Locating Closed Hyperstreamlines in Second Order Tensor Fields

Thomas Wischgoll[1] and Joerg Meyer[1]

[1]University of California at Irvine, Irvine, CA 92697-2625, USA

Abstract. The analysis and visualization of tensor fields is an advancing area in scientific visualization. Topology-based methods that investigate the eigenvector fields of second order tensor fields have gained increasing interest in recent years. Most algorithms focus on features known from vector fields, such as saddle points and attracting or repelling nodes. However, more complex features, such as closed hyperstreamlines are usually neglected. In this article, a method for detecting closed hyperstreamlines in tensor fields as a topological feature is presented. The method is based on a special treatment of cases where a hyperstreamline reenters a cell and prevents infinite cycling during hyperstreamline calculation. The algorithm checks for possible exits of a loop of crossed edges and detects structurally stable closed hyperstreamlines. These global features cannot be detected by conventional topology and feature detection algorithms used for the visualization of second order tensor fields.

1 Introduction

Many problems in natural science and engineering involve tensor fields. For example, stresses, viscous stresses, rate-of-strain, and momentum flux density are described as symmetric tensor fields. Due to the multivariate nature of tensor fields, appropriate methods for visualization are required in order to investigate the data. This, of course, includes the detection of special properties of a tensor field, for instance topological features, that can be emphasized on in the visualization to reduce visual clutter.

The topological analysis of tensor fields as described by Hesselink et al. [DH94] focuses on degenerate points and their topological meaning. A special type of degenerate point, the trisector point, corresponds to saddle points in vector fields from a topological point of view. Hyperstreamlines, that follow the vectors in a particular, previously chosen eigenvector field, lead to separatrices inside a 2-D tensor field. Similarly, a detailed analysis of the degenerate points [HLL97] in a symmetric, second order 3-D tensor field leads to hyperstreamlines [HD93] depicting parts of the topology of the tensor field. To incorporate the two remaining eigenvector fields that are not used for integrating the hyperstreamline, an ellipse spanned by those two eigenvectors is used, resulting in a tube-shaped representation that follows the main eigenvector field.

Obviously, integrating curves inside an eigenvector field plays an important role in such a visualization. The qualitative nature of these curves can be

studied with topological methods developed originally for dynamical systems. Especially in the area of fluid mechanics, topological analysis and visualization have been used with success [GLL91,HH91,Ken98,SHJK00].

Besides point-shaped singularities, other topological features exist in tensor fields. Similar to closed streamlines in vector fields [WS01,WS02], closed hyperstreamlines can be found in tensor fields. These integral curves within an eigenvector field are closed, therefore forming a loop. Their importance stems from the fact that quite often neighboring integral curves either tend to bend toward the loop or originate from the loop (i. e. tend to move toward the loop after reversing the direction of time). This is a well established result from dynamical systems theory [GH83,HS74]. Consequently, being able to determine closed hyperstreamlines in tensor fields is an important addition to tensor field topology.

Several publications have dealt with related topics in vector fields. Hepting et al. [HDER95] 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. [WLG97] present visualization techniques for known features of dynamical systems. Bürkle et al. [BDJ+99] use a numerical algorithm developed by some of the coauthors [DJ99] to visualize the behavior of more complicated dynamical systems. In the literature on numerical methods, one can find several algorithms for the calculation of closed curves in dynamical systems [Jea80,vV87], but these algorithms are tailored to dealing with smooth dynamical systems where a closed form solution is given.

In most cases, visualization deals with piecewise linear, bilinear or trilinear data. In this paper, a suitable algorithm for this situation is presented which can be integrated into a computational algorithm for standard hyperstreamlines. While computing a hyperstreamline, the algorithm tracks the visited cells and checks for repetition. Upon revisiting a cell, the algorithm tests if the hyperstreamline stays in the same cell cycle indefinitely. For this purpose, the boundary of the current cell cycle is investigated to determine if the integral curve can cross this boundary.

The structure of the remainder of this paper is as follows. First, a short description of the mathematical background is given. Subsequently, the algorithm for detecting closed hyperstreamlines is discussed. Finally, results of the algorithm are presented and concluding remarks are given.

2 Mathematical Background

This section provides the necessary theoretical background and the mathematical terms used in the algorithm. The scope of this article is restricted to steady, linearly interpolated three-dimensional second order tensor fields

defined on a tetrahedral grid:

$$t : \mathbb{R}^3 \supset D \to Mat(3 \times 3, \mathbb{R}), \quad (x, y, z) \mapsto \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} =: (t_{ij}) \ .$$

D is assumed to be bounded. This is the case for almost every experimental or simulated tensor field that has to be visualized. A tensor described as a three-by-three matrix can be decomposed into two Matrices $S = (s_{ij})$ and $A = (a_{ij})$ such that the equation $T = A+S$ holds where $s_{ij} = \frac{1}{2}(t_{ij}+t_{ji})$ and $a_{ij} = \frac{1}{2}(t_{ij} - t_{ji})$. Then S is called the symmetric part of the tensor T since the equation $s_{ij} = s_{ji}$ holds for every i and j, while A is the antisymmetric part of the tensor T with $a_{ij} = -a_{ji}$. Since the antisymmetric part of a tensor basically describes a rotation only, this paper focuses on symmetric, second order tensors. For these, the eigenvalues exist and are real. The corresponding eigenvectors form an orthogonal basis of $\mathbb{R}^3$. This allows the computation of hyperstreamlines by using one eigenvector field for calculating integral curves, while the remaining two eigenvalues can be used to span an ellipse resulting in a tube that follows the direction of the followed eigenvector field [HD93]. In addition, the ellipse can be scaled using the remaining eigenvalues.

According to the definition by Hesselink et al. [DH94], the topology of a tensor field is the topology of its eigenvector fields. Consequently, critical points known from vector fields, such as saddles, nodes, and foci, occur as singularities in tensor fields as well. Due to the way hyperstreamlines are defined, an additional kind of topological feature results from that definition. In the case of two or more eigenvalues being identical, only two or one eigenvector(s) can be determined. As a consequence, hyperstreamlines end at such a point, since there is no unique way to continue. Therefore, these locations are usually referred to as degenerate points. Recent studies as described in chapter 14 by Zheng et al. show that these points form lines instead of individual points. Consequently, this type of singularity occurs more often compared to vector field singularities. However, with respect to hyperstreamlines, degenerate points play only a minor role since a hyperstreamline that is terminated by a degenerate point can no longer be a closed hyperstreamline.

Since hyperstreamlines follow one of the eigenvector fields, the behavior of integral curves $h_a : \mathbb{R} \to Mat(3 \times 3, \mathbb{R})$, $\tau \mapsto h_a(\tau)$ can be described by their properties $h_a(0) = a$ and $\frac{\partial h_a}{\partial \tau}(\tau) = t(h_a(\tau))$. For a Lipschitz continuous eigenvector field, one can prove the existence and uniqueness of integral curves h_a through any point $a \in D$, see [HS74,Lan95]. The actual computation of integral curves is usually done by numerical algorithms, such as Euler methods, Runge-Kutta-Fehlberg methods or Predictor/Corrector methods [SB90].

The topology of a tensor field is defined as the topology of its eigenvector fields. Thus, the topological analysis considers asymptotic behavior of integral curves in these different eigenvector fields. In order to be able

to clearly identify where integral curves are coming from and where they are going, one can define two different sets describing the area covered by the integral curve for approaching a positive or a negative infinite parameter value resulting in the α- and ω-limit set, respectively. The α-limit set of an integral curve h is defined by $\{p \in \mathbb{R}^3 | \exists (\tau_n)_{n=0}^{\infty} \subset \mathbb{R}, \tau_n \rightarrow -\infty, \lim_{n \rightarrow \infty} h(\tau_n) \rightarrow p\}$. The ω-limit set of an integral curve h is defined by $\{p \in \mathbb{R}^3 | \exists (\tau_n)_{n=0}^{\infty} \subset \mathbb{R}, \tau_n \rightarrow \infty, \lim_{n \rightarrow \infty} h(\tau_n) \rightarrow p\}$. If the α- or ω-limit set of an integral curve consists of only one point, this point is a critical point or a point on the boundary ∂D. (It is usually assumed that the integral curve stays at the boundary point indefinitely.)

The most common case of an α- or ω-limit set in an eigenvector field containing more than one inner point of the domain is a closed hyperstreamline [HS74]. This is an integral curve h_a with the property that there is a $\tau_0 \in \mathbb{R}$ with $h_a(\tau + n\tau_0) = h_a(\tau) \quad \forall\, n \in \mathbb{N}$. Consequently, for every closed hyperstreamline there exist a hyperstreamline that converges to this closed hyperstreamline when integrating in positive or in negative direction. This fact will be exploited later in the algorithm for detecting this topological feature.

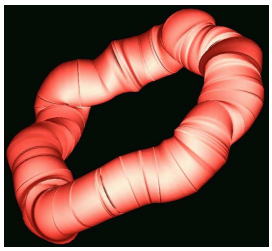

Fig. 1. Example of a closed hyperstreamline in a 3D tensor field.

Figure 1 shows a typical example of a closed hyperstreamline in a three-dimensional tensor field. Such a closed hyperstreamline is called *structurally stable* if, after small changes in the tensor field, the closed hyperstreamline remains.

3 Detection of Closed Hyperstreamlines

The concept of detecting closed hyperstreamlines in a three-dimensional second order tensor field is similar to the three-dimensional vector case [WS02], because a hyperstreamline follows one of the eigenvector fields. Once such a closed hyperstreamline is detected in one of the eigenvector fields a tubular structure can be built around the closed curve in a fashion similar to regular hyperstreamlines.

Since the algorithm for detecting closed hyperstreamlines in symmetric, second order 3-D tensor fields is very similar to the 3-D vector case it will

be repeated here only briefly. Details can be found in [WS02]. Before the algorithm itself is explained, a few notations need to be defined. We use the term *current hyperstreamline* to describe the hyperstreamline currently under testing for the loop condition.

To reduce computational cost, the hyperstreamline is first integrated using a Runge-Kutta method of fourth order with an adaptive step size control. In order to determine eigenvectors required for this step, the tensors are interpolated tri-linearly inside the cells. Using the interpolated tensors, eigenvalues and eigenvectors can then be calculated. This is known to be numerically more stable. Otherwise, large numerical errors have the potential to modify the topology of the tensor field in such a way that singularities do not form lines any more. Further information about tensor interpolation is described by Kindlmann et al. [KWH00].

During this integration step, every cell that is crossed by the current hyperstreamline is stored in a list. If a hyperstreamline approaches a loop it reenters the same cell. This results in a *cell cycle* consisting of a finite sequence of neighboring cells $c_0, \ldots, c_n$ with $c_0 = c_n$ crossed by the current hyperstreamline.

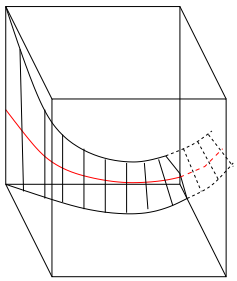

Fig. 2. Backward integrated hyperstreamsurface.

This cell cycle identifies a region where it needs to be determined if the current hyperstreamline can escape that cycle. To check this, every backward integrated hyperstreamline starting at an arbitrary point on a face of the boundary of the cell cycle has to be considered. Looking at the edges of a face it can be seen directly that it is not sufficient to just integrate hyperstreamlines backward which originate at the vertices of that edge. This is due to the fact that individually started hyperstreamlines only cover a discrete portion of the edge. Instead, a hyperstreamsurface has to be computed with the edge in question as initial condition. Figure 2 shows an example where a single cell and a backward integration is depicted. A hyperstreamsurface is started at the rear left vertical edge and turns inside the cell towards the rear lower right corner of the cell. The parts of the hyperstreamsurface that are outside the cell are drawn as dashed lines. The two edges of the hyperstreamsurface which are identical with a backward integrated hyper-

streamline started at the vertices of the rear left edge of the cell leave the cell at the lower and rear face, respectively. However, a hyperstreamline started at the center of the rear left edge of the cell (drawn in red) leaves the cell at the right face. If the cell cycle continuous at the right face the backward integration would be considered as leaving the cell if only the backward integrated hyperstreamlines starting at the vertices of an edge would have been considered. As a consequence, we have to find another definition for exits as in the two-dimensional vector case [WS01]. Thus, *potential exit edges*, which are the starting points of the backward integration step, are defined as the edges on the boundary of the cell cycle. In analogy to the two-dimensional case, curves on a boundary face of a cell contained in the cell cycle where the eigenvector field is tangential to the face is identified as a *potential exit edge* as well.

Due to the nature of the interpolation inside the tetrahedral grid, it can be shown that there will be at most a two-dimensional curve on the face of a tetrahedron of that grid where the eigenvector field is tangential to the face, the whole face is tangential to the eigenvector field, or there is no tangential area at all. An isolated point on the face where the eigenvector field is tangential to the face cannot occur and do not need to be considered as a potential exit.

The potential exit edges as previously defined then serve as initial conditions for the backward integration step. Hyperstreamlines are computed originating at the potential exit edges. These hyperstreamsurfaces are then called *backward integrated hyperstreamsurface*. In case part of the hyperstreamline leaves the cell cycle the current hyperstreamline can leave at the cell cycle at that location and there is no closed hyperstreamline present in the current cell cycle. Hence, this exit edge is referred to as a *real exit edge*. It is worthwhile noting that the backward integrating step is insensitive to degenerate points. On encounter of a degenerate point, a hyperstreamsurface may separate into two parts, but can still be computed. For the backward integrated hyperstreamsurface the streamsurface algorithm introduced by Hultquist [Hul92] is used. Further details about the described methodology can be found in [WS02] due to its similarity to the vector case.

Applying this motivation to symmetric, second order 3-D tensor fields, an algorithm for detecting closed hyperstreamlines can be described. First, a hyperstreamline is integrated using a standard integration method. During that process the cells covered by the current hyperstreamline are traced to check if a cell cycle is reached. Then, all potential exit edges are identified by going backward through the crossed cells. As a final step, all exit edges are validated by integrating a hyperstreamsurface backward from every potential exit edge through the whole cell cycle. If there is no real exit edge, meaning that no backward integrated hyperstreamline left the cell cycle, the current hyperstreamline cannot leave the cell cycle. Consequently, there exists a closed hyperstreamline within the cell cycle on condition that there is

no singularity contained in the cell cycle. On the other hand, if a real exit edge exists, then there is no closed hyperstreamline present in the current cell cycle. Consequently, the criterion serves as both a necessary as well as a sufficient condition. The proof for this algorithm is similar to the vector case and can be found in [WS02].

4 Results

To test the implementation, a synthetic data set was created which includes one closed hyperstreamline in the minor eigenvector field. To compute this data set, a two-dimensional vector field which contains two sinks and is symmetrical with respect to the y-axis is used as a starting point. In addition, all vectors residing at the y-axis are zero in this vector field. By rotating this two-dimensional vector field about the y-axis, a three-dimensional flow is created. To convert each vector v in this vector field into a tensor, basic linear algebra methods are used. First, two vectors v_1 and v_2 are determined in such a way that these two vectors in combination with the one from the vector field form an orthonormal basis of $\mathbb{R}^3$. Defining a matrix $T = (v, v_1, v_2)$ yields to a tensor $t = T \cdot E \cdot T^{-1}$, where E is a matrix defined as

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} .$$

Determining the minor eigenvector in a tensor field that was created in such a way, results in exactly the same vector that was plugged in initially. Consequently, the tensor field contains a single closed minor hyperstreamline. Figure 3 shows the hyperstreamline that was detected by the algorithm. The wavy appearance of the closed hyperstreamline is due to the way this data set was generated; the location of the closed hyperstreamline is determined very accurately by the algorithm. The hyperstreamline is drawn only as the center line without considering the medium and major eigenvectors. In addition, a hyperstreamline which was computed by a regular hyperstreamline algorithm

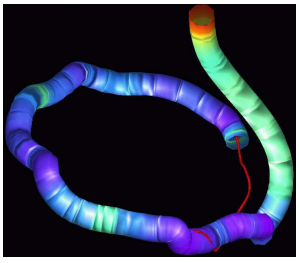

Fig. 3. Closed hyperstreamline (minor eigenvector field) in combination with a regular hyperstreamline.

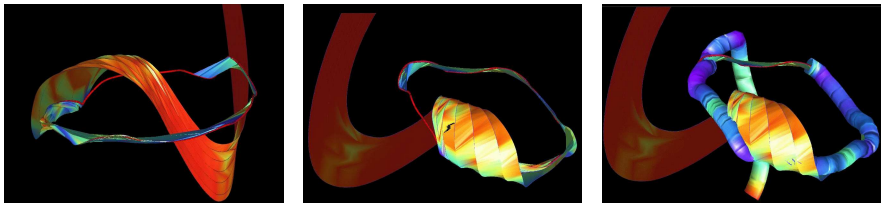

Fig. 4. Closed hyperstreamline including hyperstreamsurfaces exposing the surrounding tensor field (minor eigenvector field).

was started in the vicinity of the closed hyperstreamline. The segments of this hyperstreamline are colored according to the minor eigenvalue. Due to the attracting nature of this closed hyperstreamline, the regularly computed hyperstreamline approaches it and eventually merges with the first as can be seen in the figure.

Figure 4 shows the same closed hyperstreamline with two hyperstreamsurfaces. The hyperstreamsurfaces – just like the hyperstreamline – are attracted by this closed hyperstreamline. The hyperstreamsurface becomes smaller and smaller while it spirals around the closed hyperstreamline. After a few turns around the closed hyperstreamline, the ellipses are only slightly wider than the closed hyperstreamline itself, finally causing the hyperstreamsurface to completely merge with the hyperstreamline. A rather arbitrary color scheme is used for the hyperstreamsurfaces to enhance the three-dimensional impression.

To apply the algorithm to a more common data set, a single point load data set was used. Here, a force is applied to an infinite half space. The stress-strain tensor field is determined to describe the pressure inside that infinite half space. Figure 5 shows the result of the algorithm. Two major hyperstreamlines were computed, starting at the top of the figure and ending at the lower left corner. The force acting on the infinite half space is attached to the center of the funnel-shaped end of the larger hyperstreamline in the

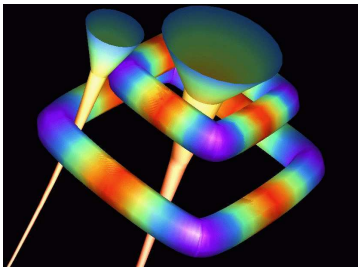

Fig. 5. Closed hyperstreamlines in single point load data set.

figure. Several closed hyperstreamlines can be found in the minor eigenvector field of this data set. In fact, there is a complete surface totally covered by closed hyperstreamlines. Figure 5 shows just two of these hyperstreamlines to avoid visual clutter. Due to the interpolation used for computing the tensors in the data set, the closed hyperstreamlines do not appear as perfect circles as would be expected from the simulation. Similar to the previous example, the segments of all hyperstreamlines are colored according to the eigenvalue whose eigenvector is used to follow the hyperstreamline.

5 Conclusion

A method for detecting closed hyperstreamlines in symmetric, second order tensor fields was presented. It was found that most methods that have been established for vector fields can be applied to tensor fields as well. As can be seen in the first example of section 4, closed hyperstreamlines can have an attracting property and therefore form a topological feature. Consequently, this feature, which is missing in topological analysis algorithms, was added.

6 Acknowledgments

This research was supported by the National Institute of Mental Health (NIMH), the National Partnership for Advanced Computational Infrastructure (NPACI), and the Electrical Engineering and Computer Science Department of the University of California, Irvine. Further, the authors like to thank Gerik Scheuermann at the University of Leipzig, Germany, and the visualization team in Kaiserslautern, Germany for providing the single point load data set.

References

[BDJ+99] 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*, pages 33 – 36, San Francisco, 1999.

[DH94] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In *IEEE Visualization '94 Proceedings*, pages 140–147, Los Alamitos, 1994. IEEE Computer Society.

[DJ99] M. Dellnitz and O. Junge. On the Approximation of Complicated Dynamical Behavior. *SIAM Journal on Numerical Analysis*, 36(2):491 – 515, 1999.

[GH83] J. Guckenheimer and P. Holmes. *Dynamical Systems and Bifurcation of Vector Fields*. Springer, New York, 1983.

[GLL91] 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*, pages 33 – 40, San Diego, 1991.

[HD93] L. Hesselink and T. Delmarcelle. Visualizing second order tensor fields with hyperstreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.

[HDER95] 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*, pages 342 – 345, Atlanta, GA, 1995.

[HH91] J. L. Helman and L. Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May 1991.

[HLL97] L. Hesselink, Y Lavin, and Y. Levy. Singularities in nonuniform tensor fields. In *IEEE Visualization '97*, pages 59–66. IEEE Computer Society, July 10 1997.

[HS74] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems and Linear Algebra*. Academic Press, New York, 1974.

[Hul92] J. P. M. Hultquist. Constructing Stream Surface in Steady 3D Vector Fields. In *Proceedings IEEE Visualization '92*, pages 171–177. IEEE Computer Society Press, Los Alamitos CA, 1992.

[Jea80] M. Jean. Sur la méthode des sections pour la recherche de certaines solutions presque périodiques de systèmes forces periodiquement. *International Journal on Non-Linear Mechanics*, 15:367 – 376, 1980.

[Ken98] 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*, pages 151–158, Research Triangle Park, NC, 1998.

[KWH00] G. Kindlmann, D. Weinstein, and D. Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):124–138, 2000.

[Lan95] S. Lang. *Differential and Riemannian Manifolds*. Springer, New York, third edition, 1995.

[SB90] J. Stoer and R. Bulirsch. *Numerische Mathematik 2*. Springer, Berlin, 3 edition, 1990.

[SHJK00] G. Scheuermann, B. Hamann, K. I. Joy, and W. Kollmann. Visualizing local Vetor Field Topology. *Journal of Electronic Imaging*, 9(4):356–367, 2000.

[vV87] M. van Veldhuizen. A New Algorithm for the Numerical Approximation of an Invariant Curve. *SIAM Journal on Scientific and Statistical Computing*, 8(6):951 – 962, 1987.

[WLG97] 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*, pages 119 – 125, Phoenix, AZ, 1997.

[WS01] T. Wischgoll and G. Scheuermann. Detection and Visualization of Closed Streamlines in Planar Flows. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165–172, 2001.

[WS02] T. Wischgoll and G. Scheuermann. 3D Loop Detection and Visualization in Vector Fields. In *Visualization and Mathematics 2002*, pages 151–160, Berlin, Germany, 2002.