On the Velocity Gradient Tensor and Fluid Feature Extraction

Robert Haimes*
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
haimes@orville.mit.edu
and
David Kenwright†
MRJ Technology Solutions
NASA Ames Research Center
davidk@nas.nasa.gov

A 3-D Computational Fluid Dynamics flow field may contain many topological features such as vortex cores, separation surfaces, shock surfaces, and recirculation bubbles. We describe several techniques that identify these global features using local analytical tests that can applied independently to any point or cell in a vector field. These techniques draw on concepts from critical point theory and phase plane analysis and utilize the velocity gradient tensor. For this 3x3 tensor an eigen-analysis produces 3 eigenvalues. Mapping these to the complex plane produces the classification signature. Vector field topology can be used as the foundation of automated fluid feature extraction.

Introduction

Fluid flow features such as vortices, separation, and shocks are items of interest that can be found in the results obtained from Computational Fluid Dynamics (CFD) simulations. Most commercial visualization systems provide users with a suite of general-purpose tools (e.g., streamlines, iso-surfaces, and cutting planes) with which to analyze their data sets. In order to find important flow features, users must interactively search their data using one or more of these exploratory tools. Scientists and engineers that use these tools on a regular basis have reported the following drawbacks:

- Exploration Time.
  Interactive exploration of large-scale CFD data sets is laborious and consumes hours or days of the scientists/engineers time.

- Field Coverage.
  Interactive techniques produce visualizations based on a limited number of sample points in the grid or solution fields. Important features may be missed if the user does not exhaustively search the data set.

- Non-specific.
  Interactive techniques usually reveal the flow behavior in the neighborhood of a flow feature rather than displaying the feature itself.

- Visual Clutter.
  After generating only a small number of visualization objects (e.g., streamlines, cutting planes, or iso-surfaces) the display becomes cluttered and makes visual interpretation difficult.

Flow visualization research is now concentrating on feature extraction algorithms that automate the data analysis and extract the salient features with little or no human intervention. One of stumbling blocks has been the lack of precise mathematical definitions for flow structures such as vortex cores, separation lines, recirculation bubbles, and shock surfaces.

Using local representations of the velocity vector and velocity gradient fields, feature extraction techniques have been developed that locate many of the aforementioned features. These techniques, based on concepts from critical point theory, utilize the property that linear vector fields have a finite number of flow topologies. Feature extraction techniques have the following advantages over exploratory visualization tools:

- Fully Automated.
  The analysis can be done off-line in a batch computation.

- Local Analysis.
  The computations for each cell are independent of any other cell and may be performed in parallel.

- Deterministic Algorithms.
  There are no "parameters" that the users can adjust.

*Principal Research Engineer
†Senior Research Scientist

- Data Reduction.
The output geometry is several orders of magnitude smaller than the input data set.

- Quantitative Information.
Precise locations for all the flow features can be extracted.

This paper discusses how the velocity gradient tensor has been used in the visualization of CFD results. In particular, it describes the relationship between the tensor and the topology of the flow field and describes several automated feature extraction techniques based on this relationship.

Calculating the Velocity Gradient

The analyses presented in this paper are three-dimensional. In this context velocity is a vector with 3 components in Cartesian space

$$\vec{u} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Velocity can be obtained for every node in the discretized volume from the results of CFD simulations. This vector field is commonly used in the visualization phase of the analysis to generate streamlines or streaklines in an attempt to assist the investigator in understanding the topology of this vector field. The gradient of the velocity field, $\underline{V}$, is rarely used for visualization purposes, although it is central part of vector field topology.[3]

The velocity gradient contains the information on how the velocity is changing in space. This 3×3 tensor is defined as

$$\underline{V} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} \quad (1)$$

$\underline{V}$ is not usually a quantity that is directly output by a CFD solver but can easily be derived from the vector field and the supporting mesh. For example, a first-order approximation to $\vec{u}$ gives rise to the following linear interpolation function:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} + \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2)$$

A tetrahedron has exactly the correct amount of information to compute $\underline{V}$ and the coefficient vector $(c_1, c_2, c_3)^T$. This means that a unique (and constant over the entire element) $\underline{V}$ can be constructed. Elements with more nodes (e.g. pyramids, prisms, hexahedra, and etc.) have a changing $\underline{V}$ based on the cell's interpolant. For all grid types, elements can be broken up into tetrahedra, $\underline{V}$ calculated and then distributed back to the nodes in a finite element manner.

It should be noted that $\underline{V}$ is not dimensionless. It has the units of $time^{-1}$. Therefore most all data derived directly from the velocity gradient tensor has the form of a rate.

Velocity Gradient Decomposition

A common technique used to better understand the local flow is to decompose $\underline{V}$ to its symmetric and anti-symmetric parts:

$$\underline{S} = \frac{1}{2}\left(\underline{V} + \underline{V}^T\right) \quad (3)$$

$$\underline{\Omega} = \frac{1}{2}\left(\underline{V} - \underline{V}^T\right) = \frac{1}{2}\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (4)$$

Where $\underline{S}$ is a measure of the strain which contains both bulk and shear components. $\underline{\Omega}$ contains the rotational part of the flow. As can be seen in Equation (4) there are only 3 components of this tensor. Vorticity is more commonly viewed as a vector:

$$\vec{\Omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \end{bmatrix} \quad (5)$$

The trace of $\underline{S}$, which equals the trace of $\underline{V}$, reflects the divergence of the field:

$$\nabla \cdot \vec{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \quad (6)$$

This decomposition has been used by a number CFD visualization researchers, notably:

- Darmofal and Haimes[4]. During the integration of streamlines both $\nabla \cdot \vec{u}$ and $\vec{\Omega}$ are tracked. The streamlines optionally are plotted as ribbons, where one edge is the actual streamline and the other rotates to reflect the curl. The streamline could be plotted as a tube centered on the streamline where the thickness of the tube relates to the local divergence. This is similar to the *Stream Polygon*.[16] Also, an option would draw a spiral pattern on the tube to display the direction (the tube itself), the divergence (the tube cross-section size) and the vorticity (the spiral pattern).

- deLeeuw and vanWijk[5]. Here an interactive probe is constructed that points in the local direction of the flow and has the ability to concurrently display curvature, $\nabla \cdot \vec{u}$, $\vec{\Omega}$, acceleration and shear in a physically meaningful manner.

These techniques are interactive and attempt to locally display components of $\underline{V}$.

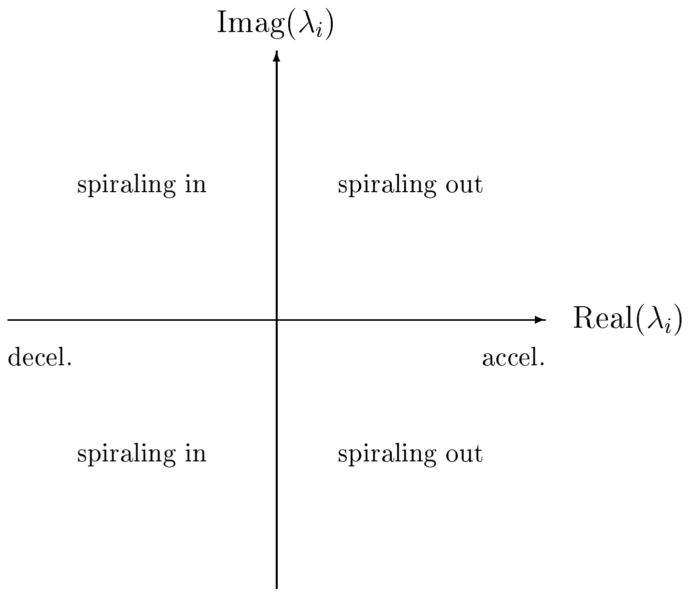

Fig. 1 Flow classifications in eigenvalue plane, $\lambda_{1,2,3} = \mathbf{eig}\,(\underline{V})$.

Critical Point Theory

Critical points of a vector field are those points where the magnitude of the velocity vanishes. Stated in another way, these points are defined where the streamline slope is indeterminate and the velocity is zero relative to an observer.[3] From this definition it is clear that the process of locating critical points is not Galilean invariant.

Topologically, critical points in the field mark changes. An eigen-analysis is used on the velocity gradient tensor at critical points to classify the local topology. For $\underline{V}$, the eigenvalues ($\lambda_{1,2,3}$), are the fundamental quantities which determine the qualitative features of the flow pattern. In general, various possible flow patterns and the corresponding eigenvalue-based flow classifications are summarized in Figure 1. It should be noted that when 3 non-zero eigenvalues exist, there are two general types: 3 real eigenvalues or 1 real (λ_r) and a complex conjugate pair (λ_c).

Any real eigenvalue sits on the real axis of Figure 1. The sign of the eigenvalue indicates whether the flow is accelerating (positive) or decelerating (negative). The magnitude of the value reflects the strength. The complete suite of critical point classification for 3 real eigenvalues from $eig(\underline{V})$ can be seen in Figure 2. In this Figure, each type is displayed by a *cartoon* of the flow pattern approaching the critical point and a plot of the λs in the complex eigenvalue plane.

In Figure 2 the portrait is displayed as flow on a 2D plane and then arrows indicating a flow direction orthogonal to that surface. The choice is somewhat arbitrary. The plane is defined by the 2 of eigenvectors (the ones plotted on the plane).

When there is a complex conjugate pair of eigenvalues, λ_r acts as described above. The magnitude of the imaginary part of λ_c indicates the strength of the spiraling flow. If the value is small (near the real axis) the flow (in the plane) is hardly swirling. If the magnitude is great then the flow is rotating rapidly about the point. The sign of the real part of λ_c indicates whether the flow is converging (negative) or diverging (positive) from the point, with the magnitude of the value again reflecting the strength of the attraction or repulsion. The special case where the real part of λ_c is zero just produces concentric periodic paths.

The complete suite of critical point classification for λ_r and λ_c can be seen in Figure 3. In this Figure, each type is again displayed by a *cartoon* of the flow pattern approaching the critical point on the complex eigenvalue plane. The portrait is displayed as flow on a 2D plane and then arrows indicating a flow direction orthogonal to that plane (this direction is defined by the eigenvector associated with λ_r). In these cases, the plane always depicts the spiraling flow. The plane is defined by the 2 eigenvectors associated with λ_c.

Critical Points and general eigen-analysis of $\underline{V}$ has been used by a number CFD visualization researchers, these include:

- Globus, Levit and Lasinski[7]. This paper introduced the vector field topology module to FAST.[1] In this work, critical points were located and marked with *glyphs*. The shape of these icons reflected the critical point classification. One could use this module to interactively mark vortex cores. This was drawn by selecting *glyphs* that indicate spiraling flow and integrating a streamline in the direction of the eigenvector associated with λ_r. This was an important step in the right direction for feature extraction. Unfortunately, this module of FAST was not frequently used because:
  - A knowledge of Critical Point Theory. Without the knowledge and understanding of the concepts, looking at these icons told the investigator little.

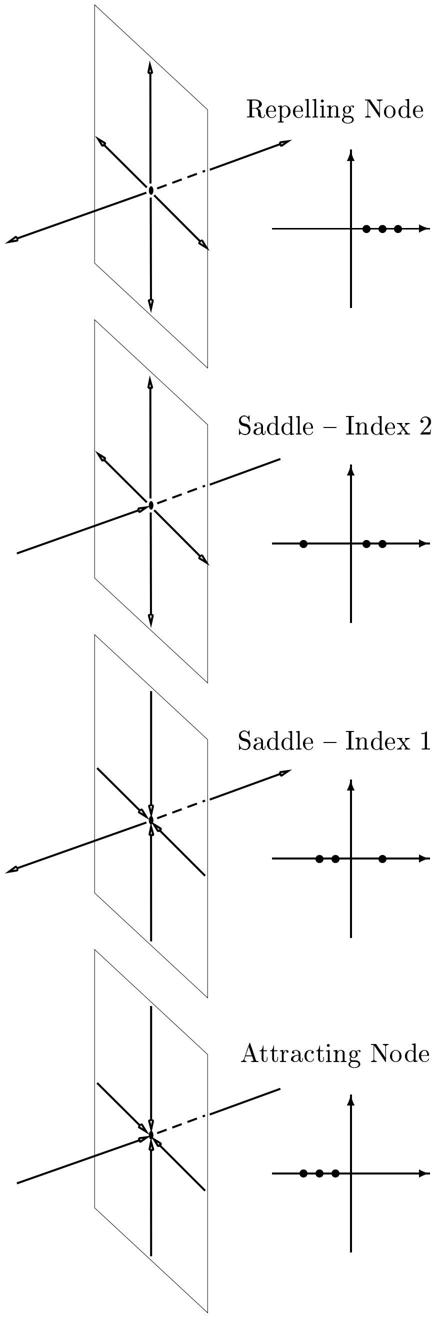

Fig. 2 Critical point portrait and eigenvalue plane for non-spiral flow

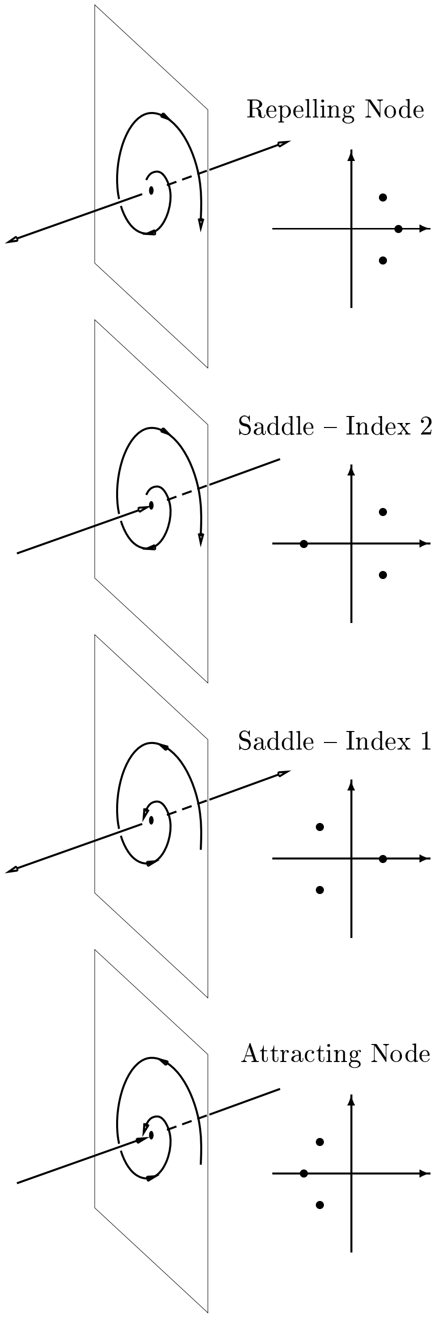

Fig. 3 Critical point portrait and eigenvalue plane for spiral flow

  - Results were difficult to interpret. Even with the knowledge, one needs a good spatial imagination to *connect the dots*.
  - Complex Flows suffered from clutter. With an interesting (and therefore complex) flow topology the number of Critical Points may approach 100. It is not clear what one can do with this many (sometimes overlapping) *glyphs*.

- Delmarcelle and Hesselink[6]. The concept of a *Hyperstreamline* is introduced. This concept uses tensor field lines as the direction of the *Hyperstreamline* and the cross-section is defined from the eigen-analysis. This technique also suffers from difficult to interpret graphics.

Fluid Feature Extraction

Visualization is the final phase of the suite for traditional CFD analysis. In this step, the investigator attempts to understand the results by visually probing the data. These tools (cuts, iso-surfaces, streamlines and others) only hint at the real answers. Usually, much interaction is required to get a full understanding of the flow field, in particular where the flow topology is complex.

Feature extraction is the next important step in visualization. The results can be used interactively to point directly to areas of interest. As analysis suites become more integrated, feature extraction can be used directly to enhance the fidelity of the solution by grid adaptation in the appropriate regions. During parametric studies, the classification of features found in the simulation can be quickly scanned to find transitions in topology.

The discussion below will review the important fluid flow features and how the velocity gradient tensor either contains the ability to extract the feature or is otherwise affected.

Shocks

Normal shocks express themselves as abrupt changes in the magnitude of the velocity field. This can be seen in the eigenvalues of $\underline{V}$ as at least one strong negative real λ. This is not sufficient for constructing a shock finder because one can not easily differentiate a strong compression wave from a shock. See Lovely and Haimes[13] for a complete description on shock extraction.

Vortex Cores – Eigen-Analysis

Vortices can be automatically detected by using $\underline{V}$ throughout the mesh looking for situations of swirling flow. All mesh elements are broken into tetrahedra (if not already this type of element). The unique $\underline{V}$ is constructed and then classified. If swirling, the direction orthogonal to the spiral plane (the eigenvector associated with λ_r) is used as the axis of swirl. This direction is subtracted from the nodal velocities. These *reduced velocities* are used to see if any faces display a zero. If so, that location on the face is marked. With two (or more) marked faces on the tetrahedron, it is determined that the core center-line has pierced the cell. These lines are collected and drawn to display the core segments.

This particular algorithm is fully described in Sujudi and Haimes[18] and contains no constants and requires no user intervention. The core finder has been applied to a fairly complete suite of applications, see Kenwright and Haimes.[10] For an example see Figure 4.

The strength of the vortex is best described by $|\vec{\Omega}|$.

This technique, although satisfying, does not work under all circumstances. The problems are:

- Not producing contiguous lines.
  The method, by its nature, does not produce a contiguous line for the vortex core. This is due to two reasons; (1) for element types that are not tetrahedra the interpolant that describes point location within the cell is not linear. This means that if the core passes through these elements the line can display curvature. By subdividing pyramids, prisms, hexahedra and higher-order elements into tetrahedra for this operation produces a piecewise linear approximation of that curve. And (2) there is no guarantee that the line segments will meet up at shared faces between tetrahedra. This is because the eigenvector associated with the real eigenvalue will not be exactly the same in both neighbors, so when this vector is subtracted from the vector values at the shared nodes each tetrahedra sees a differing *reduced velocity* field for the face.

- Locating flow features that are not vortices.
  This method finds patterns of swirling flow (of which a vortex core is the prime example). There are other situations where swirling flow is detected, specifically in the formation of boundary layers. Most implementations of this technique do no process cells that touch solid boundaries to avoid producing line segments in these regions. But this does not always solve the problem. In some cases (where the boundary layer is large in comparison to the mesh spacing) this boundary layer generation is still found.

- Sensitive to other non-local vector features.
  Critical point theory gives one classification for the flow based on the local flow quantities. 3D points can display a limited number of flow topologies including swirling flow, expansion and compression (with either acceleration or deceleration). The flow outside this local view may be more complex and have aspects of all of these components. The local classification will depend on the strongest type. Also if there are two (strong) axes of swirl, the scheme will indicate a rotation that is a combination of these rotation vectors based on the relative strength of each. This has been reported by Roth and Peikert[14] where the overall vortex core strength was not much greater that the global curvature of the flow. The result was that the reported core location was displaced from the actual vortex.

The first point can be addressed by re-casting the algorithm to be face-based instead of cell-based. Enforcing the cell piercing to match at cell faces insures that the line segments generated will produce a contiguous core. This can be done via the following modification to the algorithm:

1. Compute the $\underline{V}$ at each node.
   This requires much more storage – 9 words are needed for each node in the flow field. This has the advantage that the stencil used for the operation is larger than the cell and therefore the result will be generally smoother.

2. Average the node tensors (on the face) to produce a face-based $\underline{V}$.
   This insures that the same tensor is produced for the two cells touching the face.

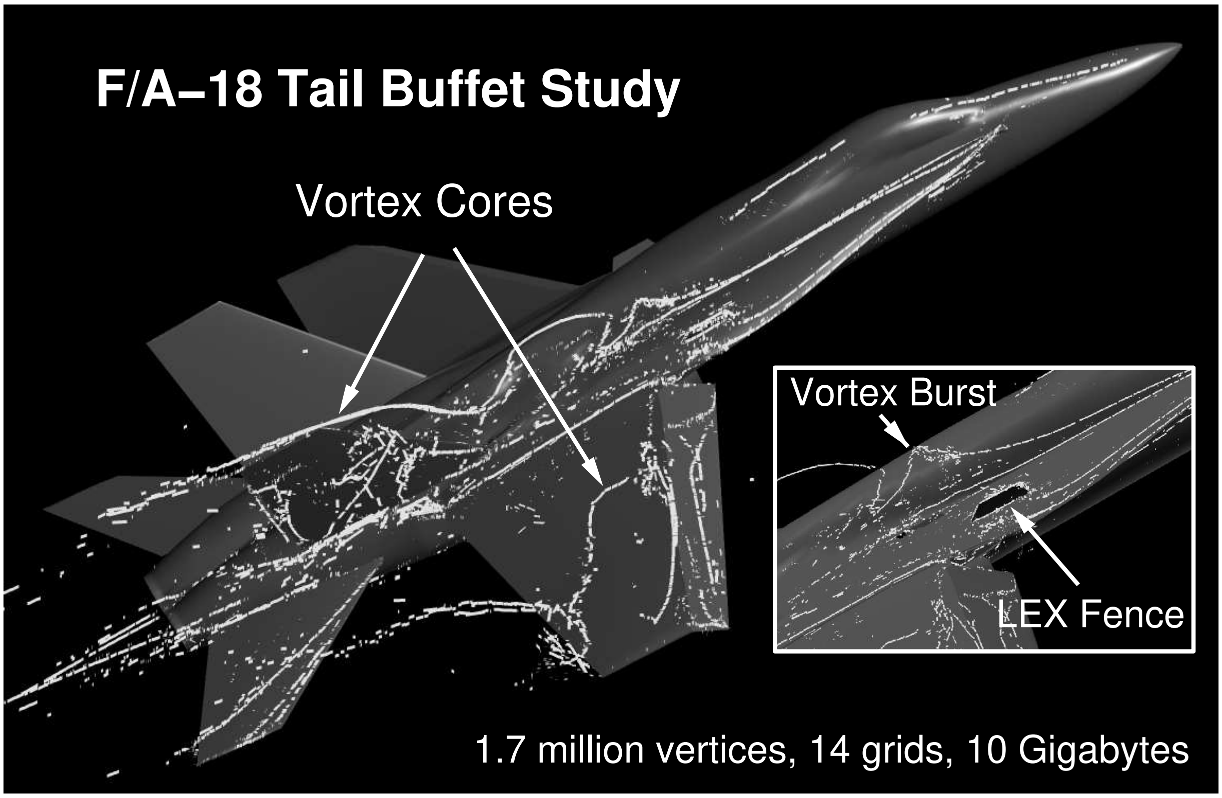

Fig. 4 Vortex cores that were extracted using eigenvector analysis from one of the 304 vector fields in a transient, high alpha, F/A-18 simulation. The vortex cores provided insight into the vortex breakdown; a phenomenon that was promoted by the addition of a trapezoidal fence on the wing's leading edge extension (LEX). The vortex burst occurs where the vortex core starts to spiral, just above and behind the fence.

3. Perform the eigen-analysis on the face tensor.
   If the system signifies swirling flow, determine if the swirling axis cuts through the face by looking at the *reduced velocity*. If, so mark the location on the face.

This is not a good result in terms of CPU cycles, in particular for structured blocks where each individual hexahedron is broken up into 6 tetrahedra (5, the minimum does not promote face matching). This means that for each element in the mesh a minimum of 12 eigen-analyses are required.

This performance problem suggest another, related, technique:

1. Compute the velocity gradient tensor at each node.

2. Perform the eigen-analysis on the node tensor.
   The tensor can be overwritten with the critical point classification and the swirl axis vector for rotating flow.

3. Average the swirl axis vectors for the nodes that support the tetrahedral face.
   This should only be done if all nodes on the face indicate swirling flow. Some care needs to be taken to insure that the sense of the vectors are the same. Determine if the swirling axis cuts through the face, and if so, mark the location on the face.

For tetrahedral meshes, the reduction of compute load is by a factor of 5 to 6 over the original method (there are roughly 5.5 tetrahedra per node in 'good' unstructured grids). For structured blocks, where the number of nodes is about equal to the number of hexahedra, the eigen-analyses count is on the order of one per cell.

Vortex Cores – Alignment

The updated eigen-analysis algorithm is still linear and will produce incorrect results when the flow is under 2 or more (relatively equal) rotating influences (the third problem listed above). Roth and Peikert[15] presented a paper that, on first viewing, appears to solve this problem. The authors suggest that by looking at higher-order derivatives of the velocity field one can capture curvature. They first recast the technique described above then apply the higher-order correction:

- Parallel alignment
  An intersection point on a face is where the *reduced velocity* is zero. Therefore the velocity vector is parallel to the real eigenvector of $\underline{V}$. $\vec{u}$ is an eigenvector of $\underline{V}$ and therefore a solution of

$$\underline{V}\vec{u} = \lambda\vec{u}. \tag{7}$$

This suggests that looking for parallel alignment of $\underline{V}\vec{u}$ with $\vec{u}$ is the same as the vortex algorithm described above.

- Second Order
Computing the velocity second derivatives ($\mathbf{T}$) generates a $3x3x3$ tensor. Adjusting Equation (7) for the second derivative produces:

$$\underline{VV}\vec{u} + \mathbf{T}\vec{u}\vec{u} = \kappa\vec{u}. \tag{8}$$

- Alignment with Sencond Order Correction
Equation (8) is used to find where in the flow field $\underline{VV}\vec{u} + \mathbf{T}\vec{u}\vec{u}$ is parallel to $\vec{u}$. These locations are marked and collected like the eigen-analysis technique.

In practice, this does not seem to work well. The velocity field generated by most CFD solvers is not smooth. As a result this technique produces both many false positives and also misses intersections due to the local saw-tooth variations. The other problem is the storage requirements. For large data sets, requiring 27 words/node of the rank 3 tensor and 9 words/node for the rank 2 tensor becomes prohibitive.

The issue of alignment coupled with the linear limitations of the classification and results of the eigen-analysis hint at an alternative technique. This approach uses $eig(\underline{\Omega})$ instead of $eig(\underline{V})$. The result is invariably one zero and a complex conjugate pair (λ_c) of eigenvalues – always the indication of swirling flow. The eigenvector associated with the zero eigenvalue can be used with the face-piercing algorithm to produce core segments. This technique has the effect of removing the bulk and shear components of stress from the analysis.

The result of this algorithm is, at times, longer and more contiguous cores. For example, tracking a horse-shoe vortex in a turbine cascade, the core line would disappear as the flow accelerated around the crown of the suction surface. This was due to a change in classification from swirl to non-swirling flow. With the irrotational components removed the core can continue to be extracted. The influence of the irrotational terms also explains why the core segments would break-up when the core strength would diminish. Under these conditions the stress components end up corrupting the direction of the eigenvector used for the swirl axis.

It should be noted that the eigenvector associated with the zero eigenvalue of $eig(\underline{\Omega})$ is just $\vec{\Omega}$. This means that the storage requirements are no longer 9 words per node but 3 and no eigen-analyses are required. The result is that vortex cores can simply be found where $\vec{\Omega}$ aligns with $\vec{u}$.

Vortex Cores – Maximum Vorticity

The techniques discussed thus far identify the centers of rotation in velocity vector fields. However, the assumption that a vortex core lies at the center of a rotating flow can be misleading in flows where there are multiple vortices in close proximity. If vortices rotate in the same direction and their rotational cores interact, then the resulting velocity field will exhibit only one rotational center. The basic problem is that the combination of rotational vortex velocities does not preserve the center of rotation for each individual vortex. Strawn et al.[17] demonstrated this phenomenon using both analytical and CFD flow fields for rotorcraft problems. To identify the individual vortex centers, Strawn et al. defined the vortex center as a local maximum of vorticity in a plane that is normal to the vorticity vector. This technique identifies free-shear vortex centers, but correctly rules out shear layers, which have high vorticity values, but no local vorticity maxima. The differences between the vorticity maxima and the swirling flow techniques are apparent in Figure 5.

Separation and Attachment

Helman and Hesselink[8,9] developed a visualization scheme for generating separation surfaces based on vector field topology. The topology of a vector field consists of critical points and the tangent curves that connect these points. Because the velocity at a critical point is zero, the velocity field in the neighborhood of the critical point is determined from the velocity gradient tensor. This technique starts by finding the critical points on the surface of the object. Streamlines are integrated along the principal directions of the saddle critical points and then linked to these critical points to produce a 2-D skeleton of the flow topology near the object, as shown in Figure 6. Streamlines are then integrated out to the external flow starting from points along certain curves in the skeleton. These streamlines are tessellated to generate the separation surfaces.

Kenwright[11] developed a local technique for extracting separation and attachment lines based eigen-analysis of the velocity gradient tensor. This local scheme detects two types of separation; open and closed. Closed separation lines originate and terminate at critical points, i.e., the type found by Helman and Hesselink's algorithm. Open separation lines do not need to start or end at critical points and cannot be detected by vector field topology methods.[2] Kenwright's technique applies phase plane theory to the surface flow on triangular elements near a body. The three velocity vectors at the vertices of each triangle are used to construct a two dimensional linear vector field with a constant velocity gradient tensor. Tensors with two real eigenvalues produce flow patterns that contain separation and attachment lines. The trajectories for these curves are found by solving a non-homogeneous second-order ordinary differential equation. The separation and attachment lines are special lines along which the stream function is zero or singular. If one of

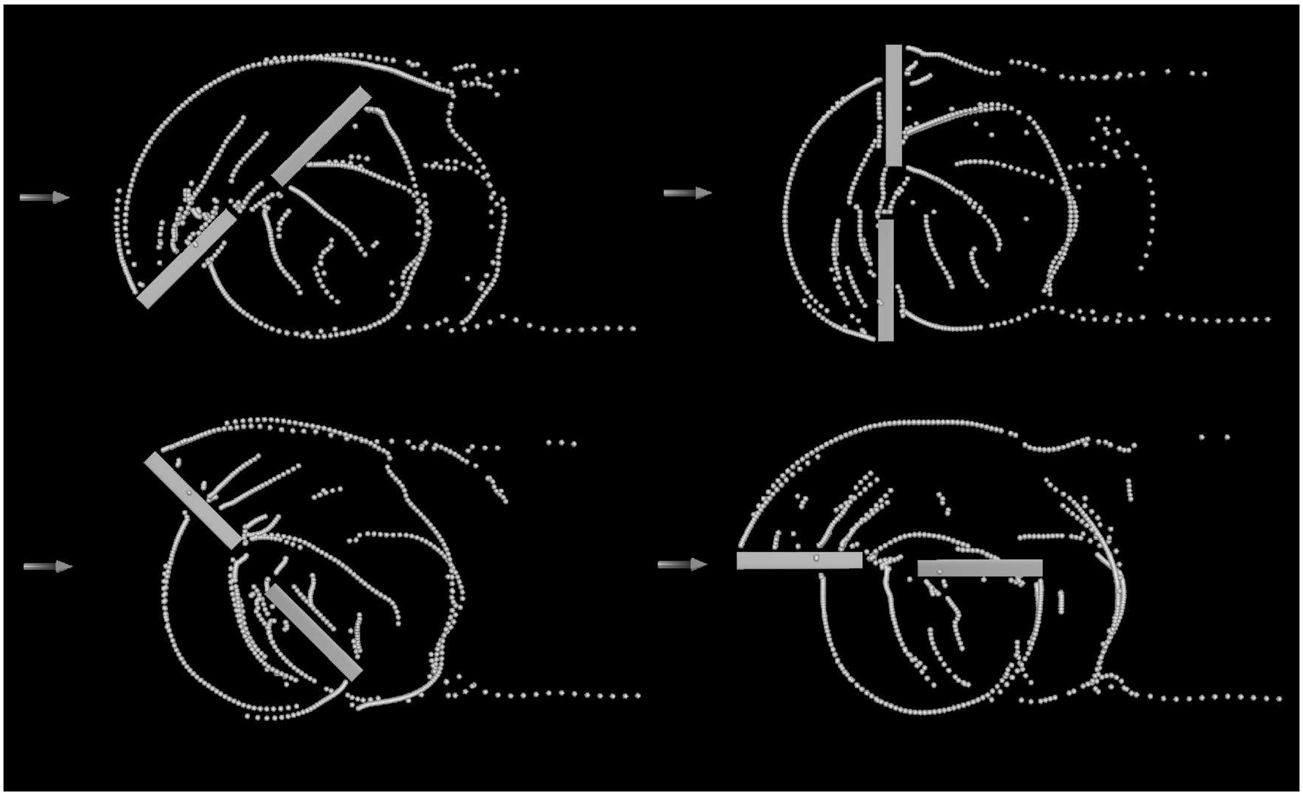

Fig. 5 Rotorcraft simulations produce complex flow fields that contain multiple vortex cores in close proximity. The vortex cores shown here were extracted using a technique that identifies local maxima of vorticity. Techniques that identify centers of swirl using eigenvector analysis can fail to distinguish individual vortices if the rotational vortex cores overlap.

these lines happens to cross its associated triangle, the line segment bounded by the triangle is collected for rendering. By repeating this process in every triangle, the independent line segments combine to form global separation or attachment lines. Kenwright[11] showed that this technique identified all the separation and attachment lines on the surface of a 65 degree delta wing in a simulated high-alpha flow. The majority of the separation and attachment lines in that flow did not start or end at critical points, i.e., they were open separation and attachment lines.

The phase plane technique[11] has one drawback. The linear interpolant does not guarantee that line segments in neighboring cells will meet at cell boundaries. This is because the location and direction of the separation lines is dictated by the gradients of the velocity interpolant, which, in the phase plane algorithm, are constant over each triangle but discontinuous between triangles. This problem was overcome by using a point-based scheme developed by Kenwright, Henze, and Levit.[12] Separation and attachment lines are said to occur at points where the velocity vectors are parallel to the eigenvectors of the velocity gradient tensor, i.e., points where the streamline curvature is zero. This condition also identifies inflection lines, although they can be removed using a directional derivative test. As with the phase-plane technique, the parallel vector technique is based on linear approximations. It produces slightly better visualizations than the phase plane algorithm because it ensures piecewise continuity of the velocity and eigenvector fields. An example of this technique is shown in Figure 7.

Boundary Layers & Wakes

There has been little success in the ability to automatically locate boundary layers from the output of CFD solvers. In fact, many algebraic and one equation turbulence models need to know the edge of the boundary layer. In most all cases vorticity is used as the marker. This has been determined to be inadequate.

Boundary layers (and wakes) display two features; (1) the generation of vorticity and (2) the fluid is under shear stress. This suggests a marker that is a function of both $\vec{\Omega}$ and shear.

$\underline{S}$ from Equation (3) can be used to compute the rate of shear stress. This stress tensor contains both the bulk and shear stresses and is dependent on the coordinate system. To extract a single scalar that is coordinate system invariant and has the bulk terms removed it is necessary to diagonalize this tensor (again another eigen-analysis). The result is always 3 real eigenvalues ($\underline{S}$ is symmetric positive definite). These eigenvalues (λ_{s1}, λ_{s2}, λ_{s3}) produce a vector which signifies the 'principle axis of deformation'. By employing techniques from Solid Mechanics, the norm of the sec-

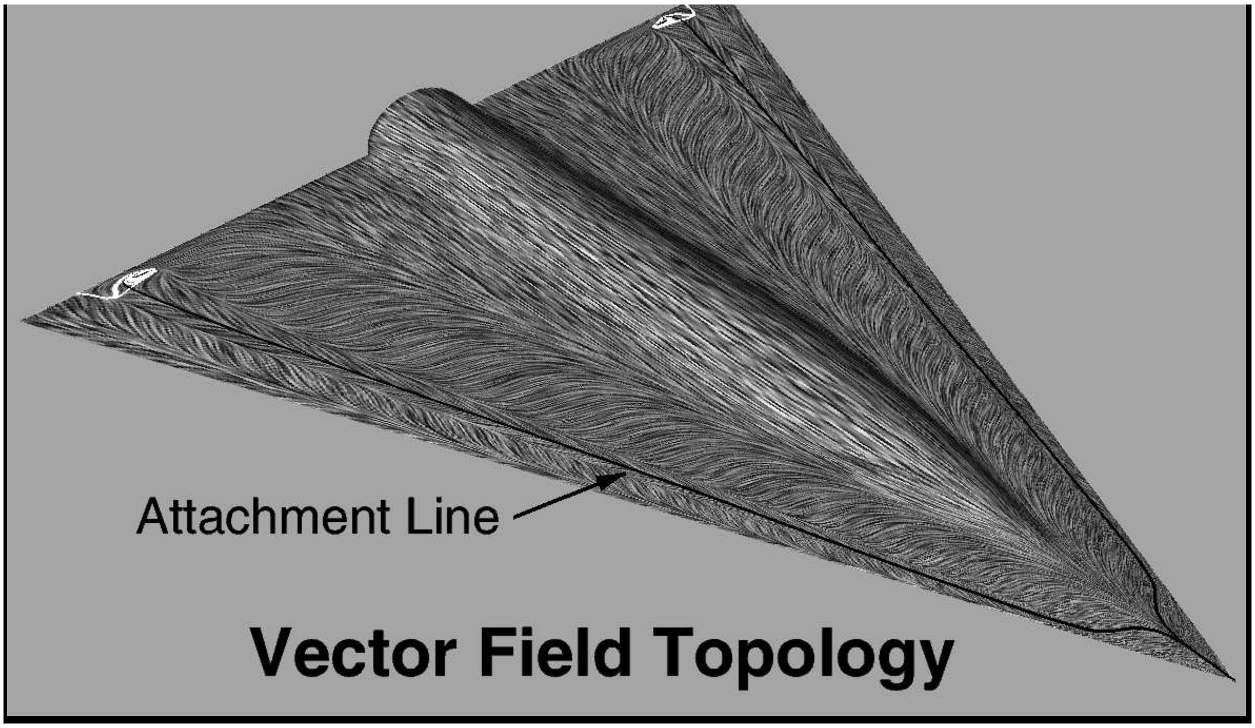

Fig. 6 The separation and attachment lines, colored white and black, respectively, identify the places where the airflow detaches and returns to the surface of this delta wing. These separation and attachment lines were computed using vector field topology.

ond principal invariant of the 'stress deviator' can be used as a measure of the shear. This is

$$\sqrt{\frac{(\lambda_{s1}-\lambda_{s2})^2+(\lambda_{s1}-\lambda_{s3})^2+(\lambda_{s2}-\lambda_{s3})^2}{6}}. \quad (9)$$

The boundary layer equations are essentially 2D. Using only the 2 strongest eigenvalues empirically gives better results. Therefore Equation (9) reduces to

$$\lambda_{s1}-\lambda_{s2}. \quad (10)$$

A scalar field can be constructed from a function of $|\vec{\Omega}|$ and the value from Equation (10). This node-based field can be passed through an iso-surface algorithm. This marks those areas that have a certain amount of both rotation and shear stress. This empirical approach has shown good results in detecting boundary layers and wakes but has the following drawbacks:

- A function of shear and rotation is *ad hoc.*
- The value is not non-dimensional, but has units of inverse time.
  This means that the iso-surface value used to define the edge of the layer changes from case to case. This scalar needs to be multiplied by some characteristic time associated with the problem.
- The value used for the iso-surface is not specified via theory.

Conclusions

The ability to automatically detect flow features from the results of CFD codes is closely tied to the velocity gradient tensor $\underline{V}$. Decomposition into the rotational and irrotational parts allows examination of some aspects of the local flow. Critical Point Theory can also be used to map the eigenvalues of $\underline{V}$ onto the complex plane thus providing a local classification of the flow. These aspects can be used to attempt to automatically detect fluid flow features by following the local topology of the vector field to gain a global view of the features.

Acknowledgments

This work was partially sponsored by NAS at NASA Ames Research Center with David Kao as the Technical Monitor and by the Army Research Office with Stephen Davis as the Technical Monitor. Additional support has come from the IBM SUR Project and the IBM UUP Project.

References

[1] G. Bancroft, F. Merritt, T. Plessel, P. Kelaita, R. McCabe, A. Globus. FAST: A Multi-Processing Environment for Visualization of CFD. Proceedings of IEEE Visualization '90, 1990.

[2] G.T. Chapman. Topological Classification of Flow Separation on Three Dimensional Bodies. AIAA Paper 86-0485, 1986.

[3] M. Chong, A. Perry and B. Cantwell. A General Classification of Three-Dimensional Flow Fields. Phys. Fluids A, vol. 2, pp. 765-777, 1990.

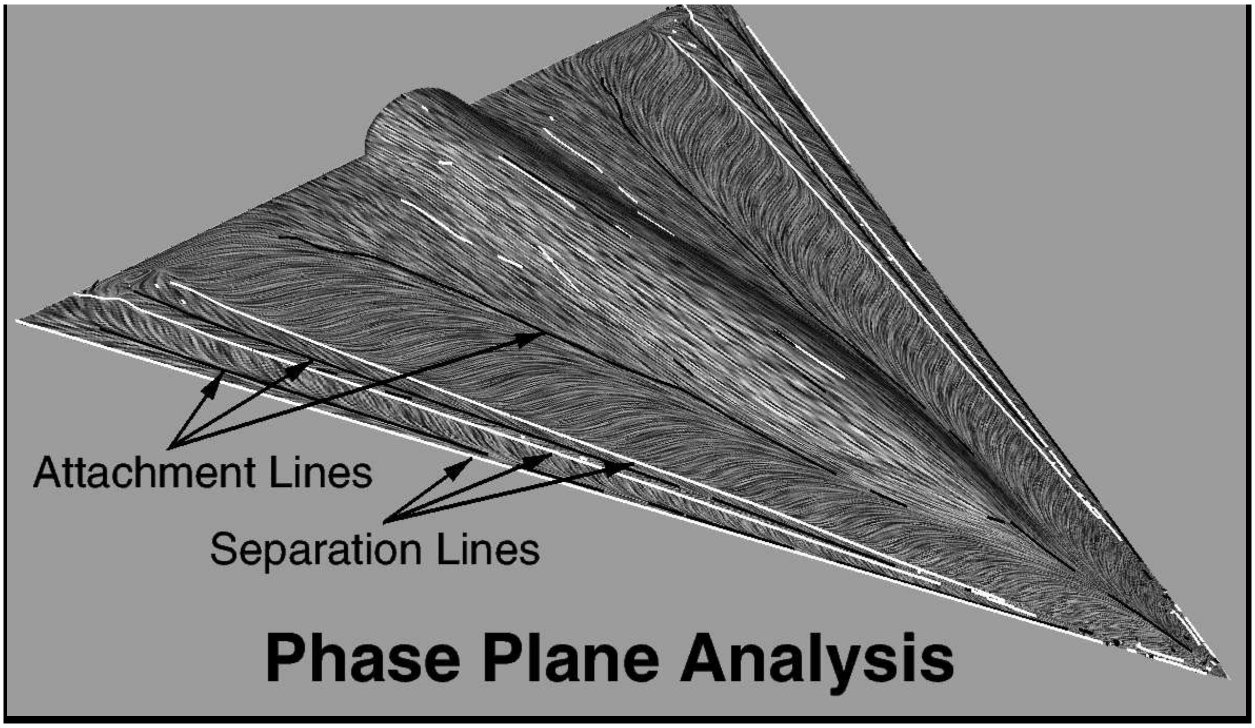

Fig. 7 The separation and attachment lines, colored white and black. These lines were computed using the parallel vector technique. This method identifies more lines because it detects both open and closed separation lines.

[4]D. Darmofal and R. Haimes. Visualization of 3-D Vector Fields: Variations on a Stream. AIAA Paper 92-0074, 1992.

[5]W. deLeeuw and J. vanWijk. A Probe for Local Flow Field Visualization. Proceedings of IEEE Visualization '93, 1993.

[6]T. Delmarcelle and L. Hesselink. Visualization of Second Order Tensor Fields and Matrix Data. Proceedings of IEEE Visualization '92, 1992.

[7]A. Globus, C. Levit and T. Lasinski. A Tool for Visualizing the Topology of Three-Dimensional Vector Fields. Proceedings of IEEE Visualization '91, 1991.

[8]J. Helman and L. Hesselink. Analysis and Representation of Complex Structures in Separated Flows. SPIE Proceedings, Vol 1459, 1991.

[9]J. Helman and L. Hesselink. Visualizing Vector Field Topology in Fluid Flows. IEEE Computer Graphics and Applications, May 1991.

[10]D. Kenwright and R. Haimes. Vortex Indentification – Applications in Aerodynamics. Proceedings of IEEE Visualization '97, 1997.

[11]D. Kenwright. Automatic Detection of Open and Closed Seperation and Attachment Lines. Proceedings of IEEE Visualization '98, 1998.

[12]D. Kenwright, C. Henze, and C. Levit. Feature Extraction of Separation and Attachment Lines. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society Press, Vol. 5, No. 2, June 1999.

[13]D. Lovely and R. Haimes. Shock Detection from Computational Fluid Dynamics Results. AIAA Paper 99-3285, 1999.

[14]M. Roth and R. Peikert. Flow Visualization for Turbomachinery Design. Proceedings of IEEE Visualization '96, 1996.

[15]M. Roth and R. Peikert. A Higher-Order Method For Finding Vortex Core Lines. Proceedings of IEEE Visualization '98, 1998.

[16]W. Schroeder, C. Volpe and W. Lorensen. The Stream Polygon: a Technique for 3-D Vector Field Visualization. Proceedings of IEEE Visualization '91, 1991.

[17]R. Strawn, D. Kenwright, and J. Ahmad. Computer Visualization of Vortex Wake Systems. AIAA Journal, Vol. 37, No. 4, April 1999.

[18]D. Sujudi and R. Haimes. Identification of Swirling Flow in 3-D Vector Fields. AIAA Paper 95-1715, 1995.