Eighth Eurographics Workshop on Virtual Environments (2002)
S. Müller, W. Stürzlinger (Editors)

Locating Closed Streamlines in 3D Vector Fields

Thomas Wischgoll and Gerik Scheuermann

University of Kaiserslautern
P.O. Box 3049, D-67653 Kaiserslautern
Germany

Abstract

The analysis and visualization of flows is a central problem in visualization. Topology based methods have gained increasing interest in recent years. This article describes a method for the detection of closed streamlines in 3D flows. It is based on a special treatment of cases where a streamline reenters a cell to prevent infinite cycling during streamline calculation. The algorithm checks for possible exits of a loop of crossed faces and detects structurally stable closed streamlines. These global features are not detected by conventional topology and feature detection algorithms.

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. A similar problem is discussed in [5] where residence time is used to find recirculation regions. When reaching a closed streamline the residence time is infinite. Residence time is also important information in combustion applications. There one is interested in recirculation zones with sufficient residence time for the reactions to approach completion. For instance, burning processes of gas injected into a swirling jet need a special amount of time to completely burn. Here, 2D closed streamlines of the vector field projected onto a cutting plane can be a hint for these regions as previously proposed by the authors [22].

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 criterion like elapsed time or number of integration steps to prevent infinite loops. We present here an algorithm that detects this behavior and that can be used to visualize closed streamlines. These features are an essential topological property of the field.

Topological methods have got increasing interest in Scientific Visualization since their introduction by Helman and Hesselink [3], [6], [11], [15]ch.21, [17], [18]. Our problem here is also related to the study of dynamical systems [4], [8] which have also been an application area for visualization. Koçak et al. [12] concentrate on the use of computer graphics for understanding Hamiltonian systems that appear frequently in mechanics. Hepting et al. [7] study invariant tori in four-dimensional dynamical systems by using suitable projections into three dimensions to enable detailed visual analysis of the tori. Wegenkittel et al. [21] present visualization techniques for known features of dynamical systems. Bürkle et al. [1] use a numerical algorithm developed by some of the coauthors [2] to visualize the behavior of more complicated dynamical systems. In the numerical literature, we can find several algorithms for the calculation of closed curves in dynamical systems [10], [20], but these algorithms are tailored to deal with smooth dynamical systems where a closed form solution is given. In contrast, visualization faces far more often piecewise linear or trilinear vector fields. Here, the knowledge of the grid and the linear structure of the field in the cells allow a direct approach for the search of closed streamlines. The algorithm can be integrated in the streamline calculation as we will show.

Several applications exist where closed streamlines play an important role. For instance, Wong [23] presented a filtering technique to interpret climate modeling datasets. A typical feature of a hurricane is an external high-velocity circulation with a tranquil region inside. These regions are separated by a closed streamline. Consequently, a hurricane can be identified by finding this closed streamline. Another application is the *Terrestrial Planet Finder Mission* [14]. In this mission one is interested in flying a constellation of five satellites in

formation around a 3D periodic halo orbit. These orbits are nothing else than closed streamlines in a 3D vector field.

We repeat necessary terms on vector field topology in section 2. Section 3 describes the algorithm for detecting closed streamlines. Results are presented in section 4 while concluding remarks can be found in section 5.

2. Theory

This section repeats the theoretical background and the terms used in vector field topology which are used for our algorithm. We restrict our consideration in this article to steady, linearly interpolated vector fields $v : \mathbb{R}^3 \supset D \rightarrow \mathbb{R}^3, \quad (x, y, z) \mapsto v(x, y, z)$. D is assumed to be bounded. This is the situation for many experimental or simulated vector fields that have to be visualized. We are interested in the behavior of streamlines

$$c_a : \mathbb{R} \rightarrow \mathbb{R}^3, \qquad t \mapsto c_a(t)$$

with the properties

$$\begin{aligned} c_a(0) &= a \\ \frac{\partial c_a}{\partial t}(t) &= v(c_a(t)). \end{aligned}$$

For Lipschitz continuous vector fields, we can prove the existence and uniqueness of streamlines c_a through any point $a \in D$, see [8], [13]. The actual computation of the streamlines is usually done by numerical algorithms like Euler methods, Runge-Kutta-Fehlberg methods or Predictor/Corrector methods [16], [19].

The topological analysis of vector fields considers the asymptotic behavior of streamlines. The origin set or α-limit set of a streamline c is defined by $\{p \in \mathbb{R}^3 | \exists (t_n)_{n=0}^{\infty} \subset \mathbb{R}, t_n \rightarrow -\infty, \lim_{n\rightarrow\infty} c(t_n) \rightarrow p\}$. The end set or ω-limit set of a streamline c is defined by $\{p \in \mathbb{R}^3 | \exists (t_n)_{n=0}^{\infty} \subset \mathbb{R}, t_n \rightarrow \infty, \lim_{n\rightarrow\infty} c(t_n) \rightarrow p\}$. If the α- or ω-limit set of a streamline consists of only one point, this point is a critical point or a point at the boundary ∂D of our domain D. (It is assumed that the streamline stays at the boundary point forever in this notation.)

The most common case of a α- or ω-limit set in a vector field containing more than one inner point of the domain is a limit cycle [8]. This is a streamline c_a, so that there is a $t_0 \in \mathbb{R}$ with

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

Figure 1 shows a typical example. Such a cycle is called structurally stable if, after small changes, the vector field still contains a closed streamline.

3. Detection of Closed Streamlines

We present an algorithm that detects if an arbitrary streamline c converges to a closed streamline γ as defined in section 2. This means that c has γ as α- or ω- limit, depending

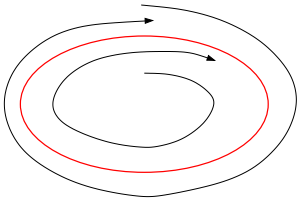

Figure 1: *A limit cycle may attract streamlines in its neighborhood.*

on the orientation of the integration. We do not assume any knowledge on the existence or location of the closed curve. The principle of the algorithm works on any piecewise defined vector field where one can determine the topology inside the pieces. In order to illustrate the main ideas of the algorithm let us start with the two dimensional case which is already proposed by the authors [22].

The basic idea of the algorithm is to determine a region of the vector field that is never left by the streamline. We assume that the data of the vector field is given on a grid consisting of triangles and/or quadrilaterals. The vectors inside a cell are interpolated so that we get an at least continuous vector field.

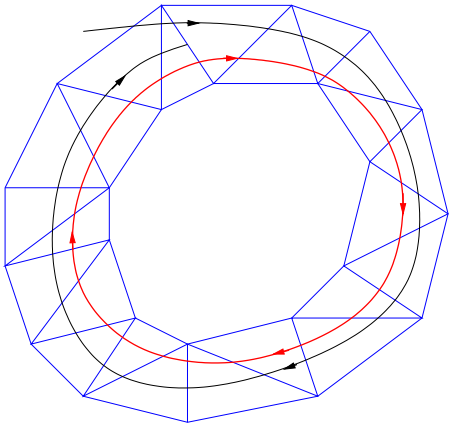

Figure 2: *A streamline approaching a limit cycle has to reenter cells.*

A streamline approaching a limit cycle has to reenter the same cell again as shown in Figure 2. In this case we check if the cells crossed by the streamline have not changed for the last two turns. This results in a *cell cycle* which identifies the above mentioned region. To examine if this cell cycle is left by the streamline we detect possible changes by checking the edges of the cells of the cell cycle. Therefore we find the points on each edge, which we call *potential exits*, where an outflow out of the cell cycle may occur near these points. These points are identical with the vertices of the edge and points where the vector field is tangential to the edge.

Then we have to figure out if the actually investigated

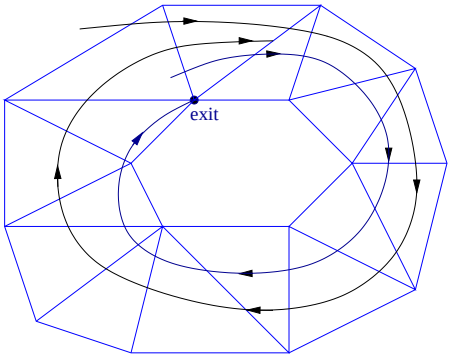

Figure 3: *If a real exit can be reached, the streamline will leave the cell cycle.*

streamline will leave the cell cycle near such an exit. Therefore we integrate a streamline backwards from the potential exit to see if it leaves the cell cycle. If this is not the case after the streamline crosses every cell involved in the cell cycle it is shown that this backward integration converges to the streamline we actually investigate. If, for instance, two vectors on an edge point in the same direction as the streamline, all vectors point in similar direction since we interpolate linearly at an edge. While detecting the cell cycle we found two vectors on an edge during following the streamline for the last two turns that fulfill this criterion. Consequently, the streamline cannot turn around and cross the edge in the opposite direction in between. Therefore it is sufficient to consider only one full turn of the backward integration as in figure 3. We call this potential exit a *real exit* because the streamline will leave the cell cycle after a finite number of turns near that exit. Figure 3 displays an example for that case.

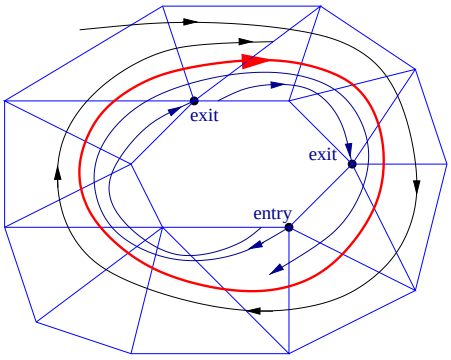

Figure 4: *If no real exit can be reached, the streamline will approach a limit cycle.*

If case of the backward integrated streamline leaving the cell cycle it diverges from the actually investigated streamline. Consequently, the streamline we want to check cannot leave the cell cycle in that potential exit, because then we have an inflow into our region which will leave again at the exit as shown in figure 4. Consequently this is not a real exit.

If there is no real exit for the streamline, we have proven that the streamline will never leave the cell cycle. Then there exists a closed streamline in our cell cycle and the integral curve tends toward it. If we can find a real exit we have to continue the streamline calculation. A proof of this algorithm can be found in [22].

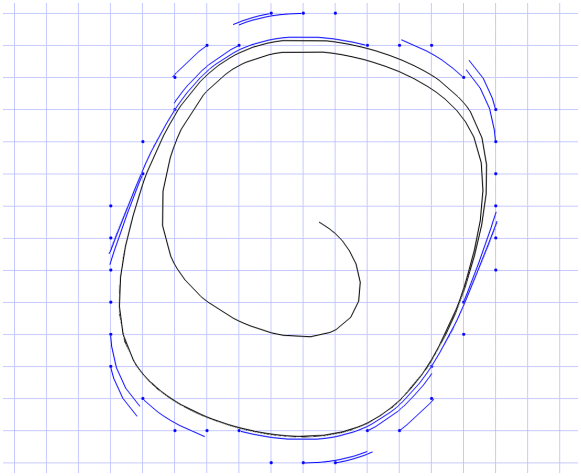

Figure 5: *Exits of a cell cycle.*

Figure 5 illustrates the situation which shows a real example. There we start a streamline near the source in the center of the figure. This streamline spirals until we find the first cell cycle. The figure also shows all exits and its backward integrations which are drawn in blue color and the streamline itself colored black. The grid is displayed in light blue. In this example, every potential exit is shown.

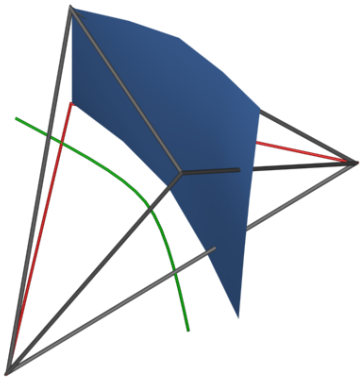

Figure 6: *Backward integrated surface.*

The principle works similar in 3D. Again, we follow the streamline until we detect a cell cycle. The difference exists in the exits. In 3D it is not sufficient to simply integrate backwards at the vertices. Figure 6 illustrates that. In some cases the backward integrations starting at the vertices leave the cell cycle. But some parts of the surface still stay inside the cell cycle which may approach the streamline. We need to figure out if there is any part that is backward integrated starting at the edge approaches the actually investigated streamline. Therefore we have to integrate backwards starting at the whole edge. Consequently we have to compute a streamsurface instead of a streamline. Therefore we use a simplified version of the streamsurface algorithm introduced by Hultquist [9]. 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 created an approximation of the streamsurface that respects the given error limit. Figure 7 demonstrates this case. At the red point the two backward integrations are too far away from each other. So an intermediate streamline is started in the middle to achieve a better accuracy.

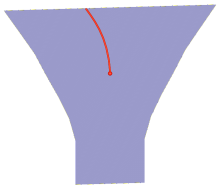

Figure 7: *Intermediate backward integrated streamline.*

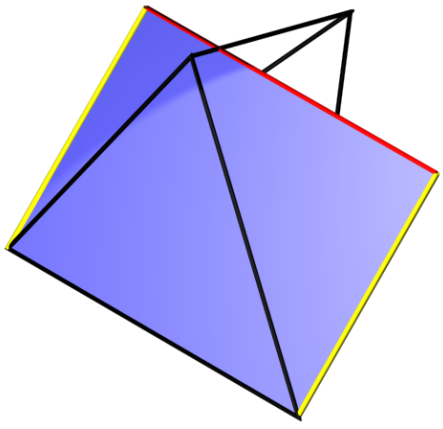

Figure 8: *Backward integration in one cell.*

The integration stops if the whole streamsurface leaves the cell cycle. But to construct the surface properly we may have to continue one single 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 already left it. Figure 8 shows this situation. Both streamlines - shown as yellow lines - leave the cell, in fact they leave right after they started. But the integration process must be continued until the whole surface created inside the cell by these two streamlines leaves the cell. This is marked by the red line at the end of the streamsurface.

Figure 9 (see color plates) shows an example of our backward integration step. There, also the closed streamline is shown in red and the cell cycle is shown in blue. Every backward integrated streamsurface leaves the cell cycle. Consequently, the existence of a closed streamline is proven. Then we can find the exact location by continuing the integration process of the streamline that we actually investigate until the difference between two turns is small enough. This numerical criterion is sufficient at this point since we have proven that the streamline will never leave the cell cycle.

4. Results

To test our implementation we created a synthetic dataset which includes one closed streamline. We first produced a two dimensional vector field which is symmetrical with respect to the y-axis. Additionally, all vectors residing at the y-axis where zero. Then we rotated it around the y-axis and distorted it a little bit to get a three dimensional flow. Figure 10 (see color plates) shows the result. The closed streamline is colored red. To visualize a little bit of the surrounding flow several streamlines are drawn. Obviously, every streamline is attracted by the closed streamline. After a short while the streamline spirals around the closed streamline until it completely merges into that one. Figure 11 shows this spiraling effect in detail with some more streamlines. Again, the closed streamline is shown in red and the other streamlines are colored white. Closed streamlines in three-dimensional flows can act like sources or sinks as we can see from this visualization.

5. Conclusion

We presented an algorithm which is able to detect if a streamline runs into a closed streamline. We only need to extend the integration process with a check routine. Since it uses no information on the existence or location of the closed streamlines, it can be used to find these important features. The algorithm relies on the fact that the vector field is interpolated linearly. All examples were calculated using a vector field given on a tetrahedral grid. But the algorithm also works with parallelepiped grids.

In the future we want to extend our algorithm so that it is able to also detect strange attractors like for instance the

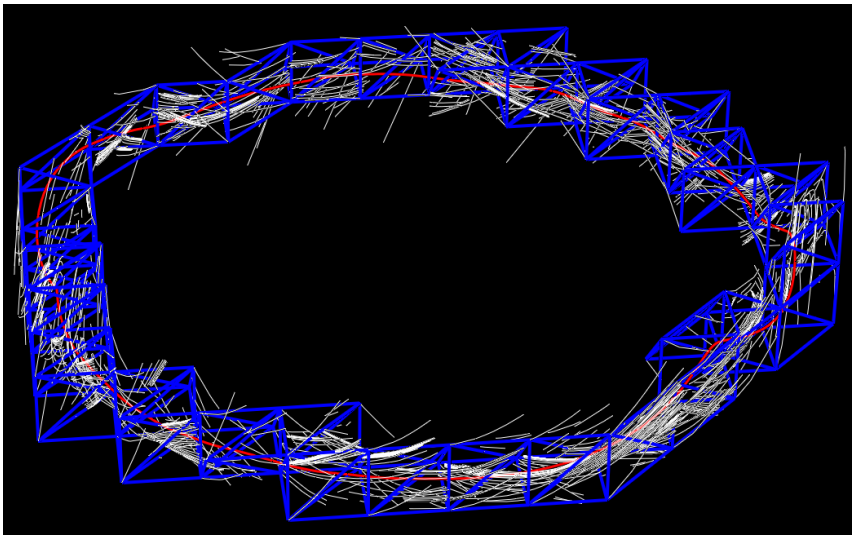

Figure 9: *Closed streamline including cell cycle and backward integrations.*

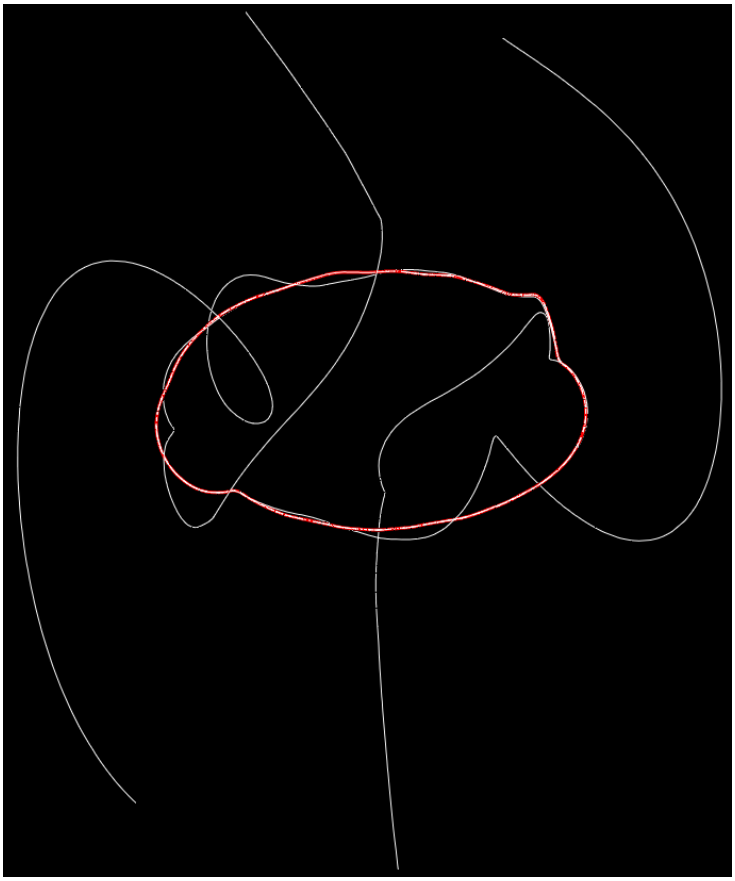

Figure 10: *Closed streamline in a 3D vector field.*

Lorentz Attractor. There we do not get a simple cell cycle but also a bunch of cells that are always crossed by the streamline and never left. The difference is that these cells are not topologically linked in a list but may be arranged in a more complicated way.

6. Acknowledgments

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, Martin Öhler, Max Langbein, Aragorn Rockstroh, René Schätzl and Xavier Tricoche for their programming efforts and Inga Scheler for helping with some of the figures. The continuous support of all members of the computer graphics and visualization team in Kaiserslautern gives us a nice working environment.

References

1. 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.
2. M. Dellnitz and O. Junge. On the Approximation of Complicated Dynamical Behavior. *SIAM Journal on Numerical Analysis*, 36(2), pp. 491 – 515, 1999.
3. 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.
4. J. Guckenheimer and P. Holmes. *Dynamical Systems and Bifurcation of Vector Fields*. Springer, New York, 1983.
5. R. Haimes. Using residence time for the extraction of recirculation regions. *AIAA Paper 99-3291*, 1999.
6. 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.
7. D. H. Hepting, G. Derks, D. Edoh, and R. R. D. 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.
8. M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems and Linear Algebra*. Academic Press, New York, 1974.
9. J. P. M. Hultquist. Constructing streaam surface in steady 3d vector fields. In *Proceedings IEEE Visualization 1992*, pp. 171–177. IEEE Computer Society Press, Los Alamitos CA, 1992.
10. M. Jean. Sur la méthode des sections pour la recherche de certaines solutions presque périodiques de syst'emes forces periodiquement. *International Journal on Non-Linear Mechanics*, 15, pp. 367 – 376, 1980.
11. D. N. Kenwright. Automatic Detection of Open and Closed Separation and Attachment Lines. In D. Ebert, H. Rushmeier, and H. Hagen, editors, *IEEE Visualization '98*, pp. 151–158, Research Triangle Park, NC, 1998.
12. H. Koçak, F. Bisshopp, T. Banchoff, and D. Laidlaw. Topology and Mechanics with Computer Graphics. *Advances in Applied Mathematics*, 7, pp. 282 – 308, 1986.
13. S. Lang. *Differential and Riemannian Manifolds*. Springer, New York, third edition, 1995.
14. 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.
15. G. M. Nielson, H. Hagen, and H. Müller, editors. *Scientific Visualization, Overviews, Methodologies, and Techniques*. IEEE Computer Society, Los Alamitos, CA, USA, 1997.
16. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1992.
17. G. Scheuermann, B. Hamann, K. I. Joy, and W. Kollmann. Visualizing local Vetor Field Topology. *Journal of Electronic Imaging*, 9(4), 2000.
18. G. Scheuermann, H. Krüger, M. Menzel, and A. Rockwood. Visualizing Nonlinear Vector Field Topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2), pp. 109–116, April–June 1998.
19. J. Stoer and R. Bulirsch. *Numerische Mathematik 2*. Springer, Berlin, 3 edition, 1990.
20. 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.
21. R. Wegenkittel, H. Löffelmann, and E. Gröller. Visualizing the Behavior of Higher Dimensional Dynamical Systems. In R. Yagel and H. Hagen, editors, *IEEE Visualization '97*, pp. 119 – 125, Phoenix, AZ, 1997.
22. T. Wischgoll and G. Scheuermann. Detection and Visualization of Closed Streamlines in Planar Flows. *IEEE Transactions on Visualization and Computer Graphics*, 7(2), 2001.
23. P. C. Wong, H. Foote, R. Leung, E. Jurrus, D. Adams, and J. Thomas. Vector fields simplification – a case study of visualizing climate modeling and simulation data sets. In *Proceedings IEEE Visualization 2000*, pp. 485–488. IEEE Computer Society Press, Los Alamitos CA, 2000.

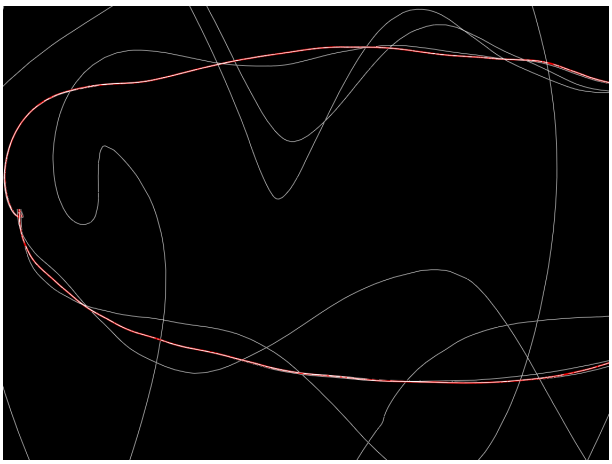

Figure 11: *Close up of the spiraling effect around the closed streamline.*