

Accurate and Efficient Visualization of Flow Structures in a Delta Wing Simulation

Xavier Tricoche, Christoph Garth, Tom Bobach, Geric Scheuermann

Visualization Group, Department of Computer Science, University of Kaiserslautern

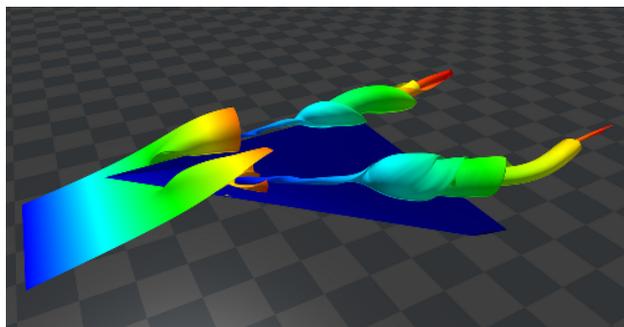
P.O. Box 3049, 67653 Kaiserslautern, Germany

Markus Rütten

Institute for Aerodynamics and Flow Technology, German Aerospace Center

Bunsenstr. 10, 37073 Göttingen, Germany

The paper is concerned with the extraction and visualization of structural features in delta wing dataset resulting from a large-scale Reynolds Averaged Navier-Stokes simulation. Its main contributions are two novel schemes designed for this purpose. First, we introduce a new method for fast and precise extraction of separation and attachment lines on the body of the delta wing. Second, we improve significantly standard stream surface computation schemes to achieve accurate integration even in regions exhibiting complex structures. Additional state-of-the-art flow visualization techniques are used to complete the study of the dataset. The main focus is on the interaction between open separation and attachment lines of the wall shear stress on one hand, and vortices that form in the vicinity of the delta wing on the other hand. These features are key factors in flight stability and their study is mandatory in aircraft design. The paper demonstrates the power of a combined use of line-type feature extraction and stream surface computation to deeply improve understanding of three-dimensional structures and ease the analysis of flow separation and vortex breakdown.



I. Introduction

In both the civil and military fields, the demand for shorter flight times and faster aircrafts has been a driving force behind research in recent years. Although it is not a recent development, together with supersonic speeds becoming more attractive, the delta wing design has found its way back into aircraft construction, as is demonstrated by a number of military aircrafts and the transatlantic passenger jet Concorde. An increased perception of security and the ever-shortening take-off and landing intervals on modern airports mandate a thorough examination of delta wing configurations with the aim of controllable flight even in exceptional flight situations, e.g. at a high angle of attack at subsonic speed. Furthermore, in military airflight, exceptional maneuverability is of prime importance. Due to this, the understanding of flow

phenomena related to delta wing setups has become a major point of research activity. Among the most interesting of these is *vortex breakdown* due to its severe impact on flight stability.

On the practical side of research, numerical simulations in Computational Fluid Dynamics (CFD) provide engineers with increasing amounts of multivariate data for analysis and evaluation. In the field of aircraft design, state-of-the-art calculations produce large-scale scalar and vector fields defined on unstructured grids (e.g. pressure, helicity, velocity and vorticity) which are typically very large (currently up to 50 million points) over a large number of timesteps. Hence, fast algorithms are needed for analysis and only some form of automatism can make the examination of many timesteps feasible. The mission of scientific visualization techniques in this context is to offer effective and efficient means of extracting the essential properties of the flow from the sets of raw data. The classical approach is to focus the visualization on features of interest that most analysts consider essential for both scientific and industrial applications. Prominent examples of such features are vortex cores, shock waves, and separation or attachment lines. For each type of feature, specific methods have been designed over the last decades to locate, characterize and depict subsets of the original volume. They are usually based on empiric definitions of the structures to be visualized that are fitted to the considered application. This leads to several ad-hoc and possibly contradictory definitions of the features, due to the lack of formal definition for the natural and intuitive concepts behind them, like vortices for instance. For the visualization of a particular dataset engineers are confronted with the problem of experimentally finding which of the existing methods will perform best for the given problem. Restrictions on the type of method might be induced by the size or the structure of the data. It is the focus of this paper to present a collection of well-established as well as new and improved methods and algorithms that can aid the analysis of a single timestep of a delta wing data set. Moreover, these methods may prove useful in adapting the CFD computations to give more accurate results.

The structure of the paper is organized as follows. The delta wing data set is presented more closely in section II. The corresponding numerical simulation is briefly described along with the resulting grid structure and the associated multivariate fields. We also discuss the preprocessing steps required for visualization purposes. In section III, we give a critical overview of existing methods for the analysis of both 2D and 3D parts of the dataset. Section IV is concerned with the description of a new method for the extraction of attachment and separation lines on the surface of the delta wing. We then introduce an improved scheme for stream surface computation in section V. Next, we present visualization results in section VI, mainly focusing on the shear stress vector field defined on the surface of the delta wing and stream surfaces that show the flow behavior in interesting parts of the dataset, e.g. vortex genesis and breakdown as well as flow separation. Finally, we conclude and discuss future work in section VII.

II. Delta Wing Dataset

A. Presentation

In this paper, we study a viscous Navier-Stokes simulation of airflow around a sharp-edged delta wing at high angle of attack at subsonic speed of Mach 0.2 with a Reynolds number of 1.2 million. The angle of attack increases with time, resulting in vortex breakdown above the wing in later timesteps. The simulation of the full configuration without the assumption of symmetry is based on a Reynolds Averaged Navier-Stokes (RANS) solver with a $k-\omega$ turbulence model. The computation was carried out on a Linux cluster using the DLR Tau Code solver. The grid consists of 3 million unstructured points forming 11.1 million unstructured elements, 8.8 million tetrahedra and 3 million prisms. The simulation data is stored at the grid points and contains density, pressure, kinetic energy, eddy viscosity and velocity. Among the significant physical features are secondary and tertiary vortical structures and corresponding separation and attachment structures in the shear stress field.

Aside from the full volume dataset, we are also interested in the structure of the two-dimensional flow on the wing that permits insights into the three-dimensional separation behavior of the flow above the wing. Although in this dataset a no-slip boundary condition was employed for the wing surface, analysis of the wing shear stress, treated as a two-dimensional flow field, is performed. The wing is given as the triangulated inner boundary of the volumetric flow field and consists of roughly 63k points forming 127k cells. The pointwise shear stress is computed from the volumetric flow field.

While the original dataset consists of 900 timesteps, this paper deals with analyzing a single timestep as opposed to the full sequence. In our opinion, examination of time-dependent simulations has to rely in part on efficient processing of individual timesteps. Hence the presented techniques may serve as building blocks

in the study of time-dependent data.

Earlier, similar simulations were computed using three-dimensional Euler equations on structured grids, mostly finite volume or finite difference methods, which lacked the reproduction of secondary vortices and thereby attachment and separation features. (cf.⁴). However, primary vortices were reproduced reliably. Following a period where typical variations of viscous Navier-Stokes computations prevailed (see e.g.⁶), currently unstructured hybrid methods become available together with the possibility of mesh refinement, resulting in datasets with up to 10 million points, together with enhancements for the more accurate simulation of complicated structures, e.g. Large Eddy Simulation (LES) or Detached Eddy Simulation (DES).¹³ Here it is important to mention that these techniques can be applied more reliably if it is possible to determine areas of interest during the simulation and adapt the mesh according to the detected features.¹² Hence, reliable detection of features of interest is of major importance not only in the context of post-computation analysis but also for the simulation process itself.

The major task for the analysis and visualization is a detection and examination of fluid flow structures like vortex cores, separation lines, attachment lines and vortex breakdown which includes an analysis of the three-dimensional fields as well as an examination of pressure and wall shear stress distribution on the wing. Together with good visualization, a deeper understanding of the relevant flow structures can be provided. In the focus of this paper are especially vortices and vortex breakdown together with the related attachment and separation structures on the wing in the described dataset.

B. Preprocessing

As we will see next, the visualization methods to come imply the following requirements.

- The interpolation of a scalar or vector field at an arbitrary position within the grid must be accurate and fast. Therefore, fast cell location of any 3D position is necessary.
- The fields must be provided along with their derivatives. These are not contained in the original data set and hence must be computed.
- A smoothing of the original data might be necessary to avoid unpleasant artifacts in the visualization results (e.g. numerical noise emphasized by the derivative computation). However, the smoothing step must be carried out carefully to preserve significant flow features.

Moreover, since the whole visualization process is performed on standard PCs, interactivity induces a need for efficient schemes, both in computational time and memory load. The solutions used are discussed next.

CELL LOCATION To permit fast cell location of an arbitrary position within a unstructured grid of about 6 million cells, we use an efficient data structure presented recently.¹¹ In essence, this structure is build upon the combination of a kD-tree based on the grid vertices and cell adjacency information. For every query, the given position is located within the leaves of the kD-tree. Each leaf contains a reference to a grid cell which allows us to start an iterative cell-wise path through the grid toward the position. This search structure is joined by robust numerical methods to handle cells with degenerate geometry.

DERIVATIVE COMPUTATION Since the grid under consideration is unstructured, commonly used derivative computation based on finite differences is impossible here. Practically, the methods presented in the following require the computation of the Jacobian matrix at each grid vertex. Observe that this computation must be carried out both in the 3D flow surrounding the delta wing and in the 2D shear stress vector field defined on the wing body. Derivative computation must be done very carefully since it typically amplifies noise, leading to meaningless results in further processing. A natural but insufficient solution is to apply a least squares fit to the vertices located in the 1-neighborhood of the considered position. This performs satisfyingly for 3D data but gives very poor results when applied to the shear stress defined over the polygonal surface of the delta wing. This is mainly due to the tangent plane indeterminacy at each vertex of such a triangulation. To overcome this problem, we adopted an alternative approach, which turned out to be extremely fast and robust to noise. We assume that every cell around a vertex is a simplex. This condition is not fulfilled by the 3D grid but an implicit tetrahedrization can be carried out on the fly. Each simplex is linearly interpolated which corresponds to a cell-wise constant value of the Jacobian. To obtain the point-wise value at a particular vertex we compute a weighted combination of the cell-wise values associated with the cells that directly surround it. We found an angle-based weighting strategy to give good results, especially in regions with inhomogeneous cell sizes. Remark that this scheme shows a smoothing effect.

LOCAL SMOOTHING For smoothing of grid-sampled data, a number of methods are readily available, e.g. exact numerical diffusion by means of a finite element method. It is however impractical to apply such methods to the dataset under consideration due to its complexity and size. Hence we apply an approximated scheme.

We note that diffusion-type smoothing methods on scalar fields can be described with the aid of the Laplacian operator, hence we require a suitable discretization that can be applied to both three-dimensional grids as well as 2d grids in three-space. Taubin introduces the point-based umbrella operator on the one-neighborhood of a grid point in analogy to the centered-difference approach on rectangular grids.²¹ Some care has to be taken to ensure convex-combination properties of this operator, but the ensuing implementation is straightforward and fast. Applying an explicit discretization scheme for the diffusion with multiple passes as needed yields a fast and easily implemented algorithm. Note that in smoothing vector data it is uncommon to diffuse componentwise. However, since the emphasis here is on removing numerical noise from the data, this method delivers the desired accuracy at good computational speed.

III. Flow Visualization

Before presenting our new methods, we briefly review in the following state-of-the-art flow visualization techniques directly related to the present work. More precisely, our focus is on line-type feature extraction, texture-based flow visualization and stream surface computation.

A. Extraction of Line-Type Features

We are interested in two main types of line features: vortex core lines and separation/attachment lines. While the former indicate the center of swirling flow structures, the latter represent lines on a surface where the flow separates or attaches and does not follow the surface of an embedded object. Both vortices and regions of separation are of essential interest in aerodynamics due to the severe effects on lift and drag behavior. We first introduce a convenient mathematical description called *Parallel Operator* that permits a unified presentation of existing techniques.

THE PARALLEL OPERATOR As introduced by Peikert and Roth,¹⁴ the parallel operator \parallel compares two vector fields v and w on a domain Ω , and delivers the set of points where they are parallel, that is where their cross product is zero. Hence, it is a union of zero-isolines and can be extracted with suitable methods. Intrinsically, these isolines are closed or end only at domain boundaries and do not intersect each other. The algorithmic realization of the isoline computation looks for the zeros of $v \times w$ on each grid face by performing a two-dimensional Newton iteration, starting at the face center. An analytical method is available for simplicial elements (derived in¹⁴). After the computation, it is natural to connect straight line segments from zeros on faces belonging to the same cell to achieve a consistent polygonal description of the isoline.

Using the *parallel operator* we now turn to the presentation of standard algorithms for the extraction of vortex core lines and separation and attachment lines.

VORTEX CORE LINES Sujudi and Haimes describe an *eigenvector method* for finding vortex cores in flow fields based on tetrahedral grids with linear interpolation.²⁰ Basically, their scheme looks for linear swirling patterns on a cell by cell basis. Practically, in every tetrahedron T , the constant Jacobian J of the flow field v is computed and checked for one real eigenvalue λ_0 and a pair of conjugate-complex eigenvalues λ_1 and $\lambda_2 = \bar{\lambda}_1$. In this case, the reduced velocity is computed at the vertices of T by subtracting the eigenvector v_0 corresponding to λ_0 from the original velocity. Then, the zeros of the reduced velocity are computed on T 's faces and are successively connected to form the straight line segment of zero velocity. The union of all such line segments makes up all the detected vortex cores in the flow field. From practical experience, the method is able to detect continuous vortex cores reasonably well.¹⁷ A drawback in this direct algorithm is that it does not deliver connected line segments between neighboring tetrahedra since the calculation is performed independently on each tetrahedron and the Jacobian is discontinuous on cell boundaries.

Roth manages to reformulate this algorithm by observing that the points of zero reduced velocity are the points where \vec{v} is parallel to the real eigenvector \vec{v}_0 , thereby making the vortex core definition of Sujudi and Haimes appear as an application of the parallel operator.¹⁷ In fact, the vector fields to be checked for parallelity are the velocity \vec{v} and the acceleration of the flow \vec{a} :

$$\vec{v} \times \vec{a} \equiv \vec{v} \times J \cdot \vec{v} = \vec{0}. \quad (1)$$

In other words, vortex core lines are characterized as streamlines with zero curvature. Since the parallel test can be done on cell faces, the resulting feature lines are connected. Moreover this technique is not restricted to simplicial grids. Observe however that false positives still must be discarded in a post-processing step which might be a difficult task.¹⁷ Nevertheless, we show in the following that this algorithm works well for the data at hand.

SEPARATION AND ATTACHMENT LINES Following the idea of Sujudi and Haimes, Kenwright proposed a simple and fast method for the extraction of separation and attachment lines.⁸ His basic observation is that these feature lines exist for two linear patterns, namely saddle points and proper nodes where they are aligned with an eigenvector of the Jacobian. The original method works cell-wise and results in disconnected line segments, due to the discontinuity of the Jacobian. However, applying the parallel operator leads again to Equation 1 which corresponds to the 0-isoline of a scalar field in the 2D case.¹⁰ This reformulation permits to obtain continuous lines too. However this definition is quite restrictive because it assumes that separation and attachment lines always have zero curvature. Moreover, since it is based on a point-wise analysis of the Eigensystem it is very sensitive to noise. Consequently it requires strong pre-smoothing of the data which in turn can deform and shift the features. We discuss and illustrate these problems in the presentation of our new method in section IV.

B. Texture-based Flow Visualization

A very appealing way to depict global structures of a planar flow is to use a texture-based approach. Among existing techniques, the most popular is LIC.^{2,19} The basic principle consists in smoothing a white noise texture along the flow direction, achieving a strong correlation along individual streamlines, and no correlation orthogonal to them. The resulting depictions surprisingly mimic the oil patterns known from wind tunnel experiments and provide a very intuitive understanding of the flow behavior. This technique was originally designed for planar vector fields. Extensions to two-dimensional manifolds either assume a global parameterization of the surface⁵ or require the geometry to be rather simple to allow cell-wise texture mapping.¹ For the delta wing data set however, we use the simplicity of the surface geometry to derive a global parameterization.

C. Stream Surface Computation

For steady vector fields, stream surfaces can be considered the natural extension of streamlines to the 3D case. However, there exist only few methods to compute stream surfaces for visualization purposes. Hultquist pioneered research in this field with an advancing front algorithm.⁷ To handle converging and diverging behavior of the flow he introduces adaptive resolution during integration. Streamline integration is based on a second-order numerical scheme with constant step size which enables a simple on-the-fly triangulation of the surface. This scheme is fast and straightforward to implement, and performs well for simple vector fields but it is unsuited for complex flow behaviors like vortical motion or flow recirculation, present in the delta wing data set. We show in section V how to improve this technique to handle such data. Other approaches exist. van Wijk proposed an implicit technique that reduces stream surface computation to isosurface extraction, which permits to use fast and reliable techniques.²⁴ The results however are strongly dependent on the choice of a convenient scalar field to be advected through the grid. The starting curves are given by isolines of the 2D scalar field. Adopting the advancing front idea from Hultquist, Scheuermann et al.¹⁸ exploit certain properties of triangulated grids with linear interpolation, and in particular the existence of closed formula for streamlines inside each tetrahedron. The major drawback of this method is its high computational cost when dealing with large-scale grids. Furthermore, it is unable to process arbitrary grid structures.

IV. Efficient and Robust Extraction of Separation and Attachment Lines

As mentioned previously, the standard scheme of Kenwright for extracting separation and attachment lines turned out to provide rather coarse results when applied to the delta wing data set. More precisely, we experienced following problems.

- Point-wise eigensystem analysis strongly depends on the smoothness of the underlying data. The results obtained without pre-smoothing of the shear stress vector field have very poor quality and

cannot be used for further processing. It is worth mentioning here that the very simple geometry of the delta wing actually permits a quite stable Jacobian computation and should lead to good results which was not observed in practice.

- Smoothing the data to reduce the impact of noise is likely to shift or even destroy the feature lines.
- Many so-called false positives are produced that must be carefully discarded in a post-processing step. Feature length and flow parallelity are typical criteria. The corresponding thresholds must be determined in accordance with the data which requires user interaction.

These deficiencies are illustrated in Fig. 1 and Fig. 2. Color coding: separation (red), attachment (blue), actual separation line (green).

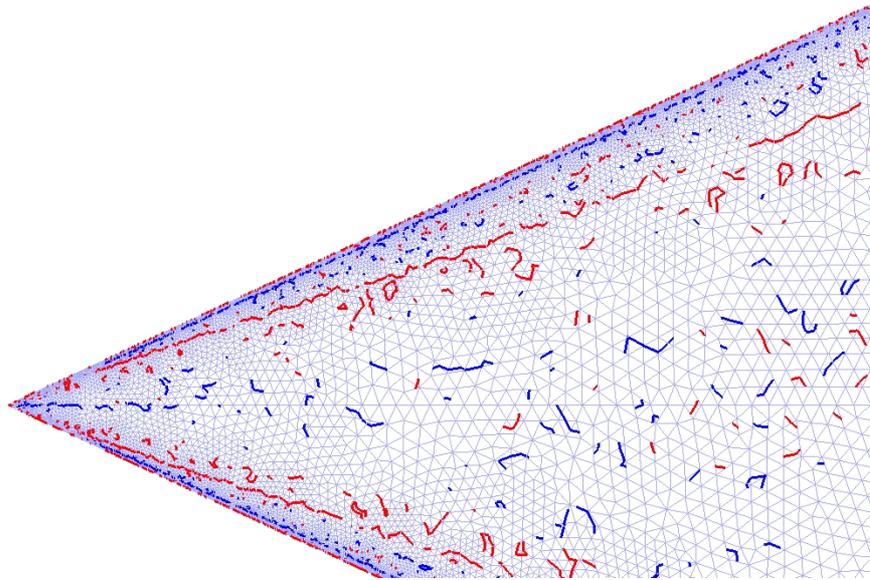


Figure 1. Kenwright's method without pre-smoothing

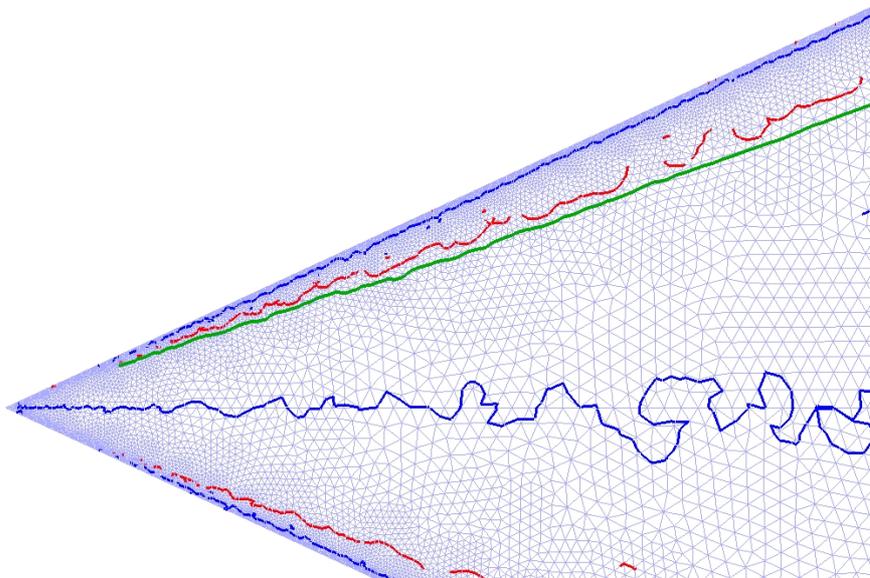


Figure 2. Kenwright's method after pre-smoothing

Our new scheme is based on the following observation: the spatial distribution of the shear stress divergence is closely related to the position and geometry of the separation and attachment lines. However, since this quantity is obtained by derivative computation, the restrictions mentioned about Kenwright’s method apply to any feature extraction based directly on divergence. Yet, we obtain large connected domains of positive (resp. negative) divergence that provide a first approximation of separation and attachment lines. The necessary correction toward actual feature lines can then be achieved by streamline integration. Practically the structure of our algorithm is as follows:

1. Compute at each grid vertex the divergence of shear stress vector field
2. Extract ridge and valley lines of divergence scalar field.
3. Starting along each ridge and valley lines compute in a LIC-like manner a cell-wise scalar field that accounts for the number of streamlines that crossed each cell.
4. Extract ridge lines of this new scalar field.
5. Draw streamline along each line to obtain separation and attachment lines.

We detail next the successive steps of our method. Corresponding results are shown in Fig. 6.

DIVERGENCE COMPUTATION For continuous vector fields the divergence at a point P is defined as the amount of flow generated in an infinitesimal region around P . In a Cartesian basis, it is given by the expression $\text{div}\vec{v} = \frac{\partial}{\partial x}v_x + \frac{\partial}{\partial y}v_y$. Practically we compute it at each grid vertex as the trace of the Jacobian matrix, see section B. Remark that the fact that the divergence distribution is only interpreted as a “first guess” in further processing allows us to smooth out local oscillations from the divergence.

RIDGE AND VALLEY LINE EXTRACTION For a scalar field interpreted as a height field, a ridge or valley line is defined as the set of points where the slope is locally minimal compared to points of the same elevation. Ridge and valley line extraction is a classical task in image processing. In this case, these lines are interpreted as edges in a scalar picture. The existing methods in this context are however of little help for our problem since they typically assume structured grids and require first and second order derivative computation.¹⁷ This cannot be done satisfyingly on a scalar field obtained itself by derivation. For this reason, we adopted an alternative, much easier approach that is fast and gives good results.

We reformulate the definition of ridge and valley lines as follows. Ridge (resp. valley) lines are curves starting at local maxima (resp. minima) and proceeded so as to minimize descent (resp. rise). The corresponding algorithm thus starts at vertices corresponding to local maxima (resp. minima) and proceeds ridge (resp. valley) line extraction toward the direct neighbor with maximum (resp. minimum) value. Nevertheless, since the data at hand is typically noisy (see above), some improvements are necessary to use this method in practice. First, we want to restrict ridge line extraction to major features and discard minor ones. So, we need to discard local extrema due to high frequency oscillations. Practically, we restrict our considerations to values that are extrema within a large neighborhood around them. Second, we want the extracted feature lines to be as straight as possible and to avoid u-turns and self-intersections. Therefore we impose an angle criteria to every new segment, driven by the mean direction followed during the last few steps. Moreover, we exclude from further processing every vertex that is a direct neighbor of a vertex selected previously. Modified in this way, the algorithm is very fast, straightforward to implement, though robust to noise.

CONVERGENCE MONITORING We expect the ridge and valley lines to lie close to the separation and attachment lines. Now we need a correction that provides the actual line position. Motivated by the empiric characterization of separation and attachment lines, we chose to measure the flow convergence from the ridge and valley lines on a cell-wise basis. Observe that ridge lines (positive values of the divergence) are associated with backward convergence (attachment) whereas valley lines (negative values) are related to forward convergence (separation). To monitor this convergence we use an approach somehow similar to the LIC technique:² we count for each grid cell the number of streamlines that were integrated through it. Since we expect the starting locations to lie close to feature lines, we only need to integrate streamlines along a short arc length to highlight the converging behavior. The cell-wise scalar field obtained is converted to a point-wise field with the technique discussed in section B to serve as input for edge detection. Remark that, compared to a classical LIC picture, we both reduce the complexity by shortening the streamlines’ length and by restricting the starting locations to 1D subsets. This speeds up computation by several order of magnitude.

FEATURE LINE INTEGRATION The ridge and valley lines of the scalar field obtained during the last step are extracted as explained previously. Now, the corresponding lines still are not the actual locations of separation and attachment since their accuracy is restricted to the grid resolution (remember that ridge and valley lines always consist of polylines made of grid edges). Nevertheless, previous processing guarantees that actual features' positions lie at most one cell away from the lines found. Therefore, we finally obtain the feature lines by integrating streamlines starting at one end of the ridge and valley lines, and directed toward the converging direction. Integration is interrupted when the streamline reaches the vicinity of the other extremity.

V. Improved Scheme for Stream Surface Computation

Applying the original Hultquist's algorithm,⁷ we face shortcomings with flow of inhomogeneous magnitude and a rather coarse linear interpolation of the surface front, especially for strong surface curvature. These issues gain even more importance as streamsurface are applied e.g. to the study of vortical structures and flow recirculation.

INHOMOGENEITY The fundamental task of stream surface integration is to produce a well conditioned triangulation and good approximation of the real surface. Hultquist achieves this by using a second order ODE solver for streamline integration and selective streamline advancement. This produces triangles whose shape depends on the magnitude of the vector field, even if the resulting surface has no distortions at all to justify this higher resolution. To overcome this limitation, we integrate streamlines over a fixed arc length in combination with a fourth-order Runge Kutta integrator with adaptive step size.¹⁶ We thus operate on precise streamlines and still maintain control over the production of graphical primitives with respect to a user-prescribed threshold.

FRONT CURVATURE The second problem, having the linear approximation of the surface front stand out too plainly, shows up under strong curvature of the surface perpendicular to the flow. By introducing additional criteria for the insertion and deletion of streamlines in the advancing front, we keep control over the spatial resolution based on an angular criterion. More precisely, strong curvature is indicated by high angular deviation of adjacent front segments, as shown in figure 3. The basic rule in Hultquist's front propagation scheme proposed the insertion of new streamlines if the width the quadrilateral at the end of a ribbon between two streamlines surpasses the height by factor 2. We add a new rule to enhance this, based on the angle $\alpha = \angle(q_{i-1}\vec{q}_i, q_i\vec{q}_{i+1})$ for points $q_j, j \in \{1..n\}$ being the projections of the front nodes $p_j, j \in \{1..n\}$ into the plane perpendicular to the flow at p_i :

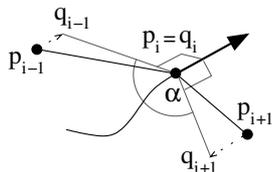


Figure 3. angular criterion

1. Insert a node if the angle exceeds the maximum threshold and the maximum front resolution is not yet reached.
2. Delete a node if the angle falls below the minimum threshold and resolution is still better than the minimum front resolution.

CONTROL PARAMETERS The choice of threshold values is vital in this context, but depends on the data at hand. The parameters influencing the surface quality are the lower and upper bounds for front node distance and intersegment angle. We found useful values with a ratio of 10 to 1 as they effectively prevented oscillation between insertion and deletion of streamlines.

SURFACE PARAMETERIZATION Advancing the front through the vector field automatically produces a parameterization of the surface based on the starting line segment and the integration length in flow direction. The first is a good basis for visualizing flow convergence and divergence as seen in most of the pictures in this paper, whereas the latter runs apart too much over the whole integration in most cases to provide useful results.

VI. Results

A. Wall Shear Stress Analysis

According to what precedes, the analysis of the wall shear stress vector field relies on the extraction and visualization of the separation and attachment lines. The corresponding pictures are shown in Fig. 6.

We start with the point-wise computation of the divergence. We display it using an adaptive color coding that maps values from blue (negative minima) to red (positive maxima). The ridge and valley lines provided by our scheme are shown in the same picture. End loops and short lines are due to the noise inherently increased during derivative computation. This could have been avoided by pre-smoothing of the scalar field, as mentioned earlier. However the picture shows that major features are properly extracted which is essential in the next steps.

The upper right picture illustrates streamline integration toward line features from the separation and attachment lines. It can be observed that this convergence takes place close to the starting locations thus requiring short streamlets to identify separation and attachment lines as asymptotic streamlines. Color coding is red for forward integrated streamlines (separation) and blue for backward integrated streamlines (attachment). Artifacts resulting from the ridge line extraction step are discarded during this flow-based correction step.

The third picture (lower left) shows the corresponding cell-based hit texture that is used to measure spatial convergence of the flow. One can see that the non-zero domains obtained are very thin and constitutes a good approximation of the feature lines, even improved by a (rather trivial) ridge and valley line extraction. The last picture (lower right) finally shows a close-up with the exact paths of separation and attachment lines obtained by streamline integration. Recall start streamlines are started at one end of the ridge and valley lines and integrated toward the converging direction, depending on the sign of the underlying divergence field. The underlying LIC texture permits to underline the accuracy of the results.

An overview of final results is proposed in Fig. 4. Remark that the attachment lines shown in green constitute a non-trivial result since the corresponding converging behavior of the flow only takes place from one side whereas there is no convergence from the inner side. A comparison with Fig. 1 and Fig. 2 shows that Kenwright's technique was unable to find such features.

B. Visualization of Three-Dimensional Structures

To give an overview of the capabilities of the improved streamsurface algorithm, Fig. 5 shows three streamsurfaces close to the delta wing surface. The color map in each case is tied to the parameterization of the starting curve (i.e. the color is transported along individual streamlines in the surface). This permits a recognition of rotation and deformation of the streamsurface.

The streamsurface at the apex was started out of a straight line segment symmetric to the wing. It can be observed how the apex divides the incoming flow in three parts, of which the two fronts that flow above the apex go on to form the characteristic delta wing vortex systems. In Fig. 5, a cutting plane was applied to the streamsurface to reveal the rolling-up of the flow. Fig. 7 (lower left) shows a close up of the vortex system. It can be observed how primary, secondary and tertiary vortices are mutually connected by the surface.

The second set of streamsurfaces in Fig. 5 was computed starting on circular curves around the vortex core lines for the primary vortices on left and right side of the wing. The vortex core lines were extracted using the method of Sujudi and Haines as described in section III and are depicted together with the resulting stream surfaces in the lower right of Fig. 7. Both primary vortices show a breakdown, though asymmetric. Fig. 7 (upper row) shows a closeup of the streamsurface at the breakdown point for the left vortex, where it is possible to observe the blow-up and "folding" of the flow and the recirculation inside the enclosed region (cut open by a cutting plane). The visual artifacts in the upper right of Fig. 7 are related to inaccuracies in the rendering system used to take the images and stem from the fact that the layers of the streamsurface in this region are extremely close.

VII. Conclusion

The objective of the work presented in this paper was to combine established methods with new approaches to provide accurate and efficient visualization results in dealing with the analysis of a delta wing dataset. Our focus was on the extraction and display of structural features like separation, attachment and vortices that are essential for aircraft design. Moreover, following the inspiring work by Dallmann,³ we aimed at proposing depictions based on streamsurfaces to offer intuitive insight into the qualitative behavior of the flow. To this aim, we both applied state-of-the-art visualization techniques and design two new methods to overcome the limitations of existing ones. Our results and observations are summarized next, followed by topics of future work.

EXTRACTION OF LINE-TYPE FEATURES One basic idea behind our visualization approach around the wing body was to start stream surfaces in the vicinity of line-type features that are known to point out “interesting” regions of the flow. The examples considered were separation and attachment lines on one hand, and vortex cores on the other hand. For both types of line features, we tried out standard techniques that compute some kind of linear pattern matching. In our implementation, the parallel operator was used leading to connected features. While the method of Sujudi and Haimes proved satisfying for the extraction of vortex core lines, we obtained rather bad results with Kenwright’s scheme for finding separation and attachment lines. One of the key issues in this case was the sensibility of the method to the noise induced by Jacobian computation over an irregular grid. We saw that pre-smoothing even worsened the problem by shifting and destroying features.

This motivated the design of a new method that takes advantage of the correlation between divergence and separation/attachment lines. Interpreted as a first guess by our method, the ridge and valley lines of the divergence are advected toward the actual features by the flow while streamline convergence is monitored on a cell by cell basis. The regions of strongest convergence are then interpreted as the loci of separation and attachment, according to their empirical characterization. Streamlines are finally properly integrated through these regions to obtain the exact features’ position. Our method is very fast, easy to implement and demonstrated with the delta wing dataset its ability to accurately extract major separation and attachment lines as well as weak features that are non-trivial to identify when looking at a LIC texture of the wall shear stress.

VOLUME VISUALIZATION WITH STREAM SURFACES To permit the visualization and analysis of complex flow behavior, we also improved the basic technique of Hultquist that is unable to properly handle strong curvature and inhomogeneous flow magnitude. Based on our new algorithm, streamsurfaces proved a powerful tool for the structural analysis of a flow field around a delta wing. In the same way that streamlines manage to convey the essential behavior of two-dimensional flows, stream surfaces are able to show the folding and twisting of flow layers. In combination with other techniques like color-coding they serve to enhance line features and dramatically enrich the visualization. Thanks to an accurate and robust implementation we demonstrated the ability of stream surfaces to give deep insight into complex structural configurations like (in our example) vortex breakdown and flow recirculation.

FUTURE WORK As mentioned previously, one of the most promising applications of the visualization techniques presented in this paper is their implication in an automatic, feature-based grid adaption, focusing the computation on interesting regions of the dataset. Future work will include research in this direction. We are also interested in techniques permitting an efficient tracking of the flow structures over time. Finally, we intend to improve our visualization framework to address issues ranging from big airliners’ trailing vortices to vortex genesis at helicopter rotor blades.

Acknowledgments

The authors wish to thank Ronald Peikert from ETH for his precious advices on vortex core extraction. Further we would like to thank the members of the FAnToM project at the University of Kaiserslautern for their implementation effort. This work was partly supported by DFG grant HA 1491/15-4.

References

- ¹Battke, H., Stalling, D., Hege, H.-C., *Fast Line Integral Convolution for Arbitrary Surfaces in 3D*. In Visualization and Mathematics, Hege, H.-C., Polthier, K. (Eds.), Springer, Berlin, 1997, pp.181-195.
- ²Cabral, B., Leedom, L., *Imaging Vector Fields Using Line Integral Convolution*. Computer Graphics (SIGGRAPH '93 Proceedings) 27(4), 1993, pp. 263-272.
- ³Dallmann, U., *Topological Structures of Three-Dimensional Flow Separations*. DFVLR-AVA Bericht Nr. 221-82 A 07, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt e.V., April 1983.
- ⁴Ekatinaras, J. A., Schiff, L. B., *Vortical flows over delta wings and numerical prediction of vortex breakdown*. AIAA paper 90-0102, 1990.
- ⁵Forssell, L. L., *Visualizing Flow Over Curvilinear Grid Surfaces Using Line Integral Convolution*. IEEE Visualization '94 Proceedings, IEEE Computer Society Press, Los Alamitos, 1994, pp. 240-247.
- ⁶Gordnier, R. E., *Computational Study of a Turbulent Delta Wing Flowfield using Two-Equation Turbulence Models*. AIAA 96-2076, 1996.
- ⁷Hultquist, J. P. M., *Constructing Stream Surfaces in Steady 3D Vector Fields*. IEEE Visualization '92 Proceedings, IEEE Computer Society Press, Los Alamitos, 1992, pp. 171-178.
- ⁸Kenwright, D. N., *Automatic Detection of Open and Closed Separation and Attachment Lines*. IEEE Visualization '98 Proceedings, IEEE Computer Society Press, Los Alamitos, 1998, pp. 151-158.
- ⁹Kenwright, D. N., Haines, R., *Automatic Vortex Core Detection*. Computer Graphics and Applications, IEEE, Volume: 18 Issue: 4, Jul/Aug 1998, pp. 70-74.
- ¹⁰Kenwright, D. N., Henze, C., Levit, C., *Features Extraction of Separation and Attachment Lines*. IEEE Transactions on Visualization and Computer Graphics 5(2), IEEE Computer Society, 1999, pp. 135-144.
- ¹¹Langbein, M., Scheuermann, G., Tricoche, X., *An Efficient Point Location Method for Visualization in Large Unstructured Grids*. In Vision, Modeling, and Visualization 2003, T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, E. Steinbach, R. Westermann (Eds.), ISBN 3-89838-048-3, 2003.
- ¹²Modiano, D. L., Murman, E. M., *Adaptive Computations of Flow Around a Delta Wing with Vortex Breakdown*. AIAA paper 93-3400, 1993.
- ¹³Mitchell, A., Morton, S., Forsythe, J., *Analysis of Delta Wing Vortical Substructures using Detached Eddy Simulation*. AIAA 2002-296, 2002.
- ¹⁴Peikert, R., Roth, M., *The "Parallel Vectors" Operator - A Vector Field Visualization Primitive*. IEEE Visualization '00 Proceedings, IEEE Computer Society Press, Los Alamitos, 1999, pp. 263-270.
- ¹⁵Polthier, K., Schmies, M., *Straightest Geodesics on Polyhedral Surfaces*. Mathematical Visualization, Ed: H.C. Hege, K. Polthier Springer Verlag, 1998, pp. 391-408.
- ¹⁶Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., *Numerical Recipes in C*. (2nd ed.) Cambridge, Cambridge University Press, 1992.
- ¹⁷Roth, M., *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD thesis, ETH Zurich, 2000.
- ¹⁸Scheuermann, G., Bobach, T., Hagen, H., Mahrous, K., Hahmann, B., Joy, K. I., Kollmann, W., *A Tetrahedra-Based Stream Surface Algorithm*. IEEE Visualization '01 Proceedings, IEEE Computer Society Press, Los Alamitos, 2001.
- ¹⁹Stalling, D., Hege, H.-C., *Fast and Resolution Independent Line Integral Convolution*. Proceedings of SIGGRAPH '95, Computer Graphics Annual Conference Series, ACM SIGGRAPH, 1995, pp. 249-256.
- ²⁰Sujudi, D., Haines, R., *Identification of Swirling Flow in 3D Vector Fields*. Technical Report, Dept. of Aeronautics and Astronautics, MIT, Cambridge MA, 1995.
- ²¹Taubin, G., *A Signal Processing Approach To Fair Surface Design*. Proceedings Siggraph 95, ACM 1995, New York, pp. 351-358.
- ²²Tobak, M., Peake, D. J., *Topology of Two-Dimensional and Three-Dimensional Separated Flows*. AIAA 79-1480, AIAA 12th Fluid and Plasma Dynamics Conference 1979, AIAA, New York, 1979.
- ²³Tobak, M., Peake, D. J., *Topology of three-dimensional separated flows*. Ann. Rev. Fluid Mechanics, Vol. 14, 1982, pp. 81-85.
- ²⁴van Wijk, J. J., *Implicit Stream Surfaces*. IEEE Visualization '93, IEEE Computer Society Press, Los Alamitos, 1993, pp. 245-252.

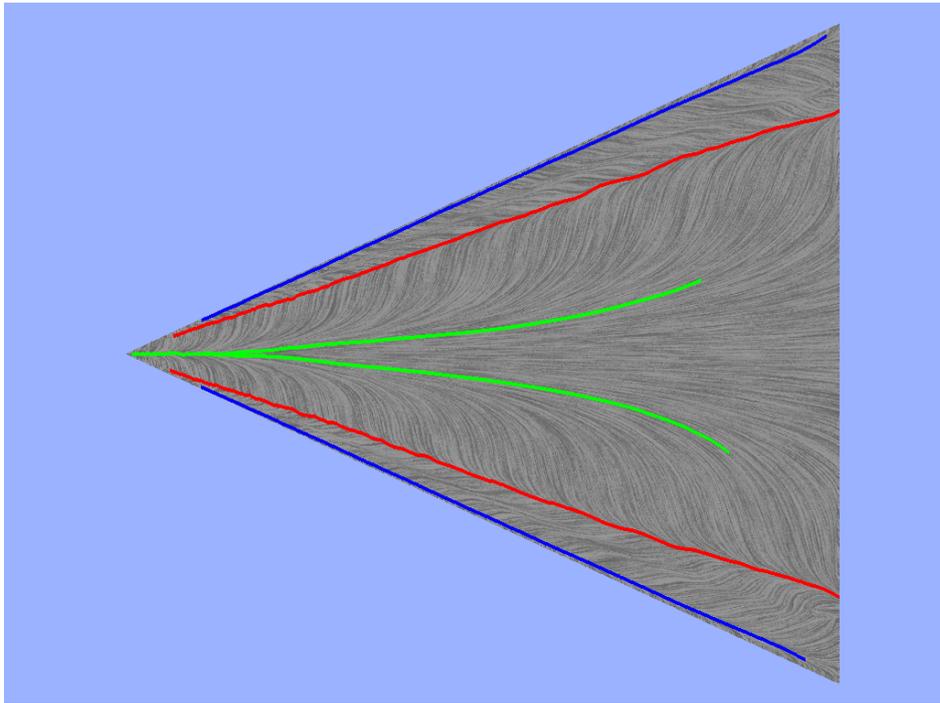


Figure 4. Shear stress analysis: separation and attachment lines over LIC texturing. Color coding: attachment: blue and green, separation: red

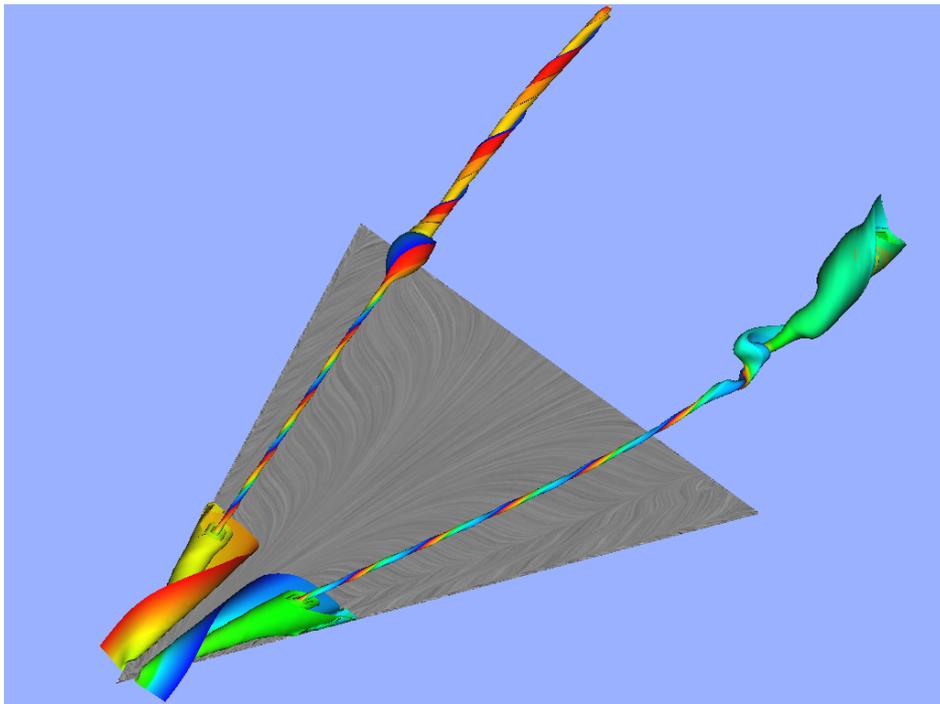


Figure 5. Streamsurfaces near the wing: vortex creation at the wing apex, streamsurfaces around the vortex core lines showing vortex breakdown

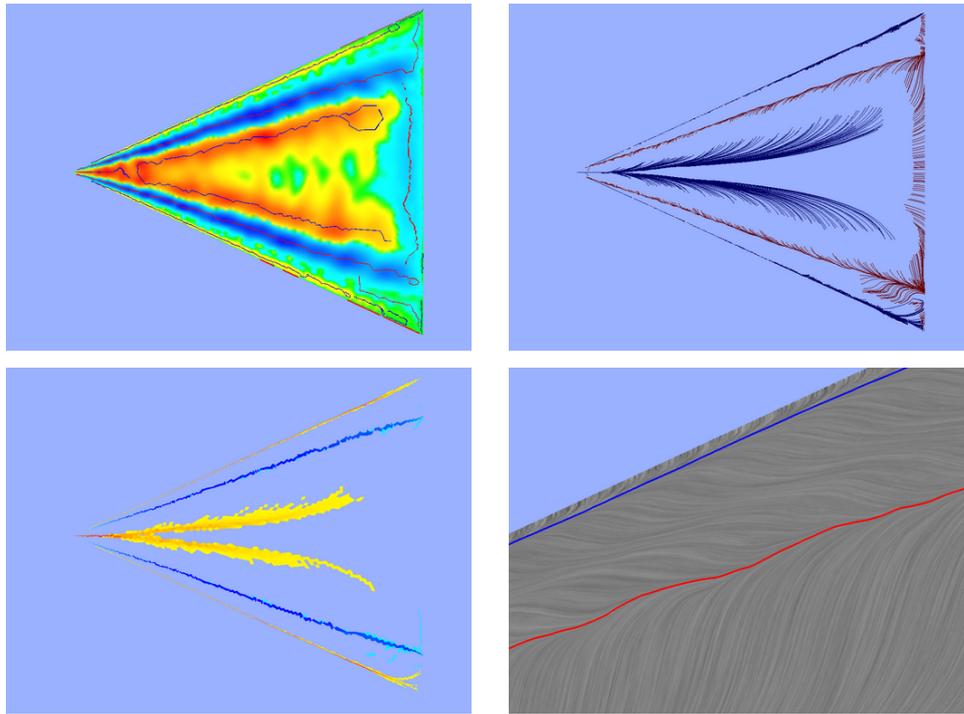


Figure 6. Shear stress analysis: divergence and associated ridge and valley lines (upper left), streamlines integrated toward feature lines (upper right), cell-wise convergence monitoring (lower left), final result and LIC texture - closeup (lower right)

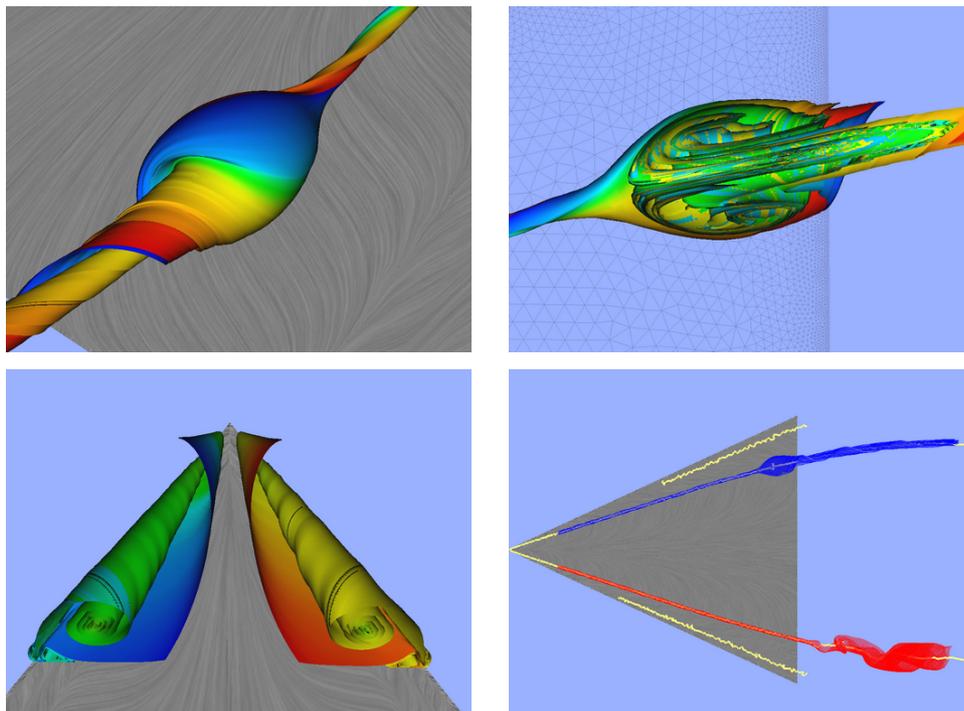


Figure 7. Streamsurfaces around vortex core lines: primary vortex breakdown (upper left), vortex breakdown cut open with a cutting plane (upper right), primary, secondary and tertiary vortex structures (lower left), overview with stream surfaces depicted as wire frame along with primary and secondary vortex core lines extracted by the technique of Sujudi and Haines (lower right)