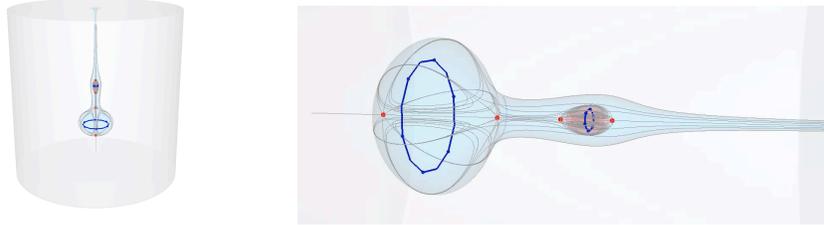
**Fig. 1.8. Left:** Overview of the delta wing dataset with its two primary vortices above the wings. Stream surfaces wrap around the vortices and are eventually distorted by vortex breakdown. Note the asymmetrical breakdown structure. **Middle and right:** Structural graphs for right and left breakdown. Again a connection between various quantities involved in vortex breakdown can be observed for the right breakdown. In the left breakdown, several oscillating breakdown structures are visible in the later time steps.

singularities. Using structural graph filtering we are still able to obtain meaningful results.

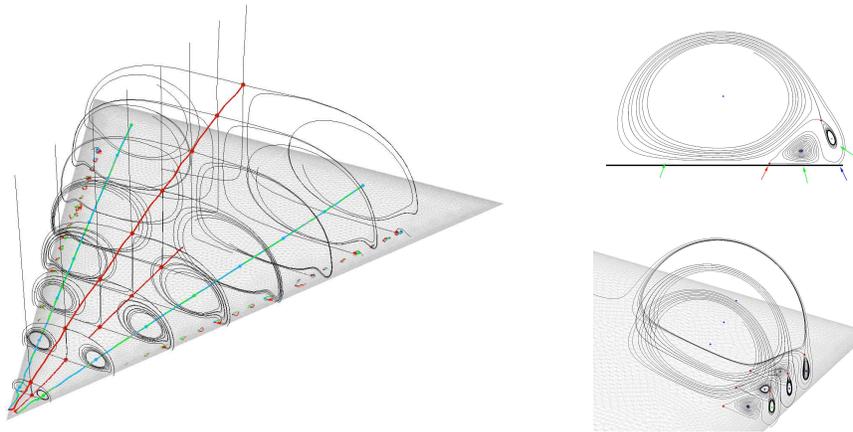
For the can dataset, the results are of almost analytical quality (see Figure 1.7). The simulation actually shows two occurrences of vortex breakdown (and two corresponding pairs of stagnation points) and it is interesting to observe how primary and secondary vortex breakdown successively merge and split again. Acceleration zeros and helicity minima show a strong correlation with the onset of the breakdown process and the bifurcation that creates the two stagnation points. It is also obvious that the structural graph helps locate interesting time steps quickly.

In treating the delta wing dataset, we focus on two regions that correspond to breakdown on both sides of the wing. After a coordinate transformation consisting in a projection onto the vortical axis, the structural graph of the right region (cf. Figure 1.8) clearly shows the evolution of the stagnation points as they move towards the wing. Again, acceleration zeros and a helicity minimum seem to play a role in formation of breakdown, although the correlation is not as obvious as in the can dataset. This is, in part to be blamed upon the lack of resolution and numerical noise. Filtering of the structural graph for the helicity gradient field reduces the number of meaningful paths from roughly 1,000 to 4, effectively eliminating all artificial singularities. The left region is even more complex, and it is clearly visible how the stagnation points begin to oscillate and disappear around time step 730, to be followed by what appears to be a sequence of short-lasting vortex breakdowns in different places. In this case, the structural graph makes identification of multiple breakdown bubbles possible by grouping the velocity field singularities according to their common origin in a bifurcation. The stream surfaces shown are separation surfaces originating in the separation planes of the (saddle) stagnation points. Although this picture conveys the basic structure of the breakdown bubbles, for an accurate interpretation the structural graph is necessary.

### 1.6.3 Moving cutting planes



**Fig. 1.9.** *Left:* An overview of the can dataset. *Right:* Parametric topology shows the essentials of vortex breakdown including the recirculation ring (blue) and a secondary vortex breakdown. To show that the separatrices accurately model the flow behavior, the breakdown bubbles are surrounded with transparent stream surfaces [4] (light blue/light red) originating at the upstream stagnation points that are reproduced as saddle points in the topology of the planes (red).

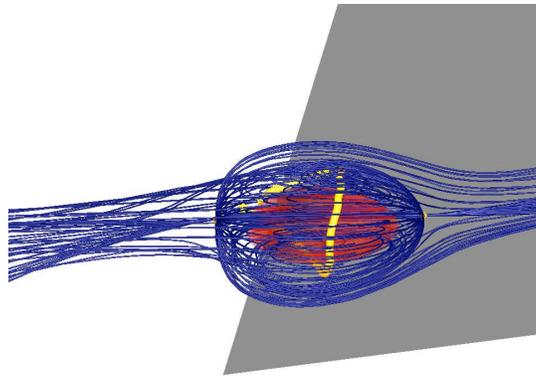


**Fig. 1.10.** *Left:* An overview of the delta wing dataset: parametric topology visualizes the primary vortices. The planes are computed along the symmetry axis of the wing and are parallel. Each planes shows two sinks/sources (primary vortices) and a number of saddle points (separation from the wing). Note how the separatrices end in cycles. This indicates very weak attracting/repelling behavior of the vortices. *Right:* Primary, secondary and tertiary vortices visualized by planar topology. Here, the planes are on the primary vortex core and oriented to the flow. Note how plane orientation affects the resulting structures. Green arrows indicate the three vortices in the top image. The red arrow shows the separation sheet between primary and secondary vortex. The primary separation at the wing edge is indicated by the blue arrow. All three vortices are present as expected.

The moving cutting planes scheme from Section 1.5 was applied to both data sets with the aim of investigating the flow structures obtained by the simulations. For the delta wing dataset, the reproduction of primary, secondary and tertiary vortices is crucial. Figure 1.10 left gives an overview of the wing created with parallel cutting planes along the wing symmetry axis. The primary vortices are presented prominently, and the vortex axis results from the tracking of the corresponding singularities. Using the cutting plane orientation scheme described in Section 1.5.1 with the vortex core as input curve for the plane generation, both secondary and tertiary vortices are visible. Moreover, the planar cut reveals interactions between the three vortices that are hard to determine by other means. This includes the separation surface between the primary and secondary vortices and the so-called primary separation, i.e. the flow sheet that emanates from the wing edge and divides the flow above the wing from the surrounding flow. Both appear as a separatrix in the plane.

The dataset had been examined for the presence of the vortical system before, using the method of Sujudi and Haines [15]. However, this scheme requires careful computation of derivatives and involves smoothing. The result is a set of disconnected line segments and is hard to interpret. In comparison, the approach employed here was easily applied. This can be attributed in part to the fact that the approximate location of the sought features was a priori known, which is usually the case in the verification of data sets.

Application of the planar topology to the can dataset has revealed a peculiarity. The simulation exhibits vortex breakdown, hence a so-called breakdown bubble is visible. Over time, this bubble grows, merges and successively re-splits with a second bubble, and shrinks until it vanishes as the breakdown is resolved.



**Fig. 1.11.** *Enlargement of the right recirculation bubble in the delta wing data set. Continuous tracking on the projected topology onto a plane rotating around the axis connected the 3D stagnation points. The resulting parametric topology exhibits numerous Hopf bifurcations that are smoothed out to yield the center type critical point (yellow) corresponding to the typical closed vortex core.*

Aside from the strict validation of data sets, parametric planar topology can also serve as a feature extraction method for vortex core lines under limited circumstances. For example, the primary vortex axes in the delta wing dataset can be extracted in this manner (cf. Figure 1.10). Although it is in this case equivalent to other algorithms, it excels in the extraction of recirculation cores. As the vortex breakdown bubble encloses a mostly rotation symmetric region of recirculation, there is essentially a bent vortex inside the bubble. Its core appears as a singularity in the section planes revolving around the original vortex axis. Then, tracking provides a connection between different planes and thus constructs the core of the recirculation vortex. Figures 1.9 and 1.11 show these recirculation rings.

## 1.7 Conclusion

We have introduced a unified algorithmic framework to address the challenging task of analyzing and visualizing the flow structures exhibited by typical CFD simulations of complex vortical flows. Building upon the central idea of parametric topology we investigate 3D flow patterns like vortical systems and recirculation bubbles to yield intuitive and accurate representations. Moreover we extend a topology tracking scheme originally designed for 2D transient vector fields to the three-dimensional setting and show how to leverage it to efficiently explore large time-dependent data sets and understand the temporal evolution of features of interest. The corresponding algorithms are easily implemented and suitable for the processing of typical CFD data sets, both online during numerical simulations and at the post-processing stage.

## References

1. A. A. Andronov. *Qualitative Theory of Second-Order Dynamic Systems*. John Wiley & Sons, 1973.
2. D. Bauer and R. Peikert. Vortex tracking in scale-space. In *Data Visualization 2002. Proc. VisSym '02*, 2002.
3. J. Chen, D. Silver, and L. Jiang. The feature tree: Visualizing feature tracking in distributed amr datasets. In *IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, 2003.
4. C. Garth, X. Tricoche, T. Salzbrunn, and G. Scheuermann. Surface Techniques for Vortex Visualization. In *Proceedings Eurographics - IEEE TCVG Symposium on Visualization*, May 2004.
5. C. Garth, X. Tricoche, and G. Scheuermann. Tracking of Vector Field Singularities in Unstructured 3D Time-Dependent Datasets. In *IEEE Visualization Proceedings '04*, October 2004.
6. A. Globus, C. Levit, and T. Lasinski. A Tool for Visualizing the Topology of Three-Dimensional Vector Fields. In *IEEE Visualization Proceedings*, pages 33 – 40, October 1991.
7. J. L. Helman and L. Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May 1991.

8. D. N. Kenwright and R. Haimes. Vortex Identification - Applications in Aerodynamics: A Case Study. In R. Yagel and H. Hagen, editors, *IEEE Visualization '97*, pages 413–416, Los Alamitos, CA, 1997.
9. S. Mann and A. Rockwood. Computing singularities of 3d vector fields with geometric algebra. In *IEEE Visualization Proceedings '03*, 2003.
10. T. Mullin, J. J. Kobine, S. J. Tavener, and K. A. Cliffe. On the creation of stagnation points near straight and sloped walls. *Physics of Fluids*, 12(2), 2000.
11. R. Peikert and M. Roth. The "Parallel Vectors" Operator - a Vector Field Visualization Primitive. In *IEEE Visualization Proceedings '00*, pages 263 – 270, 2000.
12. R. Samtaney, D. Silver, N. Zabusky, and J. Cao. Visualizing features and tracking their evolution. *IEEE Computer*, 27(2):20 – 27, 1994.
13. W. J. Schroeder, R. Volpe, and W. E. Lorensen. The Stream Polygon: A Technique for 3d Vector Field Visualization. In *IEEE Visualization Proceedings*, 1991.
14. D. Silver and X. Wang. Tracking and visualizing turbulent 3d features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2), 1997.
15. D. Sujudi and R. Haimes. Identification of Swirling Flow in 3D Vector Fields. Technical Report AIAA Paper 95–1715, American Institute of Aeronautics and Astronautics, 1995.
16. H. Theisel and H.-P. Seidel. Feature flow fields. In *Joint Eurographics-IEEE TVCG Symposium on Visualization*, 2003.
17. H. Theisel, T. Weinkauff, H.-C. Hege, and H.-P. Seidel. Saddle Connectors - An Approach to Visualizing the Topological Skeleton of Complex 3d Vector Fields. In *IEEE Visualization '03*, 2003.
18. X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology Tracking for the Visualization of Time-Dependent Two-Dimensional Flows. *Computers & Graphics*, 26(2):249 – 257, 2002.
19. C. Weigle and D. C. Banks. Extracting iso-valued features in 4-dimensional scalar fields. In *Proc. Symposium on Volume Visualization*, 1998.