EUROGRAPHICS 2001 / A. Chalmers and T.-M. Rhyne
(Guest Editors)

Volume 20 (2001), Number 3

Tensor Topology Tracking: A Visualization Method for Time-Dependent 2D Symmetric Tensor Fields

X. Tricoche, G. Scheuermann, H. Hagen

Abstract

Topological methods produce simple and meaningful depictions of symmetric, second order two-dimensional tensor fields. Extending previous work dealing with vector fields, we propose here a scheme for the visualization of time-dependent tensor fields. Basic notions of unsteady tensor topology are discussed. Topological changes - known as bifurcations - are precisely detected and identified by our method which permits an accurate tracking of degenerate points and related structures.

1. Introduction

Tensor fields play a key role in many branches of physics and engineering sciences. Second-order tensor fields in particular are involved in solid and fluid mechanics as well as in the theory of general relativity. Therefore, the visualization of second-order tensor fields is a challenging issue that has received an increasing interest in the last decade. A major contribution has been the PhD thesis of Delmarcelle[1]. For two-dimensional symmetric, second-order tensor fields, this author analyzed and visualized the topology of the related orthogonal eigenvector fields. This work partially extends the qualitative theory of dynamical systems dealing with (directional) vector fields to tensor and undirectional eigenvector fields. This has been motivated by the fact that topology-based approaches have already led to very successful methods for visualizing vector fields[3]. As in the vector case, drawing the topology of a symmetric, second-order tensor field yields a simple graph representation where the edges are formed by particular integral curves computed in the eigenvector field called *separatrices* and the vertices are singularities of the tensor field (where eigenvectors cannot be determined uniquely) called *degenerate points*. The appeal of this graph arises from decomposing the plane into sub-regions where all integral curves are structurally equivalent. A scaling method has been designed to simplify the topological graph in the case of turbulent tensor fields[4] and a topology-based technique has also been applied to the visualization of three-dimensional, symmetric, second-order tensor fields[5].

Now, when considering unsteady tensor fields, the method must be extended to depict structural changes affecting degenerate points and related separatrices during their motion over time. Such changes are called bifurcations and have a fundamental mathematical meaning. To handle time-dependent tensor fields, Delmarcelle has suggested to successively analyze discrete time steps and to correlate the topologies found in a three-dimensional picture where time is visualized as third space dimension. Nevertheless, the separatrices are not drawn which deprives the resulting picture of an essential part of the topological information. Furthermore, this discrete approach fails to properly locate and identify the bifurcations that occur between two given time steps whereas these features are indispensable for understanding the structures observed in discrete samples. In fact, what is required is to embed the discrete time steps in a three-dimensional space/time continuum to enable an analytic tracking of tensor field topology. In previous work[7], the authors presented a method based upon this basic principle to track the critical points of unsteady planar vector fields in order to visualize time-dependent topology. The precise detection and identification of local bifurcations was used as basis for the depiction of the topological graph. A similar approach is used here for the purpose of tensor topology visualization.

The paper is structured as follows: First, we introduce convenient notations to be used later on. Then we recall basic notions of symmetric tensor field topology in the steady case. In the next section, we move to time-dependent tensor fields and focus on local bifurcations (i.e. bifurcations that only affect one or several degenerate points) that may be encountered in the context of our method. The following section presents the data structure used to model the continuous space/time domain required for a precise topology tracking: Grid and corresponding interpolation scheme are introduced as well as the related method to track degenerate points through a single cell. After this, we describe our algorithm for tracking degenerate points through the whole grid and identifying bifurcations on the way. This permits tracking and depiction of the whole topological structure as explained in section 7. Results are shown in section 8.

2. Notations

In the following, we will focus on unsteady second-order, symmetric tensor fields defined over a bounded space/time domain. For convenience, we will use the following notations. The tensor field is denoted by T, where T is a function of the x and y coordinates in the plane and of time t.

$$T = \begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix}$$

The first order Taylor expansion of the tensor field in the vicinity of an arbitrary point (x_0, y_0) can be written (for a fixed value of time):

$$\begin{aligned} \frac{T_{11}-T_{22}}{2} &= a(x-x_0)+b(y-y_0)+o(||(x-x_0,y-y_0)||) \\ T_{12} &= c(x-x_0)+d(y-y_0)+o(||(x-x_0,y-y_0)||) \end{aligned}$$

3. Tensor Field Topology

We give a brief overview in this section of the fundamental notions of second-oder, symmetric tensor field topology as introduced by Delmarcelle [2] [1] and present the new concepts associated with time-dependent topology. The basic idea is to study the structure of a 2D symmetric tensor field by considering one of its two (orthogonal) eigenvector fields. These eigenvector fields have a sign indeterminacy which distinguishes their topology from the one of 2D vector fields. Nevertheless, like in the vector case, tensor topology is formed by singularities (called *degenerate points*) connected by particular integral curves (or *separatrices*) bounding subdomains in which all curves are structurally equivalent. We focus on these topics after the introduction of the tensor index.

3.1. Tensor Index

The index (also known as Poincaré index) plays a key role in the qualitative analysis of vector field topology [6]. For tensor fields, there exists a similar definition that will prove very useful in the following.

Definition 3.1 The **tensor index** of a closed, non self intersecting curve lying in the domain of definition of a tensor field T is the number of counterclockwise revolutions of the eigenvectors of T while traveling once along the curve in a counterclockwise direction.

The major interest of the tensor index is when being computed around a degenerate point (c.f. 3.2) as shown in Fig. 1. As a matter of fact, the index of a closed curve enclosing no degenerate point is zero whereas it is non zero when enclosing a single simple degenerate point. Furthermore, the indices computed on two curves enclosing the same degenerate point are equal.

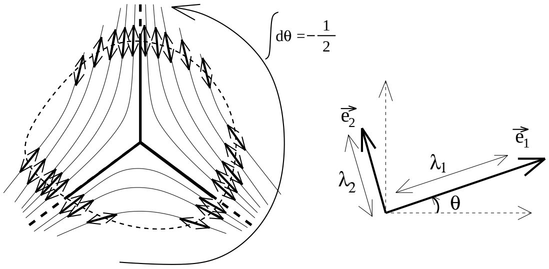

Figure 1: *Index of closed curve around a degenerate point*

3.2. Degenerate Points

As said previously, the visualized topology corresponds to a particular eigenvector field. Yet, this definition does not apply at positions where both eigenvalues are equal and the corresponding eigenspace is two-dimensional. In such a case, eigenvectors can not be (uniquely) determined: This is a singularity. Hence, we get the following definition.

Definition 3.2 A **degenerate point** of a 2D symmetric, second-order tensor field defined on an open subset of $\mathbb{R}^2$ is a point where both eigenvalues are equal.

It follows from the definition above, that a point is degenerate if and only if the corresponding tensor value is isotropic, that is of the form λI_2, where I_2 is the identity matrix. This second definition is used in practice to detect and locate degenerate points.

With first order precision, the possible types of degenerate points are the trisector point and the wedge points as shown in Fig. 2. Computing the tensor index of a closed curve around an isolated singularity, one gets $-\frac{1}{2}$ for a trisector point and $+\frac{1}{2}$ for a wedge point. (These half integers are due to the sign indeterminacy of eigenvectors.)

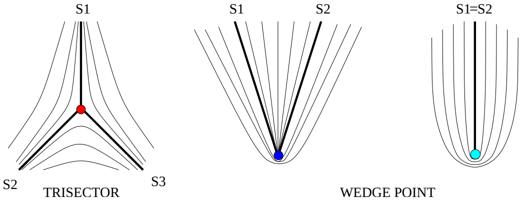

Figure 2: *First Order Degenerate Points*

3.3. Separatrices

A trisector point has three hyperbolic sectors (where integral curves lead away from the point in both directions) while a

wedge point has a hyperbolic sector and a parabolic sector (where integral curves lead away in one direction and toward the point in the other), possibly reduced to a single line. Separatrices are defined as the curves that bound hyperbolic sectors. When integrating these lines away from a singularity, one builds the edges of the topological graph.
The location of the separatrices in the vicinity of a degenerate point is based upon the following property ([1], p.120).

Property 3.1 With the notations of section 2, considering the first-order Taylor expansion of T in the vicinity of a degenerate point (x_0, y_0), the angles θ_k of possible separatrices s_k are given by the real roots of the following cubic polynomial

$$dz^3 + (c+2b)z^2 + (2a-d)z - c \ = \ 0$$

by inverting the relation $z_k = \tan\theta_k$.

The angles found in that way must be checked to correspond to an actual boundary curve of a hyperbolic sector. In the case of a degenerate point of index $+\frac{1}{2}$, solving this polynomial enables the distinction between a wedge point with a single separatrix and a wedge point with two separatrices and an actual parabolic sector.

4. Bifurcations

In the previous section, we have only considered instantaneous tensor fields. When time is introduced as additional parameter, degenerate points move in the plane and interact, appear or vanish. This induces structural changes for the topology, called *bifurcations*. Bifurcations occur for particular values of time and entail the instantaneous passage of the topology from a structurally stable state to another.
Unlike directional vector fields where bifurcations have been a major domain of interest to pure and applied mathematicians for a long time, the bifurcations of undirectionnal vector fields have apparently not received as much attention from mathematicians and a general theoretical framework seems to be less developed. For this reason, we will only describe two kinds of local bifurcations that may occur in the context of our method and let apart further discussion about all possible cases. In general, bifurcations that do not fall into one of the following categories will be detected by our method as well but their identification will be impossible (see section 7).

4.1. Pairwise Creation/Annihilation

Our first kind of bifurcation is the so-called pairwise annihilation of degeneracies. It occurs when two degenerate points of opposite index, that coexist close to another, merge. At this moment, a single degenerate point can be observed that is of no simple type and that disappears right after the fusion: No singularity exists from now on in the neighborhood. Reversing the direction of time, one gets similarly a pairwise creation of degeneracies: Starting with no singularity, there suddenly appears a degenerate point with complex structure. This instantaneous state is then replaced by two simple degenerate points that move away from another. Several configurations exist for the intermediate, unstable state. Fig. 3 shows two of them.

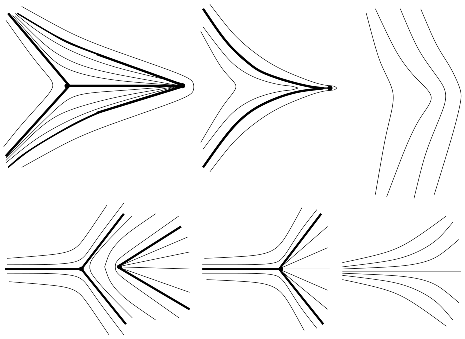

Figure 3: *Pairwise annihilation of degeneracies of opposite index*

4.2. Wedge Bifurcation

The other bifurcation that mostly occurs in our context is the transition from one type of wedge point to the other. The passage from a wedge point with two separatrices to a wedge point with one single separatrix means from a structural point of view the disappearance of the whole parabolic sector previously located between the two separatrices: This sector was 2-dimensional and becomes 1-dimensional, i.e. it is reduced to a single line. Inverting time direction, one observes the creation of a parabolic sector in the vicinity of a wedge point from an original single separatrix (see Fig. 4).

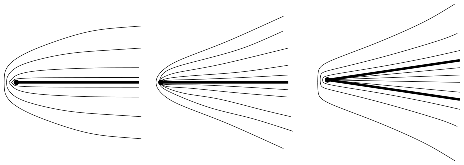

Figure 4: *Creation of a parabolic sector in the vicinity of a wedge point*

5. Data Structure

As mentioned previously, our method handles 2D, symmetric, second-order tensor fields. These tensor fields depend on time and are given in a discrete manner: For each time step, we get the tensor values at the vertices of a planar triangulation that remains constant over time (the positions are unchanged from one time step to the next). At each time step, the discrete information is piecewise linearly interpolated over the triangulation to result in an instantaneous planar C^0 tensor field. Now, the time steps must be embedded in a space/time continuum to enable the precise detection of structural transitions in the topology. This is described in the present section.

5.1. Space Time Grid

As in our work on time-dependent vector fields [7], the integration of the discrete data in a continuous mathematical space is achieved by considering time as third dimension and constructing a 3D structured grid, made of prism cells that connect together corresponding triangle cells lying in consecutive time steps as shown in Fig. 5. The prism cell has been preferred to a tetrahedrisation of the grid to reproduce better the topological continuity inside each triangle cell, tracked through time.

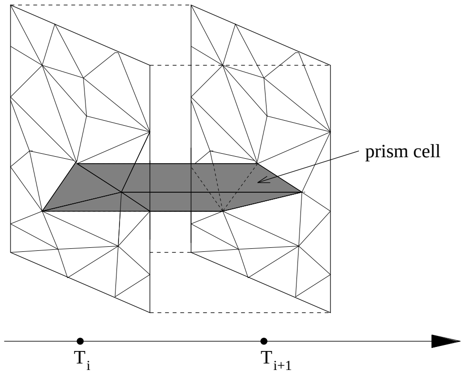

Figure 5: *Space/time grid connecting identical triangulations over time*

5.2. Interpolation

An interpolation scheme suited to this type of cell has already been used earlier [7]. It is designed to be consistent with the tensor field given at each time step. Basically, we proceed the interpolation of symmetric, second-order 2D tensor values (3 independent scalar components) over a 3D grid. More precisely, for each fixed time value, the interpolant must return a piecewise linear tensor field.

For a given prism cell lying between $t = t_i$ and $t = t_{i+1}$, let $f_j(x,y) = \alpha_j + \beta_j x + \gamma_j y,\ j \in \{i, i+1\}$ be the affin linear interpolants corresponding to a particular scalar component of the tensor field in the triangular faces of the prism lying in the planes $\{t = t_i\}$ and $\{t = t_{i+1}\}$, respectively. Then we define the interpolant over the whole prism cell by linear interpolation over time:

$$f(x,y,t) = \alpha(t) + \beta(t)x + \gamma(t)y$$

where

$$\alpha(t) = \frac{t_{i+1} - t}{t_{i+1} - t_i}\alpha_i + \frac{t - t_i}{t_{i+1} - t_i}\alpha_{i+1}$$

(idem for β and γ). This formula ensures obviously, for each fixed value t, that f_t is affin linear in x and y.

5.3. Singularity Location

For a particular prism cell, finding the locations of a singularity consists of determining the equation of the curve (or path) that a degenerate point describes in its motion over time. It also requires the precise detection, location and identification of the changes (or bifurcations) that may affect the nature of the singularity.

As said in section 3.2, locating a degenerate point is equivalent to finding a position in the plane where the tensor field is isotropic, that is of the form λI_2, which can be written as

$$\begin{cases} T_{11} - T_{22} & = & 0 \\ T_{12} (= T_{21}) & = & 0. \end{cases}$$

As we deal, for each (fixed) value of t, with a piecewise linear tensor field, this equation system is equivalent to the following linear system in x and y.

$$\begin{cases} (\alpha_{11}(t) - \alpha_{22}(t)) + (\beta_{11}(t) - \beta_{22}(t))x \ldots & & \\ \ldots + (\beta_{11}(t) - \beta_{22}(t))y & = & 0 \\ \alpha_{12}(t) + \beta_{12}(t)x + \gamma_{12}(t)y & = & 0 \end{cases}$$

Solving this system for x and y yields biquadratic rational functions in t. (Note that the linearity of the interpolant for any fixed t ensures the presence of a unique degenerate point for each value of t.) But this equation is only valid inside the cell, so we are interested in the sections of curve that intersect the interior domain of the cell. A practical and efficient way to achieve it, is to consider the singularities lying on the faces of the prism cell: 2 triangles (in the planes $t = t_i$ and $t = t_i + 1$) and 3 quadrilaterals (joining both planes together). Induced by the interpolant defined over the prism cell, these faces are linearly (resp. bilinearly) interpolated, which permits a fast singularity search. The positions found on the cell boundary are then associated pairwise to describe successive `entry` and `exit` positions of the path through the prism (see Fig. 6).

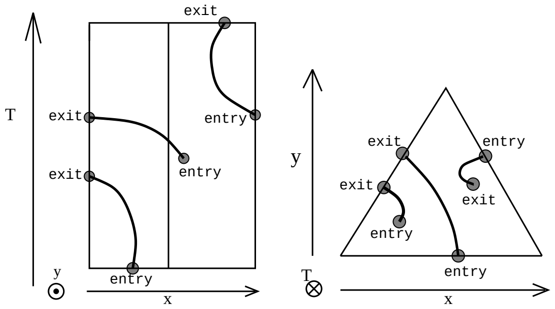

Figure 6: *entry and exit points of a path through a prism*

5.4. Bifurcation Detection

We must now consider the corresponding successive natures of this singularity to get a complete picture of its evolution through the cell. From section 3.2, we know that the degenerate points are of two major types (trisector and wedge).

Furthermore, we know that trisectors and wedges have opposite indices. Now, the tensor index of a closed curve is an invariant in continuous tensor fields, as is the Poincaré index for vector fields (this is intuitively clear but we have found no formal proof of this property so far). That means that if one takes the (triangle) cell boundary as closed curve to compute the tensor index, this index will be constant as long as no degenerate point reaches the boundary, i.e. as long as no degenerate point enters or leaves the prism cell. For our case, it entails that a wedge point remains a wedge point and that a trisector point remains a trisector point. That is the reason why, no type swap can occur in the context of our method inside a cell. The same result holds if the singularity leaves the cell to enter one of its neighbors by taking the boundary curve of both cells to compute the (constant) index.
Furthermore, as a linear tensor field can have at most one degenerate point, several singularities can not meet or split in the interior of a cell but only at the common boundary of two cells. This restricts the possible location of a pairwise creation/annihilation to the boundary of a prism cell. The last restriction concerns the merging (or splitting) of several degenerate points of the same type at the common edge of two cells. To show it, we first need the following property.

Property 5.1 If two simple degenerate points lie in two cells sharing a common edge, then they must have opposite indices.

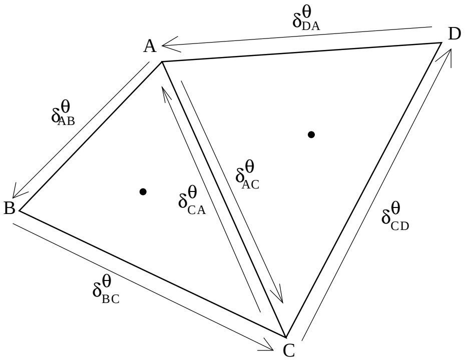

Figure 7: *Degenerate points in neighbor cells*

Proof: Consider figure 7: Let the points depicted in both cells correspond to degenerate points. By definition of the tensor index (see 3.1), computing the index along the edges of *ABC*, one gets

$$I_{ABC} = \delta^\theta_{AB} + \delta^\theta_{BC} + \delta^\theta_{CA}$$

where θ is the angle between the eigenvectors and the x-axis and δ^θ_{AB} is the change of θ along $[AB]$. Likewise, in *ACD*, one gets

$$I_{ACD} = \delta^\theta_{AC} + \delta^\theta_{CD} + \delta^\theta_{DA}$$

But of course, $\delta^\theta_{CA} = -\delta^\theta_{AC}$. Suppose for instance that $\delta^\theta_{CA} > 0$, then $\delta^\theta_{AC} < 0$. Now, we know that the angle changes along a closed curve encompassing a single isolated simple degenerate point occur always in the same direction ([1], Corollary 4, p. 119). That is, δ^θ_{AB} and δ^θ_{BC} must be both positive and δ^θ_{CD} and δ^θ_{DA} must be both negative. Consequently, we get in *ABC* a degenerate point with positive index and in *ACD*, we find a degenerate point with negative index which proves the property.

Consequently, the only bifurcation that can be observed in the interior of a prism cell is what we called a *wedge bifurcation* (see 4.2). The exact location of such a bifurcation is a difficult algebraic problem (one must find a value t_0 such that the cubic polynomial equation in 3.3 has exactly one real root for $t > t_0$ and 3 real roots for $t < t_0$, which is in our case a polynomial equation of degree 6!) that we solve numerically: We compute successive positions of the degenerate point according to the formula of 5.3 and check, in the case of a wedge point, if the number of its separatrices has changed. At this point, we start a binary search to approximate the exact bifurcation position.

6. Degeneracies' Tracking

The previous section described the preprocessing step of our method: Each prism cell is checked for a path of a degenerate point through it and the related `entry` and `exit` positions are saved as well as the wedge bifurcations on the way. The next task is to reconnect these pieces to obtain the whole path of each singularity together with its possible type changes. Furthermore, we still have to identify the bifurcations that occur on the side faces of the prism cells and that have not been detected so far.
The degeneracies' tracking algorithm processes two consecutive time steps at once (which avoids having all time steps loaded in memory). It starts by investigating all triangles lying in the first time step and tracks every found degenerate point through the corresponding prism cell until it either reaches the next time step or returns to the previous one. We give now a sketch of this forward tracking algorithm starting with time steps `TS(i)` and `TS(i+1)`. (The boolean array `ToCheck` indicates if a cell lying in `TS(i)` must still be checked. The boolean array `Reached` tells if a cell lying in `TS(i+1)` has been checked while forward tracking. At the beginning, one sets `for all j, ToCheck[j]=TRUE` and `Reached[j]=FALSE`.)

```
foreach PrismCell P(j)
 TrackingDirection = FORWARD
 if (ToCheck[j] &&
     TriangleFace in TS(i) contains
     Singularity)
  track singularity until it leaves P(j)
  get ExitPoint
  CurrentTime = time(ExitPoint)
  CurrentPoint = ExitPoint
  if (ExitPoint lies in TS(i+1))
   Reached[j] = TRUE
  else
   get next PrismCell P(l)                  :1
   track singularity until it leaves P(l)
   get ExitPoint
   if (time(ExitPoint) < CurrentTime &&
    trackingDirection == FORWARD)
    add PairwiseAnnihilation
        at CurrentPoint
    TrackingDirection = BACKWARD
   else if (timeExitPoint > CurrentTime &&
    trackingDirection == BACKWARD)
    add PairwiseCreation at CurrentPoint
    TrackingDirection = FORWARD
   endif
   CurrentTime = time(ExitPoint)
   CurrentPoint = ExitPoint
   if (ExitPoint lies in TS(i+1))
    Reached[l] = TRUE
   else if (ExitPoint lies in TS(i))
    ToCheck[l] = FALSE
   else
    goto :1
   endif
  endif
  ToCheck[j] = FALSE
 endif
endforeach
```

Once this has been done, we investigate the triangles lying in the next time step `TS(i+1)` with a backward tracking. Only cells j with `Reached[j]=FALSE` are considered. Otherwise, the triangles have been processed already. The code of this backward tracking is very similar to the one of the forward tracking. The transition to the next pair of time steps is done by substituting `ToCheck=Reached` and setting `Reached=FALSE`. At the end of this scheme, all pieces of degeneracies' paths lying between consecutive time steps are reconnected together (through time) with the corresponding bifurcations that have been encountered on the way.

7. Topology Depiction

Now, as degeneracies are only part of the topology in the 2D steady case, the paths we have tracked over time must be associated with the corresponding separatrices to depict the whole topological structure of the tensor field in the unsteady case. Practically, the separatrix curves change over time and thus span separatrix surfaces that bound volumes of the time-dependent topology.

7.1. Structures Tracking

Drawing these surfaces assumes the previous determination of each singular path and, for each discrete position along the path the computation, of the start directions of the separatrices (see 3.3). In our method, these directions are computed during the path tracking inside each prism (see 5.3). The depiction of the separatrix surfaces is then based upon the integration of separatrix curves along this direction on one hand, and on the correlation of each separatrix curve with another curve at the previous position along the path on the other hand. This correlation is needed if one deals with a trisector path or a wedge path with two separatrices to avoid binding curves that do not fit in a topological sense. This is achieved by looking for the best approximation of current separatrix angle among the previous ones.
The integration itself is done numerically by a 4th order Runge-Kutta integration with adaptive step size. Computing integral curves in tensor fields is a challenging issue for numerics since eigenvectors have neither norm nor orientation. A stop criterion must therefore be based upon topological information to ensure proper treatment of the degenerate points. The computation can end in two ways: Either the integral lines reach the domain boundary or they converge towards a wedge point inside its parabolic sector (that is, integral curves can only converge towards a wedge point with two distinct separatrices). We thus consider for a given curve, that topological consistency has been maintained with its predecessor if they both leave the grid at close positions or both converge towards the same wedge point. In other cases, consistency has been lost and we infer the presence of a global bifurcation (that we can not identify more precisely currently) if no local bifurcation was found at this position (see 4).

7.2. Surface Drawing

Once separatrices have been associated over time, one gets a set of curves that span a surface. The construction of this surface is done in two steps. Originally, the curves that result from numerical integration are piecewise linear. Therefore, they are first replaced by interpolating NURBS curves of degree 2 to increase smoothness. In the second step, these NURBS curves are embedded in a NURBS surface, after uniformization of their parametrisations.

8. Results

We present here the results of our method applied to two datasets: The first one is an artificial symmetric tensor field that we use to demonstrate the different aspects of the algorithm and to illustrate the topological concepts introduced previously. The second one is provided by a CFD simulation of a turbulent tensor field and gives us the opportunity to comment the application of this technique for the visualization of real, complicated CFD data.

8.1. Artificial Tensor Field

Our first dataset is genenerated on a small rectilinear point set of 25 vertices that we triangulate to get a grid with 32 cells. We start with a topology containing 3 trisector points and 4 wedge points (3 with a single separatrix and 1 with two

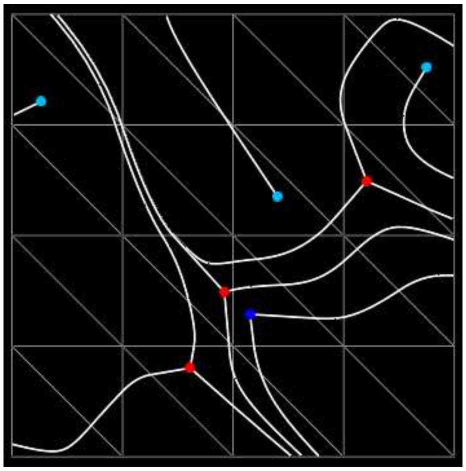

Figure 8: *Start Topology*

separatrices) as shown in Fig. 8. We process 11 consecutive time steps.

Tracking the degeneracies over time, we get the picture shown in Fig. 9. Color coding is as follows: Trisectors are depicted in red, wedge points with two separatrices in dark blue and wedges with a single separatrix are shown in light blue. Bifurcations are marked as small balls: green ones indicate pairwise creation, red ones illustrate pairwise annihilation and yellow balls designate wedge swap bifurcations. Five Bifurcations have been detected: two pairwise creations, two pairwise annihilations and one wedge swap.

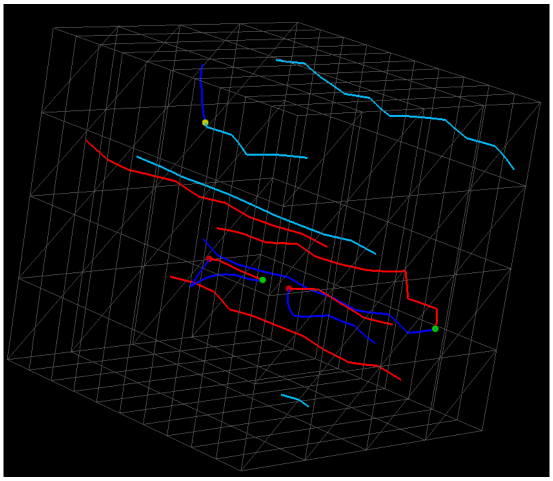

Figure 9: *Degeneracies' paths*

Adding the separatrix surfaces to the degeneracies' paths, we complete the topology depiction and obtain the structures presented in Fig. 10* (separatrix surfaces emanating from a trisector point are colored in red while those coming from a wedge point are displayed in blue).

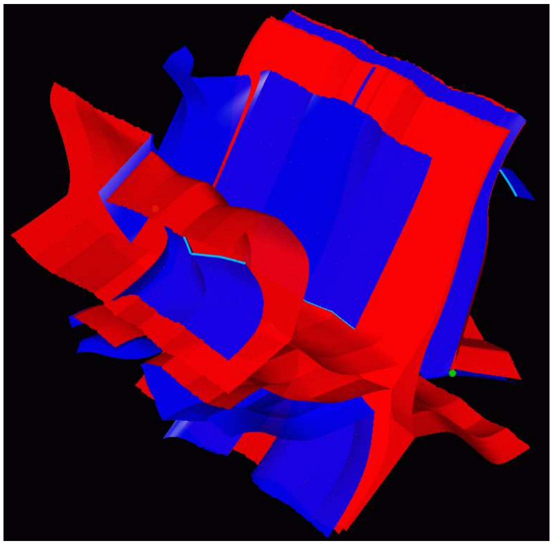

Figure 10: *Picture of the complete topology evolution (see also Fig.17)*

If we cut instantaneous time planes out of this three-dimensional picture, we can exhibit discrete samples around a bifurcation point. As an example, we depict in Fig. 11 part of the topology before, during and after a pairwise creation. A wedge bifurcation is presented in Fig. 16.

8.2. Rate of Deformation Tensor Field

Our second dataset stems from a CFD simulation. This is the rate of deformation tensor field of a swirling jet simulation provided by Wolfgang Kollmann from UC Davis. The complete structured grid has 250 x 244 cells. We consider the symmetric part of the original data. To avoid visual clutter, we process a cell group containing 45 x 60 cells as shown on Fig. 12.

Processing this set of cells over 25 time steps (see Fig. 13 to get an impression of the 3D structure), we obtain the degeneracies' paths shown in Fig. 14. We have reduced the size of the balls indicating bifurcations to avoid confusing the depiction. As a matter of fact, as one can see on this picture, tracking the topology of this turbulent tensor field results in a large number of encountered bifurcations. This effect is probably related in some extent to the linear scheme used for time interpolation. If one zooms into the data to look at small topological features, one observes structures like the one shown in Fig. 15. The presence of such structures suggests that a post-processing step would be helpful to simplify the resulting topology by suppressing features of small time or space scale. Nevertheless, structures of larger scale may

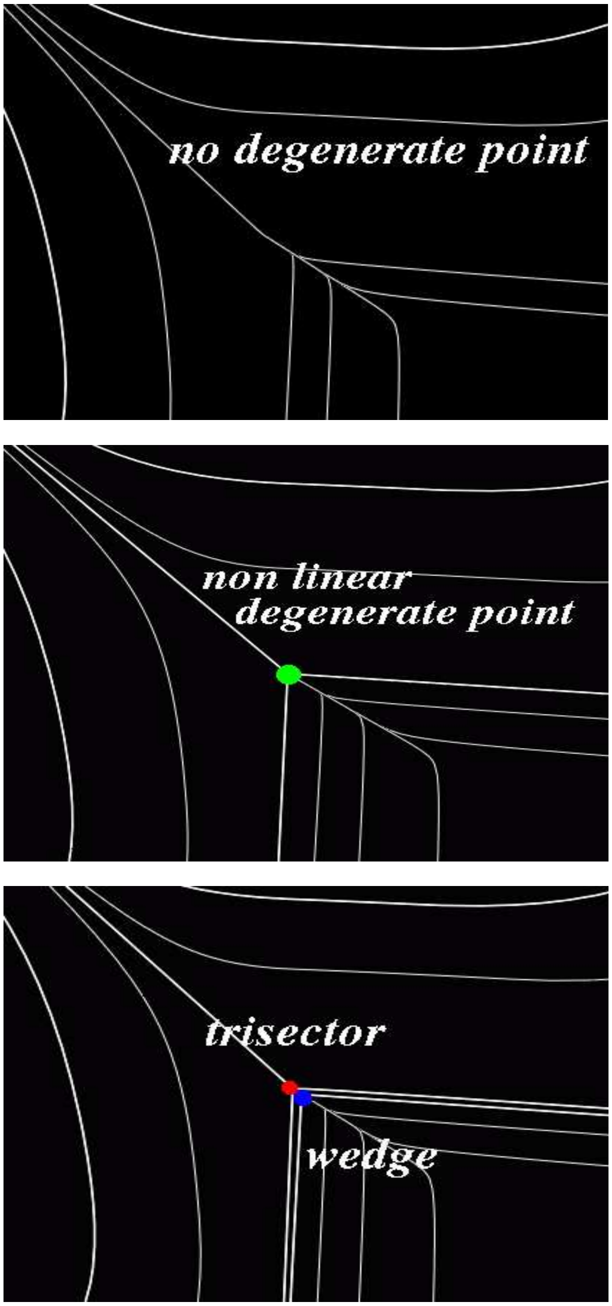

Figure 11: *Topology in the neighborhood of a pairwise creation*

be easily recognized in the global picture because of their temporal persistence.

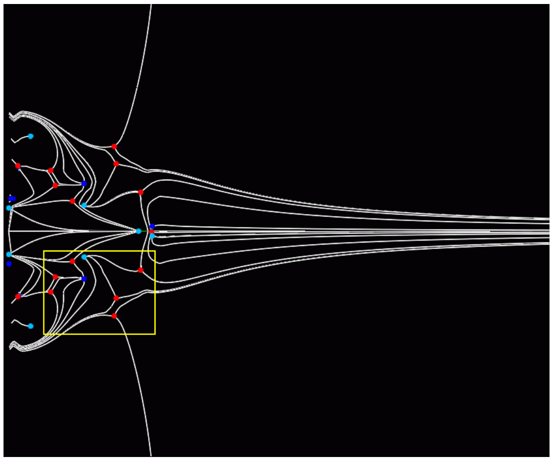

Figure 12: *Start topology and considered part*

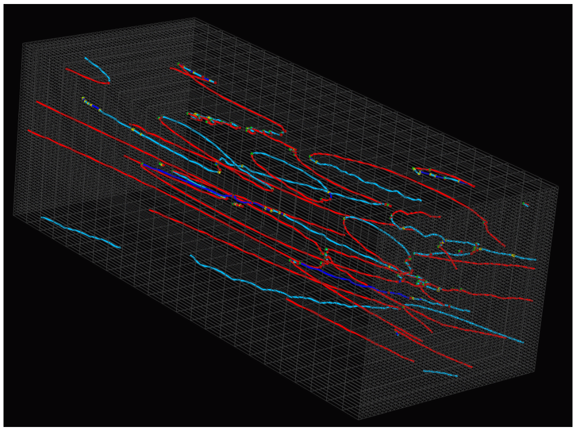

Figure 13: *Degeneracies' paths and 3D grid*

9. Conclusions

We have presented a topology-based method for the visualization of time-dependent, two dimensional, symmetric, second-order tensor fields. Degenerate points - depicted by conventional topology methods for steady fields - are tracked over time as well as the related structures. An essential aspect of this task is to find and identify the bifurcations that occur on the way and affect the position, nature and number of degenerate points. Their detection is made possible by the use of a temporal interpolation that embeds the given discrete time steps in a space/time continuum. The method results in three-dimensional pictures where topology is visualized as degeneracies' paths and separatrix surfaces. We think that this technique provides an efficient way to understand and interpret the temporal evolution of time-dependent tensor fields that could be completed by a post-processing step to suppress interpolation artifacts.

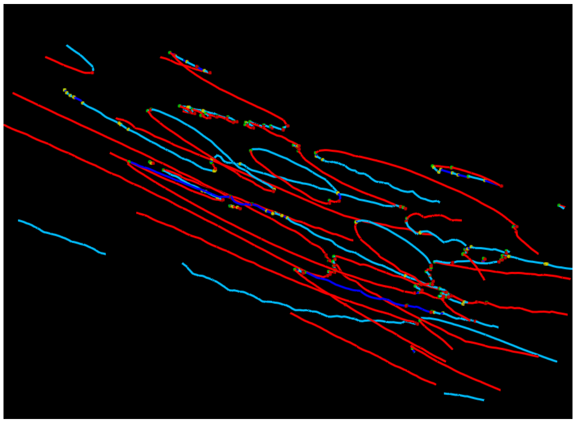

Figure 14: *Degeneracies' paths*

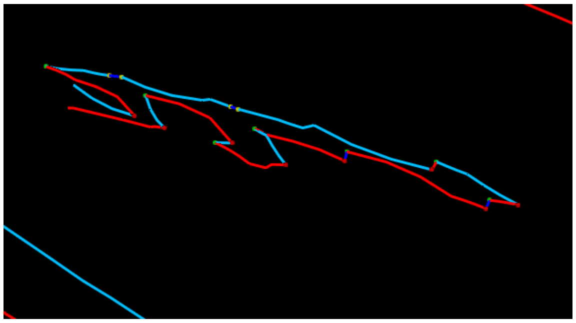

Figure 15: *Local topological structure*

Acknowledgements

The authors wish to thank Wolfgang Kollmann, MAE Department of the University of California at Davis, for providing the rate of deformation tensor dataset.

References

1. Delmarcelle T., *The Visualization of Second-Order Tensor Fields.* PhD Thesis, Stanford University, 1994. 1, 2, 3, 5

2. Delmarcelle T., Hesselink L., *The Topology of Symmetric, Second-Order Tensor Fields.* Proceedings IEEE Visualization '94, 1994. 2

3. Helman J.L., Hesselink L., *Visualizing Vector Field Topology in Fluid Flows*. IEEE Computer Graphics and Applications, 1991. 1

4. Tricoche X., Scheuermann G., Hagen H., Clauss S., *Scaling the Topology of 2D symmetric, Second-Order Tensor Fields* In NSF/DoE Lake Tahoe Workshop on Hierarchical Methods for Scientific Visualization, October 2000. 1

5. Lavin Y., Levy Y., Hesselink L., *Singularities in Nonuniform Tensor Fields* Proceedings IEEE Visualization '97, 1997. 1

6. Andronov A.A., Leontovich E.A., Gordon I.I., Maier A.G., *Qualitative Theory of Second-Order Dynamic Systems* Israel Program For Scientific Translation, Halsted Press, 1973. 2

7. Tricoche X., Scheuermann G., Hagen H., *Topology-Based Visualization of Time-Dependent 2D Vector Fields* To appear in Eurographics - IEEE TVCG VisSym '01, May 28 - May 30, 2001, Ascona, Switzerland. 1, 4

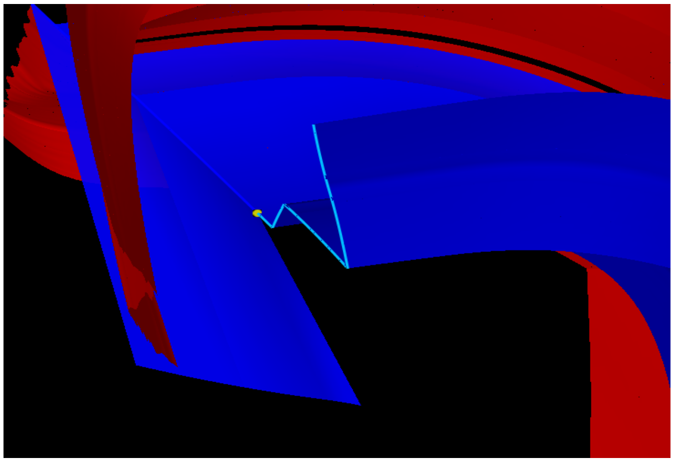

Figure 16: *A wedge bifurcation*

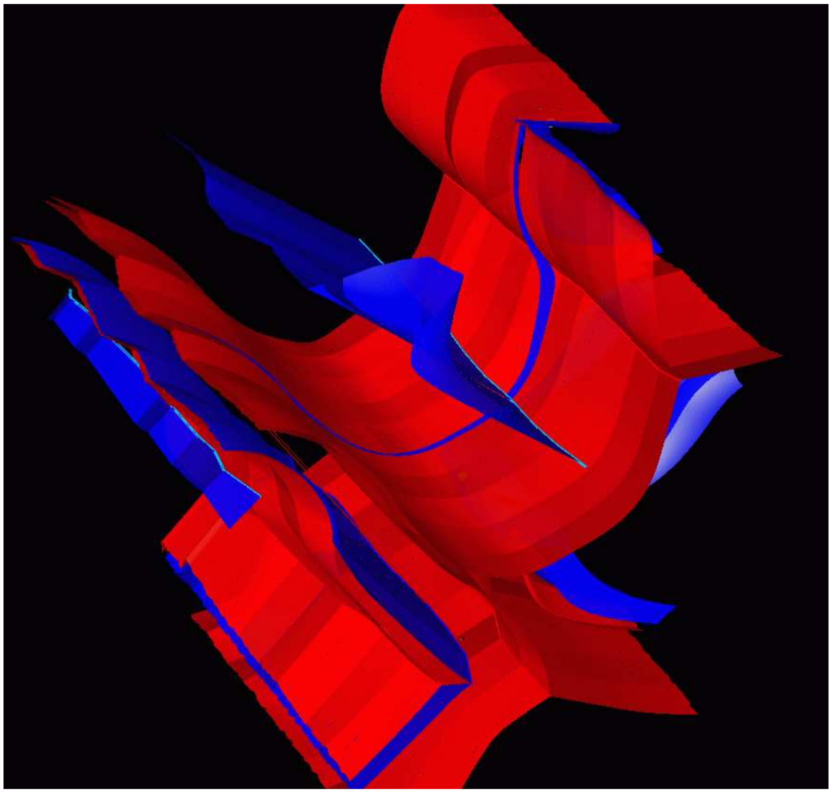

Figure 17: *Complete topology in another perspective than Fig. 10*