Closed Streamlines in Flow Visualization

Thomas Wischgoll

University of Kaiserslautern
Department of Computer Science, Computer Graphics & CAGD
P.O. Box 3049, D-67653 Kaiserslautern
Germany
E-mail: wischgol@informatik.uni-kl.de

Abstract

Vector fields occur in many of the problems in science and engineering. In combustion processes, for instance, vector fields describe the flow of the gas. This process can be enhanced using vector field visualization techniques. Also, wind tunnel experiments can be analyzed. An example is the design of an air wing. The wing can be optimized to create a smoother flow around it. Vector field visualization methods help the engineer to detect critical features of the flow. Consequently, feature detection methods gained great importance during the last years.

Topological methods are often used to visualize vector fields because they clearly depict the structure of the vector field. In previous publications about topological methods closed streamlines are neglected. Since closed streamlines can behave in exactly the same way as sources and sinks they are an important feature that cannot be ignored anymore.

To accomplish this, this work concentrates on detecting this topological feature. We introduce a new algorithm that finds closed streamlines in vector fields that are given on a grid where the vectors are interpolated linearly. We identify regions that cannot be left by a streamline. According to the Poincare-Bendixson theorem there is a closed streamline in such a region if it does not contain any critical point. Then we identify the exact location using the Poincare map. In contrast to other algorithms, this method does not presume the existence of a closed streamline. Consequently, this algorithm is able to really detect closed streamlines inside the vector field.

In combustion processes closed streamlines in a three dimensional flow are a hint for recirculation zones. These zones describe regions inside the flow where the gas stays quite long. This is necessary for the gas to completely burn. Therefore, we also show how to detect this important feature in three dimensional vector fields.

Keywords: vector field, unsteady flow, closed streamline, limit cycle, bifurcation, topology, flow visualization

1 Introduction

An intuitive and often used method for vector field visualization is the calculation of streamlines. If one uses this technique in turbulent fields, one encounters often the problem of closed streamlines. The difficulty with standard integration methods is that streamlines approaching a closed curve cycle around that curve without ever approaching a critical point or the boundary. Usually, one uses a stopping criteria like elapsed time or number of integration steps to prevent infinite loops. We are interested in the exact location of the limit cycles, so that such a vague criteria does not fit our needs. For this reason we developed an algorithm [39] that uses the underlying grid to check if the same cell is crossed while integrating the streamline: this results in a cycle of cells. In that case, the algorithm determines if the streamline can leave this cell cycle or not. If it does not leave it is proven that there exists a closed streamline inside the cell cycle on condition that there is no singularity inside the involved cells.

Additionally, in time-dependent planar flows it is possible to track the paths of the singularities and draw the separatricies as surfaces that vary when time propagates [35]. When following one particular singularity, for instance a sink, this singularity may switch its type and becomes a source: the so called Hopf bifurcation has occured. Assuming that the global structure of the field has not changed, there emerges or ends a closed streamline in the surrounding of the singularity. Then, the separatrix surface does not reach the singularity anymore but ends at the closed streamline instead.

For a better understanding of this essential topological property of the field we investigate the evolution of closed streamlines over time. There are mainly two different bifurcations involved when the life cycle of a closed streamline is started or terminated: the hopf bifurcation and the blue sky in 2D bifurcation which is a global bifurcation where a saddle gets connected to itself by a streamline.

The evolution of closed streamlines in planar flows is visualized by a third dimension representing time. We can show the evolution using a tube that interpolates the closed streamlines in different timesteps. This technique was inspired by the *The Visual Mathematics Library* by Abraham and Shaw [4] which facilitates a great way to understand dynamics by the use of discerning sketches.

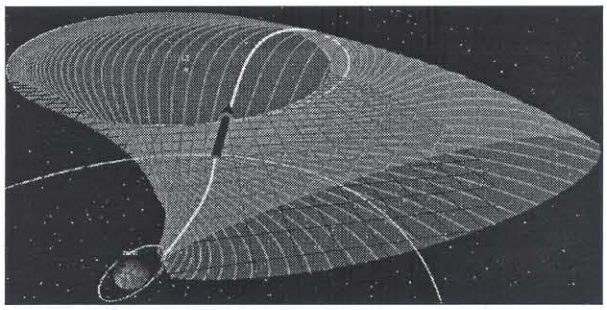

Figure 1: Terrestrial Planet Finder Mission (image courtesy of Ken Museth, Caltech[27]).

In three dimensions closed streamlines also occur. Figure 1 shows an example. There, a stable manifold from the Terrestrial Planet Finder Mission of NASA is shown. The circle in the back shows the location of the closed streamline. To accomplish such situations, we show how to detect closed streamlines in a three dimensional vector field.

In the next section we summarize previous work. Afterwards we give a short introduction to the theoretical background. Sections 4, 5, and 6 describe the algorithms that detects closed streamlines in 2D, over time, and in 3D respectively. Section 7 shows the results of our methods.

2 Related Work

Various methods exist that show different aspects of vector fields. To track a particle in the flow over time streamlines, streaklines, and pathlines[14] [24] are used. A streamline shows the path of a massless particle in the flow. Such a particle follows the trajectory of the dynamical system. A streakline visualizes the path of dye injected for a period of time at a fixed position into a time dependent flow while a pathline only follows a single particle. A particle corresponds to a point moving through the flow. If we use more general objects like lines, circles, or implicit surfaces streamsurfaces, streamribbons, streamtubes, or streamballs are created [6]. Also, an n-sided polygon can be placed perpendicular to the flow and moved along the trajectory [34]. This method additionally depicts local flow attributes, like rotation and shear.

Topological methods depict the structure of the flow by connecting sources, sinks, and saddle singularities with separatrices. Critical points were first investigated by Perry [30][28][29], Dallmann [9], Chong [8] and others. The method itself was first introduced in visualization for two dimensional flows by Helman and Hesselink [17][16][18][19][15]. Several extensions to this method exist. Scheuermann et al. [33] extended the method to work on a bounded region. To get the whole topological skeleton of the vector field, points on the boundary have to be taken into account, also. These points are called boundary saddles. To create a time dependent topology for two dimensional vector fields, Helman and Hesselink [19] use the third coordinate to represent time. This results in surfaces representing the evolution of the separatrices. A similar method is proposed by Tricoche et al. [35][36] but this work focuses on tracking singularities through time. Although closed streamlines can act in the same way as sources or sinks, they are ignored in the considerations of Helman and Hesselink and others.

To extend this method to three dimensional vector fields, Globus et al. [12] presented a software system that is able to extract and visualize some topological aspects of three dimensional vector fields. The various critical points are characterized using the eigenvalues of the Jacobian. This technique was also suggested by Helman and Hesselink [19]. But the whole topology of a three dimensional flow is not yet available. There, streamsurfaces are required to represent separatricies. A few algorithms for computing streamsurfaces exist [22][32] but are not yet integrated in a topological algorithm.

There are some algorithms to find closed streamlines in dynamical systems that can be found in the numerical literature. Aprille and Trick [5] proposed a so called shooting method. There, the fixed point of the Poincaré map is found using a numerical algorithm like Newton-Raphson. Dellnitz et al. [10] detect almost cyclic behavior. It is a stochastical approach where the Frobenius-Perron operator is discretized. This stochastical measure identifies regions where trajectories stay very long. But these mathematical methods typically depend on continuous dynamical systems where a closed form description of the vector field is available. This is usually not the case in visualization and simulation where the data is given on a grid and interpolated inside the cells. Van Veldhuizen [37] uses the Poincaré map to create a series of polygons approximating an attracting closed streamline. The algorithm starts with a rough approximation of the closed streamline. Every vertex is mapped by the Poincaré map iteratively to get a finer approximation. Then, this series converges to the closed streamline.

To get a hierarchical approach for the visualization of invariant sets, and therefore closed streamlines also, Bürkle et al. [7] enclose the invariant set by a set of boxes. They start with a box that surrounds the invariant set completely. This box is successively bisected in cycling directions. It is always ensured that the result still includes the complete invariant set. Using this bisection, an approximation of the invariant set is finally found which can be rendered using a volume renderer. The publication of Guckenheimer [13] gives a detailed overview concerning invariant sets in dynamical systems.

Some publications deal with the analysis of the behavior of dynamical systems. Schematic drawings showing the various kinds of closed streamlines can be found in the books of Abraham and Shaw [3][4]. Fischel et al. [11] presented a case study where they applied different visualization methods to dynamical systems. In their applications also strange attractors, like the Lorentz attractor, and closed streamlines occur. So called sweeps which are trajectories represented as tubes are used. These sweeps allow to introduce a color coding scheme. For instance, the color can help to recognize that a trajectory still slowly moves towards a closed streamline that weakly attracts.

Wegenkittl et al. [38] visualize higher dimensional dynamical systems. To display trajectories parallel coordinates [23] are used. A trajectory is sampled at various points in time. Then these points are displayed in the parallel coordinate system and a surface is extruded to connect these points. As an example, also a chaotic attractor derived from the Lorentz system is visualized. Hepting et al. [20] study invariant tori in four dimensional dynamical systems by using suitable projections into three dimensions to enable detailed visual analysis of the tori. This visualization can help when limits of mathematical analysis are reached to get more insight into the dynamical system.

Löffelmann [25][26] uses Poincaré sections to visualize closed streamlines and strange attractors. Poincaré sections define a discrete dynamical system of lower dimension which is easier to understand. The Poincaré section which is transverse to the closed streamline is visualized as a disk. On the disk, spot noise is used to depict the vector field projected onto that disk. By this method, it can be clearly recognized whether the flow, for instance, spirals around the closed streamline and is attracted or repelled or if it is a rotating saddle. Additionally, streamlines and streamsurfaces show the vector field in the vicinity of the closed streamline that is not located on the disk visualizing the Poincaré section.

3 Theoretical Background

We assume that the vector field is given on a grid consisting of triangles or rectangles and interpolated linearly inside the cells. This is not a limitation since most simluated and measured vector fields are given on such a grid. In the following subsections, we give a short overview of the theoretical background concerning closed streamlines. The reader should be familiar with at least the fundamental theory about vector field topology. More about the theoretical background can be found in several books [21][40][31].

3.1 Limit Sets

The topological analysis of vector fields considers the asymptotic behavior of streamlines. There we have two different kinds of so called limit sets, the origin set or α-limit set of a streamline and the end set or ω-limit set.

Definition 3.1 (α-limit set)
Let s be a streamline in a given vector field v. Then we define the ***α-limit*** *set as the following set:*

$$\{p \in \mathbb{R}^2 | \exists (t_n)_{n=0}^{\infty} \subset \mathbb{R}, t_n \to -\infty, \lim_{n \to \infty} s(t_n) \to p\}$$

Definition 3.2 (ω-limit set)
Let s be a streamline in a given vector field v. Then we define the ***ω-limit*** *set as the following set:*

$$\{p \in \mathbb{R}^2 | \exists (t_n)_{n=0}^{\infty} \subset \mathbb{R}, t_n \to \infty, \lim_{n \to \infty} s(t_n) \to p\}$$

Remark 3.3
Let v be a vector field. We speak of an α- or ω-limit set L of v if there exists a streamline s in the vector field v that has L as α- or ω-limit set.

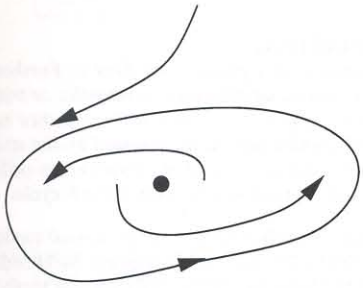

Figure 2: Example for α- and ω-limit sets.

If the α- or ω-limit set of a streamline consists of only one point, this point is a critical point. The most common case of a α- or ω-limit set in a planar vector field containing more than one inner point of the domain is a closed streamline which is declared in the next definition. Figure 2 shows an example for α- and ω-limit sets. Here we have a critical point and a closed streamline. The critical point and the closed streamline are their own α- and ω-limit set. For every other streamline the closed streamline is the ω-limit set. If the streamline starts inside the closed streamline the critical point is the α-limit set. Otherwise the α-limit set is empty. Now that we showed an example for a closed streamline let us give a precise definition.

Definition 3.4 (Closed streamline)
Let v be a vector field. A ***closed streamline*** $\gamma : \mathbb{R} \to \mathbb{R}^n$, $t \mapsto \gamma(t)$ *is a streamline of a vector field v such that there is a $t_0 \in \mathbb{R}$ with*

$$\gamma(t + nt_0) = \gamma(t) \quad \forall n \in \mathbb{N}$$

and γ not constant.

Remark 3.5
There are several different terms describing a closed streamline. The terms ***limit cycle***, ***closed orbit***, *and closed streamline are equivalent.*

Similar to critical points we define *asymptotically stability* of closed streamlines. If a closed streamline is asymptotically stable it is attracting.

Definition 3.6 (Asymptotically stability of closed streamlines)
Let $v : W \to \mathbb{R}^n$ be a vector field that is continuously differentiable. Let further ϕ be the corresponding dynamical system and $\gamma \subset W$ a closed streamline. If for every neighborhood $U \subset W$ with $\gamma \subset U$ there is a neighborhood $U_1 \subset U$ with $\gamma \subset U_1$ such that $\phi_t(x) \in U$ for all $x \in U_1$ and $t > 0$ and

$$\lim_{t \to \infty} \min\{\|\phi_t(x) - z\| | z \in \gamma\} = 0$$

then γ is called ***asymptotically stable*** *closed streamline.*

This means that an asymptotically stable closed streamline attracts the flow inside a special neighborhood. It also follows from this definition that an asymptotically stable closed streamline is isolated from other closed orbits. In the same way there are closed streamlines that are repelling. For instance, by inverting the vector field we can turn an attracting closed streamline into a repelling one.

3.2 Poincaré Map

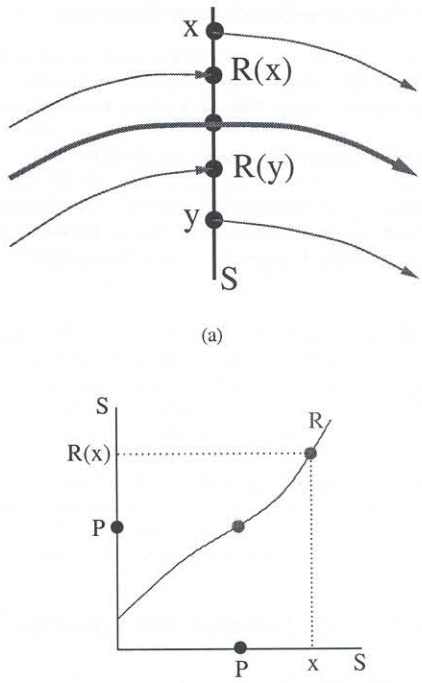

Figure 3: Poincaré section (a) and Poincaré map (b).

Let us assume we have a two dimensional vector field containing one limit cycle. Then we can choose a point P on the limit cycle and draw a *cross section* S which is a line segment not parallel to the limit cycle across the vector field. This line is called a *Poincaré section*. If we start a streamline at an arbitrary point x on S and follow it until we cross the Poincaré section S again, we get another point $R(x)$ on S. This results in the *Poincaré map* R. Figure 3 illustrates the situation. Part (a) shows the Poincaré section with the limit cycle in the middle drawn with a thicker line, while the part (b) displays the Poincaré map itself. Obviously the point P on the limit cycle is mapped onto itself. Consequently, it is a fixed point of the Poincaré map.

Let us precise this in some definitions:

Definition 3.7 (Cross section)
Let v be a vector field and $S \subset \mathbb{R}^n$ an open set on a hyperplane of dimension $n-1$ that is transverse to v. Transverse to v means that $v(x) \notin S$ for all $x \in S$. Then S is called a ***cross section.***

Definition 3.8 (Poincarè map)
Let v be a vector field and ϕ the dynamical system belonging to v. Let further be S a cross section that intersects a closed streamline at a point P. Then the ***Poincarè map*** *is defined as the map $R : S \to S$ such that*

$$x \mapsto \phi_t(x) \quad ,$$

where t is the time the streamline started at x needs to intersect the cross section again after one turn.

Remark 3.9
It is obvious that the point P on the closed streamline is a fixed point of the Poincarè map.

3.3 The Poincaré-Bendixson Theorem

In this subsection we show a fundamental result which makes it easier to find closed streamlines in a two dimensional vector field. This property is exploited by our algorithm which is introduced later.

Theorem 3.10 (Poincarè-Bendixson Theorem)
Let $W \subset \mathbb{R}^2$ be an open subset and $v : W \to \mathbb{R}^2$ a two dimensional, continuously differentiable vector field. Let further $L \subset W$ be a nonempty compact limit set of the vector field v that contains no critical point. Then L describes a closed streamline.

Corollary 3.11
Let $W \subset \mathbb{R}^2$ be an open subset and $v : W \to \mathbb{R}^2$ a two dimensional, continuously differentiable vector field. Let further $D \subset W$ be a nonempty compact subset which contains no critical point and s a streamline inside D. If the streamline s does not leave D then there exists a closed streamline inside D.

Using this corollary our algorithm to detect closed streamlines can simply integrate a streamline and check during the integration process if it runs into a compact region that is never left. If we find such a region this corollary states that we found a closed streamline.

4 Detection of Closed Streamlines in 2D

In a precomputational step every singularity of the vector field is determined. To find all stable closed streamlines we mainly compute the topological skeleton of the vector field. We use an ordinary streamline integrator, like for instance an ODE solver using Runge-Kutta. But we extended this streamline integrator so that it is able to detect closed streamlines. In order to find all closed streamlines that reside inside another closed streamline we have to continue integration after we found a closed streamline inside that region.

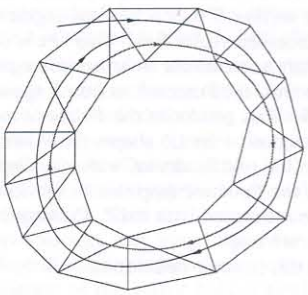

Figure 4: A streamline approaching a limit cycle has to reenter cells.

4.1 Theory

The basic idea of our streamline integrator is to determine a region of the vector field that is never left by the streamline. According to the Poincaré-Bendixson-Theorem, a streamline approaches a closed streamline if no singularity exists in that region.

Notation 4.1 (Actually investigated streamline)
*We use the term **actually investigated streamline** to describe the streamline that we check if it runs into a limit cycle.*

To reduce computational cost we first integrate the streamline using a Runge-Kutta-method of fifth order with an adaptive stepsize control. Every cell that is crossed by the streamline is stored during the computation. If a streamline approaches a limit cycle it has to reenter the same cell again as shown in figure 4. This results in a *cell cycle*:

Definition 4.2 (Cell cycle)
*Let s be a streamline in a given vector field v. Further, let G be a set of cells representing an arbitrary rectangular or triangular grid without any holes. Let $C \subset G$ be a finite sequence $c_0, \ldots, c_n$ of neighboring cells where each cell is crossed by the streamline s in exactly that order and $c_0 = c_n$. If s crosses every cell in C in this order again while continuing, C is called a **cell cycle**.*

This cell cycle identifies the region mentioned earlier. To check if this region can be left we could integrate backwards starting at every point on the boundary of the cell cycle. If there is one point converging to the actually investigated streamline we know for sure that the streamline will leave the cell cycle. If not, the actually investigated streamline will never leave the cell cycle. Since there are infinitely many points on the boundary this, of course, results in a non-terminating algorithm. To crack this problem we have to reduce the number of points we have to check. Therefore we define *potential exit points*:

Definition 4.3 (Potential exit points)
*Let C be a cell cycle in a given grid G as in Definition 4.2. Then there are two kinds of **potential exit points**. First, every vertex of the cell cycle C is a **potential exit point**. Second, every point on an edge at the boundary of C where the vector field is tangential to the edge is also a **potential exit point**. Here, only edges that are part of the boundary of the cell cycle are considered. Additionally, only the **potential exit points** in the spiraling direction of the streamline need to be taken into account.*

To determine if the streamline leaves the cell cycle we start a backward integrated streamline to see where we have to enter the cell cycle in order to leave it at that exit. We will show later that it is sufficient to only check these potential exit points if we want to figure out if the streamline can leave the cell cycle.

Notation 4.4 (Backward integrated streamline)
*We use the term **backward integrated streamline** for the streamline we integrate by inverting the vectors of the vector field starting at a potential exit point in order to validate this exit point.*

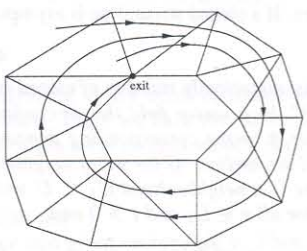

Figure 5: If a real exit point can be reached, the streamline will leave the cell cycle.

Definition 4.5 (Real exit points)
*Let P be a potential exit point of a given cell cycle C as in definition 4.3. If the backward integrated streamline starting at P does not leave the cell cycle after one full turn through the cell cycle, the potential exit point is called a **real exit point**.*

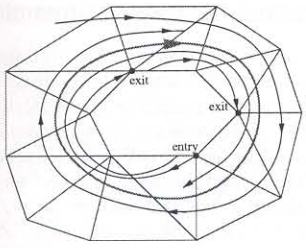

Figure 6: If no real exit point can be reached, the streamline will approach a limit cycle.

Since a streamline cannot cross itself the backward integration starting at a real exit point converges to the actually investigated streamline. Consequently, the actually investigated streamline leaves the cell cycle near that real exit point. Figure 5 shows such a real exit point.

If on the other hand no real exit point exists we can determine for every potential exit point where we have a region with an inflow that leaves at that potential exit. Consequently, the actually investigated streamline cannot leave near that potential exit point.

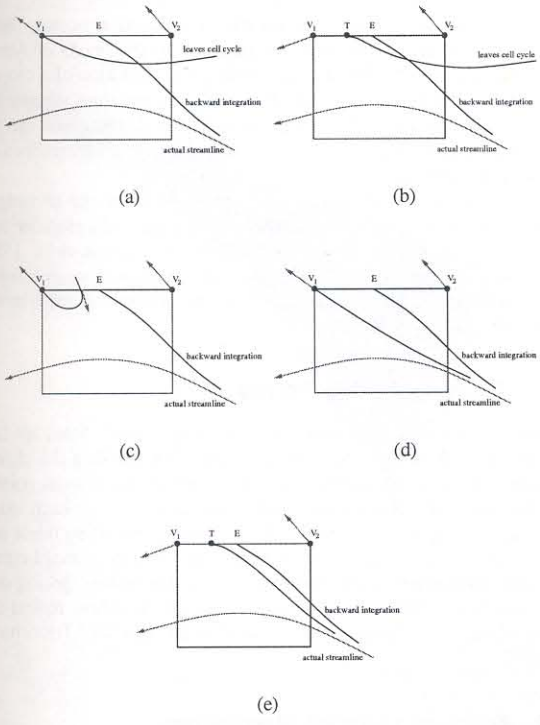

Figure 7: Different cases of potential exits. (a) and (b) is impossible because streamlines cannot cross each other, (c) contradicts with the linear interpolation on an edge, in (d) and (e) both backward integrations converge to the actual streamline so that the point E is a real exit.

With these definitions we can formulate the main theorem for our algorithm:

Theorem 4.6
Let C be a cell cycle with no singularity inside and E the set of potential exit points. If there is no real exit point among the potential exit points E or there are no potential exit points at all then there exists a closed streamline inside the cell cycle.

Proof: *(Sketch)*
Let C be the cell cycle. It is obvious that we cannot leave the cell cycle C if all backward integrated streamlines started at every point on the boundary of C leave the cell cycle C. According to the Poincaré-Bendixson-theorem, there exists a closed streamline inside the cell cycle in that case.

We will show now that it is sufficient to treat only the potential exit points. If the backward integrated streamlines starting at all these potential exit points leave the cell cycle the backward integration of any point on an edge will also do.

Figure 7 shows the different configurations of potential exits. Let E be an arbitrary point on an edge between two potential exit points. In part *(a)* both backward integrated streamlines starting at the vertices V_1 and V_2 leave the cell cycle. Consequently, E cannot be an exit. It would need to cross one of the other backward integrated streamlines which is impossible.

Part *(b)* of figure 7 shows the case where the vector at a point on the edge is tangential to the edge. Obviously, if E lies between V_1 and T the backward integrated streamline will leave the cell cycle immediately. If it lies between T and V_2 and converges to the actually investigated streamline it has to cross the backward integrated streamline started at T. This is not possible. Because of the linear interpolation at the edge, part *(c)* is also impossible.

We have shown that the actually investigated streamline cannot leave the cell cycle. Consequently, there exists a closed streamline inside the cell cycle C since there is no singularity inside C. ○

Remark 4.7
To get a possible configuration the backward integration starting at the vertex V_1 must also converge to the streamline because it cannot cross the backward integration starting at point E as in part (d) *of figure 7. Part* (e) *explains why we also need to investigate the tangential case. If we start a backward integrated streamline at point E it converges towards the actually investigated streamline. But if we only consider the vertices of the edge, both exit points may be no real exit points. Therefore we also have to start a backward integrated streamline at the point T, where the vector field is tangential to the edge, to figure out that we leave the cell cycle at this edge. On the other hand, a backward integrated streamline starting at any point between V_1 and T immediately leaves the cell cycle due to the linear interpolation.*

4.2 Algorithm

With theorem 4.6 we are able to describe our algorithm in detail. It mainly consists of the same three different states:

- streamline integration: identifying one cell change after the other, check at each cell if we complete a cell cycle.
- checking for exits: going backwards through the crossed cells and looking for potential exit points.
- validating exit: integrating backwards a curve from potential exit through the whole cell cycle.

We first use a standard integration method to compute the streamline, first. In this step we only check for cell cycles. This saves computational time since the checking of all the exits is rather expensive. If we detect a cell cycle we have to find all potential exit points. After that we need to validate each of the potential exit

points to figure out if there is a real exit point among them. If this is the case we did not run into a closed streamline yet. Therefore we continue with the standard integration. The algorithm exits if we could not find a real exit point among all the potential exit points or if we reached a critical point or the boundary of the vector field.

Remark 4.8
Theorem 4.6 guarantees that our algorithm detects closed streamlines if we check every potential exit point.

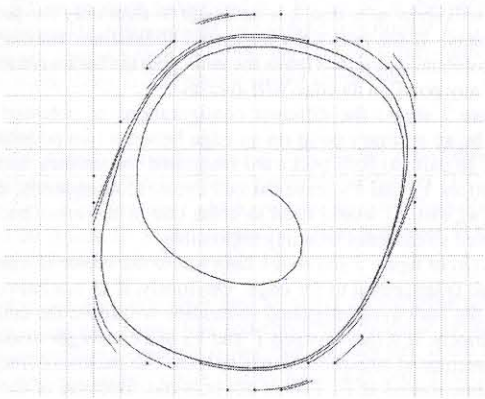

Figure 8: Exits of a cell cycle.

Figure 8 shows a real example of our algorithm. There we start a streamline near the source in the center of the figure. This streamline spirals until we find the first cell cycle. We stopped the integration there for this example. The figure also shows all exits and its backward integrated streamlines. In this example, every potential exit point is shown. We can see that potential exit points which are passed by a backward integrated streamline do not necessarily need to be investigated because if the backward integrated streamline leaves the cell cycle the other one will also do. Figure 9 shows this in detail. There the backward integrated streamline starting at *Exit 2* also has to leave the cell cycle because it cannot cross the backward integrated streamline starting at *Exit 1*. In the other case, where the backward integrated streamline started at *Exit 1* stays inside the cell cycle, we have to continue the actually investigated streamline, anyway.

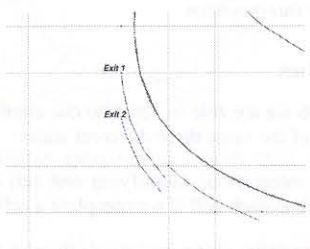

Figure 9: Exit of the cell cycle which does not need to be investigated.

Since the streamline spirals from the inner region to the outside, we only have to consider the potential exits in that direction. In the example, every backward integrated streamline leaves the cell cycle. Consequently, there is a limit cycle in this cell cycle which can be localized as described in the next section.

4.3 Exact Location of Closed Streamlines

Since we know a region that is never left by the streamline we can find the exact position of the closed streamline using the Poincaré map. This map is described in detail in the subsection 3.2.

To find the exact position of the closed streamline we can use the edge where we detected the cell cycle as a Poincaré section. Then we only have to find the fixed point of the Poincaré map. We use a binary search to find this fixed point: we divide the edge where we detected the cell cycle into two parts. At the mid point we start a streamline to see which part of the edge is intersected by the streamline after one full turn. Since the streamline cannot leave the cell cycle, it is guaranteed that the streamline intersects one part of the edge. Then, this part is subdivided again and we start another streamline at the mid point. This process continues until we are close enough to the fixed point of the Poincaré map. We use the length of the part of the edge as a stopping criterion.

This fixed point gives us a point lying on the closed streamline. If we start another streamline at that point this streamline will follow the closed streamline we are looking for. After one turn, i.e. after reaching the start point again, we know the exact location of the closed streamline.

5 Closed Streamlines in Time-Dependant Flows

When dealing with closed streamlines one question occurs: how does a closed streamline emerge? Inspired by the books of Abraham and Shaw[1] [2] [3] [4] we visualize the evolution of a closed streamline in a planar unsteady flow. We use the third dimension to represent the time. The evolution of a closed streamline can be shown as a tube shaped visulization for the closed streamlines in the various timesteps.

The singularities are used as a starting point for our investigations. Therefore we briefly describe the tracking of the singularities in the next section. This work was done by Tricoche et al. [35]. Then we show how to find and follow a closed streamline over time. In the end we explain the results of our algorithm and explain the limitations of our method.

5.1 Tracking Critical Points

When dealing with time-dependent two-dimensional flows we can use the third dimension to represent time. For tracking the closed streamlines we first determine the behavior of the critical points. For a given cell, the associated interpolant contains, for each value of time t, a single critical point. This is due to the affine linear nature inside the triangles of its restriction to any time plane. Letting the time parameter t move from t_i to t_{i+1}, the critical point position describes a 3D curve. A detailed description of how to find the paths of the critical points can be found in the article of Tricoche et al. [35].

5.2 Following Closed Streamlines

After tracking the singularities, we analyze the vector field in discrete timesteps. There must be a critical point inside each closed streamline. Therefore, we use the critical point path containing a Hopf bifurcation as a starting point for our streamline algorithm from section 4 which detects the closed streamline if it exists. We follow the critical point path in discrete steps in positive and negative directions starting at the bifurcation. After we have found the cell cycle containing the closed streamline we find the exact position using the Poincaré-map from subsection 4.3. As a last test we have to check if the closed streamline really surrounds the critical

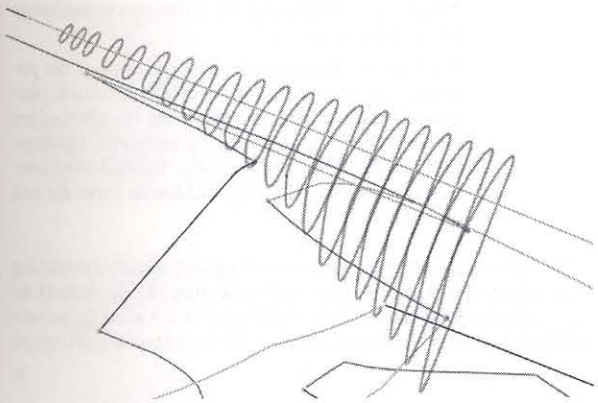

Figure 10: Closed streamlines found by the algorithm.

point. This is necessary because the streamline may have ran into another closed streamline in a totally different region of the flow. Obviously, closed streamlines surrounding the critical point occur only in one of the two temporal directions. We continue by stepping forward in the temporal direction until the closed streamlines reach either another bifurcation which breaks them up or the border of the grid.

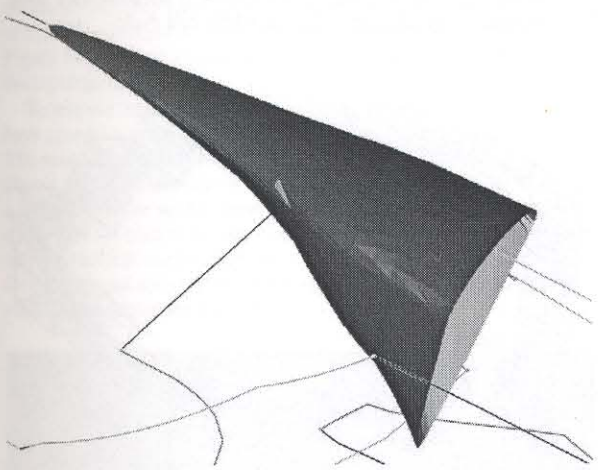

Figure 11: Closed streamlines visualized as a tube over time.

Figure 10 shows the result of this step. Here we have found the closed streamlines at various timesteps. The closed streamlines are approximated by several line segments. The paths of the critical points are also shown using the same colors as in the original paper [35]. The Hopf bifurcation, where we started to detect the closed streamlines, is marked with a yellow sphere. In this example the life cycle of the closed streamline is started by a Hopf bifurcation and terminated by a Periodic Blue Sky in 2D bifurcation.

To visualize the evolution of closed streamlines, we construct tubes from the various closed streamlines similar to the pictures by Abraham and Shaw [4]. We construct surfaces consisting of triangles which connect the approximating line segments of the closed streamlines. The bifurcation point is connected to the tube using a parabolic surface approximated with triangles. The result is shown in figure 11.

6 Detecting Closed Streamlines in 3D

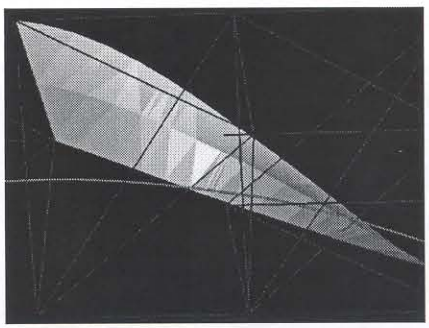

Figure 12: Backward integrated surface.

Although the principle to detect closed streamlines in a three dimensional vector field is similar to the two dimensional case there are some differences. We will describe the theoretical and algorithmical differences and similarities in the next two subsections.

6.1 Theory

We assume that the data is given on a tetrahedral grid. But the principle should work on other cell types as well. The detection of a cell cycle works the same as in definition 4.2. Of course, the cells are three dimensional in this case. To check if we can leave the cell cycle we have to consider every backward integrated streamline starting at an arbitrary point on a face of the boundary of the cell cycle. Looking at the edges of a face we can see directly that it is not sufficient to just integrate streamlines backwards. Figure 12 shows an example. We integrated a streamsurface backwards starting at an edge of the cell cycle. The streamlines starting at the vertices of that edge leave the cell cycle earlier than the complete surface. So it may be possible that a part of the streamsurface stays inside the cell cycle although the backward integrated streamlines starting at the vertices leave it. Consequently, we have to find another definition for exits.

Definition 6.1 (Potential Exit Edges)
Let C be a cell cycle in a given tetrahedral grid G as in Definition 4.2. Then we call every edge at the boundary of the cell cycle a ***potential exit edge****. Analogue to the two dimensional case we define a line on a boundary face where the vector field is tangential to the face as a* ***potential exit edge*** *also.*

Due to the fact that we use linear interpolation inside the tetrahedrons it can show that there will be at least a straight line on the face where the vector field is tangential to the face or the whole face is tangential to the vector field. Therefore, we do not need to consider any isolated point on a face where the vector field is tangential to the face because this cannot occur.

When dealing with edges as exits we have to compute a streamsurface instead of streamlines to consider every point on an exit edge. This leads us to the following notation.

Notation 6.2 (Backward integrated streamsurface)
We use the term ***backward integrated streamsurface*** *to describe the streamsurface we integrate by inverting the vectors of the vector field starting at a potential exit edge in order to validate this exit edge.*

Analogue to definition 4.5 we define *real exit edges.*

Definition 6.3 (Real exit edge)
Let E be a potential exit edge of a given cell cycle C as in definition 6.1. If the backward integrated streamsurface does not completely *leave the cell cycle after one full turn through C then this edge is called a* ***real exit edge****.*

For the backward integrated streamsurface we use a simplified version of the streamsurface algorithm introduced by Hultquist [22]. Since we do not need a triangulation of the surface we only have to process the integration step of that algorithm. Initially, we start the backward integration at the vertices of the edge. If the distance between these two backward integrations is greater than a special error limit we start a new backward integration in between. This continues with the two neighboring integration processes until we have created an approximation of the streamsurface that respects the given error limit.

The integration stops if the whole streamsurface leaves the cell cycle or if we have completed one full turn through the cell cycle. But to construct the surface properly we may have to continue a backward integration process across the boundary of the cell cycle. This is due to the fact that some part of the streamsurface is still inside the cell but the backward integrated streamline has already left it.

With these definitions and motivations we can formulate the main theorem for our algorithm:

Theorem 6.4
Let C be a cell cycle as in definition 4.2 with no singularity inside and E the set of potential exit edges. If there is no real exit edge among the potential exit edges E or there are no potential exit edges at all then there exists a closed streamline inside the cell cycle.

Proof: *(Sketch)*
Let C be a cell cycle with no real exit edges. Every backward integrated streamsurface leaves the cell cycle C completely. As in the 2D case it is obvious that we cannot leave the cell cycle if every backward integration starting at an arbitrary point on a face of the boundary of the cell cycle C leaves the cell cycle. So we have to prove that the actually integrated streamline cannot leave the cell cycle C.

We look at each face of the boundary of the cell cycle C. Let Q be an arbitrary point on a face F of the boundary of the cell cycle C. Let us assume that the backward integrated streamline starting at Q converges to the actually investigated streamline. We have to show that this is a contradiction.

First case: The edges of face F are exit edges and there is no point on F where the vector field is tangential to F.

From a topological point of view the streamsurfaces starting at all edges of F build a tube and leave the cell cycle. Since the backward integrated streamline starting at Q converges to the actually investigated streamline it does not leave the cell cycle. Consequently, it has to cross the tube built by the streamsurfaces. Because streamlines cannot cross each other a streamline cannot cross a streamsurface.

Second case: There is a potential exit edge e on the face F that is not a part of the boundary of F.

Obviously, the potential exit edge e divides the face F into two parts. In one part there is outflow out of the cell cycle C while at the other part there is inflow into C. We do not need to consider the part with outflow any further because every backward integrated streamline starting at a point of that part immediately leaves the cell cycle C.

The backward integrated surface starting at the potential exit edge e and parts of the backward integrated streamsurfaces starting at the boundary edges of the face F build a tube again from a topological point of view. Consequently, the backward integrated streamline starting at Q has to leave the cell cycle C.

We have shown that the backward integrated streamline starting at the point Q has to leave the cell cycle also. Since there is no backward integrated streamline converging to the actually investigated streamline at all, the streamline will never leave the cell cycle.

o

6.2 Algorithm

With theorem 6.4 we are able to describe our algorithm in detail. It is quite similar to the two dimensional case and mainly consists of three different states:

- streamline integration: identifying one cell change after the other, check at each cell if we reached a cell cycle.
- checking for exits: going backwards through the crossed cells and looking for potential exit edges.
- validating exit: integrating backwards a streamsurface from potential exit edges through the whole cell cycle.

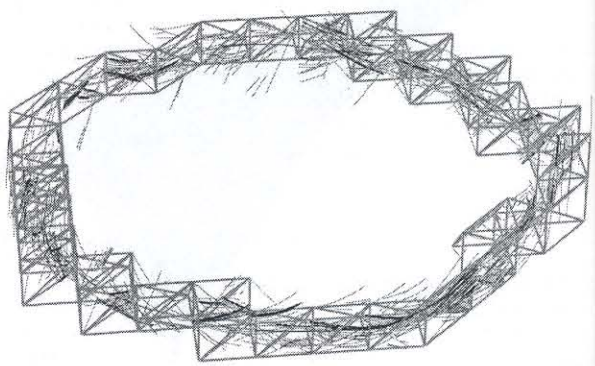

Figure 13: Closed streamline including cell cycle and backward integrations.

Figure 13 shows an example of our backward integration step. There, also the closed streamline and the cell cycle are shown. Every backward integrated streamsurface leaves the cell cycle. According to theorem 6.4, there exists a closed streamline inside this cell cycle. Then we can find the exact location by continuing the integration process of the streamline that we actually investigate until the difference between two successive turns is small enough. This numerical criterion is sufficient in this case since we have shown that the streamline will never leave the cell cycle.

7 Results

The first example is a simulation of a swirling jet with an inflow into a steady medium. The simulation originally resulted in a three

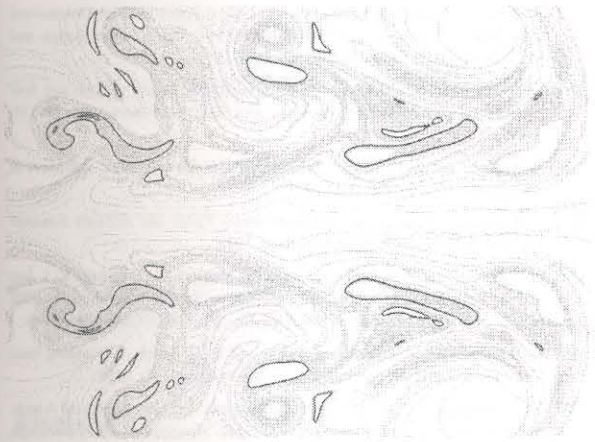

Figure 14: Vorticity vector field visualized by the topological skeleton including closed streamlines.

dimensional vector field but we used a cutting plane and projected the vectors onto this plane to get a two dimensional field. This dataset was provided by Prof. Kollmann from the mechanical engineering department at the University of California at Davis. In this application one is interested in investigating the turbulence of the vector field and in regions where the fluid stays very long. This is necessary because some chemical reactions need a special amount of time. These regions can be located by finding closed streamlines. Figure 14 shows all closed streamlines of this vector field including the topological skeleton.

To test our method that detects closed streamlines over time, we have created a synthetic vector field containing four critical points. The position of the critical points is a function of time, describing closed curves in the plane. We have sampled this vector field on a triangular point set for several values of the time parameter. The rotation of the critical points (each with a specific frequency) entails many structural changes for the topology. This is very interesting for our purpose since all different types of bifurcations which create closed streamlines are present.

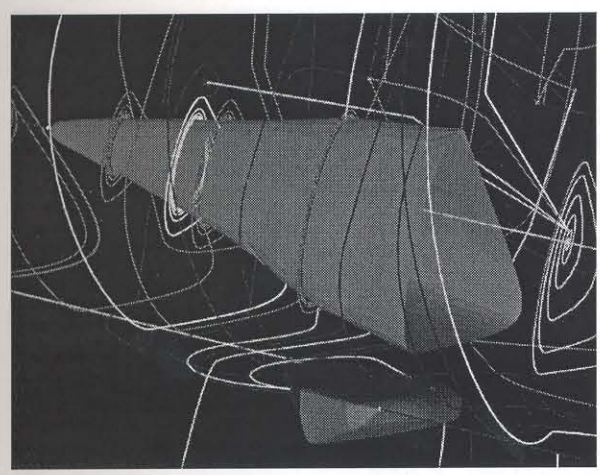

Figure 15: Detailed view of a Periodic Blue Sky in 2D bifurcation.

Figure 15 show a detailed view of the different bifurcations. Also some streamlines are drawn to show how these streamlines circle around the limit cycle but never cross it. The closed streamline is started by a Hopf bifurcation located in the upper left corner. It grows in size until it is terminated by a Periodic Blue Sky in 2D bifurcation. Consequently, the tube visualizing the evolution of the closed streamline does not get closed.

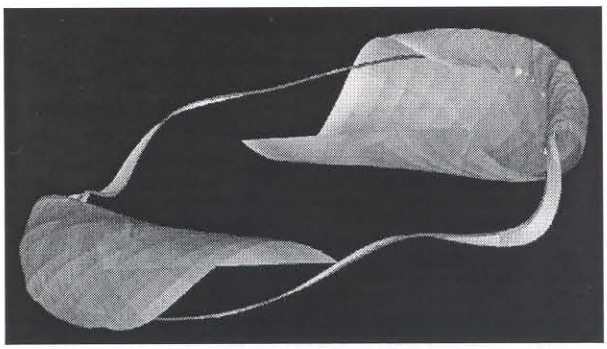

Figure 16: Limit cycle in a 3D vector field with streamsurfaces.

For the last example we created a three dimensional synthetic dataset which includes one closed streamline. We first produced a two dimensional vector field. The vector field contains a saddle singularity in the center and two symmetrical sinks. To get a three dimensional flow we rotated the two dimensional vector field around the y-axis. Due to the symmetrical arrangement of the sinks this vector field includes exactly one closed streamline.

Figure 16 shows the closed streamline found by the algorithm together with two streamsurfaces. The streamsurfaces are attracted by the closed streamline. The streamsurface gets smaller and smaller while it spirals around the closed streamline. After a few turns around the closed streamline it is only slightly wider then a streamline and finally it totally merges with the closed streamline. We used a rather arbitrary color scheme for the surface to enhance the three dimensional impression.

8 Acknowledgement

This research was supported by the DFG project "Visualisierung nicht-linearer Vektorfeldtopologie". Further, we like to thank Tom Bobach, Holger Burbach, Stefan Clauss, Jan Frey, Christoph Garth, Aragorn Rockstroh, René Schätzl and Xavier Tricoche for their programming efforts. The continuous support of all members of the computer graphics and visualization team in Kaiserslautern gives us a nice working environment. Wolfgang Kollmann, MAE Department of the University of California at Davis, provided us with the 2D dataset.

References

[1] R. H. Abraham and C. D. Shaw. *Dynamics – The Geometry of Behaviour: Bifurcation Behaviour Part 1: Periodic Behaviour.* Aerial Press, Inc., Santa Cruz, 1982.

[2] R. H. Abraham and C. D. Shaw. *Dynamics – The Geometry of Behaviour: Bifurcation Behaviour Part 2: Chaotic Behaviour.* Aerial Press, Inc., Santa Cruz, 1983.

[3] R. H. Abraham and C. D. Shaw. *Dynamics – The Geometry of Behaviour: Bifurcation Behaviour Part 3: Global Behaviour.* Aerial Press, Inc., Santa Cruz, 1984.

[4] R. H. Abraham and C. D. Shaw. *Dynamics – The Geometry of Behaviour: Bifurcation Behaviour Part 4: Bifurcation Behaviour.* Aerial Press, Inc., Santa Cruz, 1988.

[5] T. J. Aprille and T. N. Trick. A computer algorithm to determine the steady-state response of nonlinear oscillators. *IEEE Transactions on Circuit Theory*, CT-19(4), July 1972.

[6] M. Brill, W. Djatschin, H. Hagen, S. V. Klimenko, and H.-C. Rodrian. Streamball techniques for flow visualization. In *Proceedings Visualization '94*, pp. 225–231, October 1994.

[7] D. Bürkle, M. Dellnitz, O. Junge, M. Rumpf, and M. Spielberg. Visualizing complicated dynamics. In A. Varshney, C. M. Wittenbrink, and H. Hagen, editors, *IEEE Visualization '99 Late Breaking Hot Topics*, pp. 33 – 36, San Francisco, 1999.

[8] M. S. Chong, A. E. Perry, and B. J. Cantwell. A General Classification of Three-Dimensional Flow Fields. *Physics of Fluids*, A2(5), pp. 765–777, 1990.

[9] U. Dallmann. Topological Structures of Three-Dimensional Flow Separations. Technical Report DFVLR-AVA Bericht Nr. 221-82 A 07, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt e.V., April 1983.

[10] M. Dellnitz and O. Junge. On the Approximation of Complicated Dynamical Behavior. *SIAM Journal on Numerical Analysis*, 36(2), pp. 491 – 515, 1999.

[11] G. Fischel, H. Doleisch, L. Mroz, H. Löffelmann, and E. Gröller. Case study: visualizing various properties of dynamical systems. In *Proceedings of the Sixth International Workshop on Digital Image Processing and Computer Graphics (SPIE DIP-97)*, pp. 146–154, Vienna, Austria, October 1997.

[12] A. Globus, C. Levit, and T. Lasinski. A Tool for Visualizing the Topology of Three-Dimensional Vector Fields. In G. M. Nielson and L. Rosenblum, editors, *IEEE Visualization '91*, pp. 33 – 40, San Diego, 1991.

[13] J. Guckenheimer. Numerical analysis of dynamical systems, 2000.

[14] C. Hansen. Visualization of vector fields (2D and 3D). In *SIGGRAPH '93 Course Notes no. 2, Introduction to Scientific Visualization Tools and Techniques*, August 1993.

[15] J. L. Helman. *Representation and Visualization of Vector Field Topology*. PhD thesis, Stanford University, 1997.

[16] J. L. Helman and L. Hesselink. Automated analysis of fluid flow topology. In *Three-Dimensional Visualization and Display Techniques, SPIE Proceedings Vol. 1083*, pp. 144–152, 1989.

[17] J. L. Helman and L. Hesselink. Representation and Display of Vector Field Topology in Fluid Flow Data Sets. *Computer*, 22(8), pp. 27–36, 1989.

[18] J. L. Helman and L. Hesselink. Surface Representations of Two- and Three-Dimensional Fluid Flow Topology. In G. M. Nielson and B. Shriver, editors, *Visualization in scientific computing*, pp. 6–13, Los Alamitos, CA, 1990.

[19] J. L. Helman and L. Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3), pp. 36–46, May 1991.

[20] D. H. Hepting, G. Derks, D. Edoh, and R. D. Russel. Qualitative analysis of invariant tori in a dynamical system. In G. M. Nielson and D. Silver, editors, *IEEE Visualization '95*, pp. 342 – 345, Atlanta, GA, 1995.

[21] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems and Linear Algebra*. Academic Press, New York, 1974.

[22] J. P. M. Hultquist. Constructing Stream Surface in Steady 3D Vector Fields. In *Proceedings IEEE Visualization '92*, pp. 171–177. IEEE Computer Society Press, Los Alamitos CA, 1992.

[23] A. Inselberg and B. Dimsdale. Parallel Coordinates: a Tool for Visualizing Multidimensional Geometry. In *IEEE Visualization '90 Proceedings*, pp. 361–378, Los Alamitos, 1990. IEEE Computer Society.

[24] D. A. Lane. UFAT – A Particle Tracer for Time-Dependent Flow Fields. In *Proceedings IEEE Visualization 1994*, pp. 257–264. IEEE Computer Society Press, Los Alamitos CA, 1994.

[25] H. Löffelmann. *Visualizing Local Properties and Characteristic Structures of Dynamical Systems*. PhD thesis, Technische Universität Wien, 1998.

[26] H. Löffelmann, T. Kučera, and E. Gröller. Visualizing Poincaré Maps Together with the Underlying Flow. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization, Algorithms, Applications, and Numerics*, pp. 315–328. Springer, 1997.

[27] K. Museth, A. Barr, and M. W. Lo. Semi-Immersive Space Mission Design and Visualization: Case Study of the "Terrestrial Planet Finder" Mission. In *Proceedings IEEE Visualization 2001*, pp. 501–504. IEEE Computer Society Press, Los Alamitos CA, 2001.

[28] A. E. Perry. A Study of Non-degenerate Critical Points in Three-Dimensional Flow Fields. Technical Report DFVLR-FB 84-36, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt e.V., 1984.

[29] A. E. Perry and M. S. Chong. A description of eddying motions and flow patterns using critical point concepts. *Ann. Rev. Fluid Mech.*, pp. 127–155, 1987.

[30] A. E. Perry and B. D. Fairly. Critical Points in Flow Patterns. *Advances in Geophysics*, 18B, pp. 299–315, 1974.

[31] R. Roussarie. *Bifurcations of Planar Vector Fields and Hilbert's Sixteenth Problem*. Birkhäuser, Basel, Switzerland, 1998.

[32] G. Scheuermann, T. Bobach, H. Hagen, K. Mahrous, B. Hahmann, K. I. Joy, and W. Kollmann. A Tetrahedra-Based Stream Surface Algorithm. In *IEEE Visualization '01 Proceedings*, Los Alamitos, 2001. IEEE Computer Society.

[33] G. Scheuermann, B. Hamann, K. I. Joy, and W. Kollmann. Visualizing local Vetor Field Topology. *Journal of Electronic Imaging*, 9(4), 2000.

[34] W. J. Schroeder, R. C. Volpe, and W. E. Lorensen. The stream polygon: A technique for 3D vector field visualization. In *IEEE Visualization '91 Proceedings*, pp. 123–132. IEEE Computer Society, October 1991.

[35] X. Tricoche, G. Scheuermann, and H. Hagen. Topology-Based Visualization of Time-Dependent 2D Vector Fields. In R. P. D. Ebert, J. M. Favre, editor, *Proceedings of the Joint Eurographics–IEEE TCVG Symposium on Visualization*, pp. 117–126, Ascona, Switzerland, 2001. Springer.

[36] X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology Tracking for the Visualization of Time-Dependent Two-Dimensional Flows. *Computer & Graphics*, 26, pp. 249–257, 2002.

[37] M. van Veldhuizen. A New Algorithm for the Numerical Approximation of an Invariant Curve. *SIAM Journal on Scientific and Statistical Computing*, 8(6), pp. 951 – 962, 1987.

[38] R. Wegenkittl, H. Löffelmann, and E. Gröller. Visualizing the Behavior of Higher Dimensional Dynamical Systems. In R. Yagel and H. Hagen, editors, *IEEE Visualization '97 Proceedings*, pp. 119 – 125, Phoenix, AZ, 1997.

[39] T. Wischgoll and G. Scheuermann. Detection and Visualization of Closed Streamlines in Planar Flows. *IEEE Transactions on Visualization and Computer Graphics*, 7(2), 2001.

[40] Y. Yan-qian, C. Sui-lin, C. Lan-sun, H. Ke-cheng, L. Ding-jun, M. Zhi-en, W. Er-nian, W. Ming-shu, and Y. Xin-an. *Theory of Limit Cycles*. American Mathematical Society, Providence - Rhode Island, 1986.