

LLNL-CONF-400950



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Vortex Characterization for Engineering Applications

M. Jankun-Kelly, D. S. Thompson, M. Jiang, B.  
Shannahan, R. Machiraju

February 1, 2008

46th AIAA Aerospace Sciences Meeting and Exhibit  
Reno, NV, United States  
January 7, 2008 through January 10, 2008

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

# Vortex Characterization for Engineering Applications

M. Jankun-Kelly\* and D. Thompson†

*Mississippi State University, Mississippi State, MS, 39762, U.S.A.*

M. Jiang‡

*Lawrence Livermore National Laboratory, Livermore, CA, 94551, U.S.A.*

B. Shannahan§ and R. Machiraju¶

*The Ohio State University, Columbus, OH, 43210, U.S.A.*

**Realistic engineering simulation data often have features that are not optimally resolved due to practical limitations on mesh resolution. To be useful to application engineers, vortex characterization techniques must be sufficiently robust to handle realistic data with complex vortex topologies. In this paper, we present enhancements to the vortex topology identification component of an existing vortex characterization algorithm. The modified techniques are demonstrated by application to three realistic data sets that illustrate the strengths and weaknesses of our approach.**

## I. Introduction

As computing power continues to increase, large-scale computational fluid dynamics (CFD) simulations of complex vortical flows about realistic configurations are becoming routine. Visualization of the simulation results is often used to gain insights into the underlying physical phenomena. Feature-based techniques have been demonstrated to provide viable methods for visualizing large-scale simulations of this type.<sup>1-5</sup> These methods extract features (in this case vortices) and define them in terms of high-level, feature-based descriptors. This process is sometimes termed feature (vortex) characterization. These algorithms essentially act as data dimensionality reduction techniques that reduce the dimension of the simulation data from several million entities (number of nodes times number of flow variables) to a concise, more tractable description of the flow phenomena of interest.

The vortical flows associated with realistic configurations may be very complex. Furthermore, due to practical limitations related to memory availability and computational cost, the flow may not be optimally resolved, either spatially or temporally. Many existing vortex detection techniques are ill-suited to deal with data that does not have well-resolved features. For instance, noise in the data may contaminate gradient computations that are essential components of many vortex detection and characterization algorithms. Visualization researchers employ the term *uncertainty* to describe the various forms of errors that can occur in data of this type. However, since uncertainty is employed differently in the CFD literature,<sup>6</sup> we will defer to this nomenclature and use the term *error* to describe the phenomena that arise due to the discrete approximation. To be useful to application engineers, vortex characterization techniques must be sufficiently robust to handle existing simulation data with such shortcomings. Therefore, it is imperative to develop feature detection and characterization techniques that are robust even when presented with complex, coarsely-resolved data.

In this paper, we describe enhancements to our vortex characterization algorithm<sup>5</sup> that improve results for realistic engineering simulation data. We focus on enhancements to a  $k$ -means clustering<sup>7,8</sup> based topology identification algorithm. We first describe our characterization algorithm and note some of its shortcomings.

---

\*Graduate Research Assistant, Computational Simulation and Design Center, Box 9627, AIAA Student Member.

†Associate Professor, Department of Aerospace Engineering, Box A, AIAA Associate Fellow.

‡Postdoctoral Researcher, Center for Applied Scientific Computing, Box 808, L-560.

§Graduate Research Assistant, Department of Computer Science and Engineering, 395 Dreese Labs, 2015 Neil Avenue.

¶Associate Professor, Department of Computer Science and Engineering, 395 Dreese Labs, 2015 Neil Avenue.

We then discuss the enhancements to the topology identification algorithm. Results for three data sets<sup>9–11</sup> are included to highlight the strengths and weaknesses of our approach.

## II. Related Work

Comprehensive overviews of existing vortex detection algorithms are given by Jiang et al.<sup>12</sup> and Roth<sup>13</sup> including useful taxonomies to help guide the selection of algorithms appropriate for various situations. One categorization is based upon the representation of the vortex. A line-type algorithm extracts a vortex core line while a region-type algorithm extracts a region in which a vortex is located. Unfortunately, many existing techniques produce unsatisfactory results when applied to large-scale data sets with complex physics.

Banks and Singer<sup>14</sup> developed a line-type algorithm that employs a predictor/corrector sequence. Their approach is based on two assumptions: (1) a vortex core is a vorticity line and (2) the pressure in the core is a local minimum. Sujudi and Haimes<sup>15</sup> described a line-type method that extracts the vortex core by locating points that satisfy two conditions: (1) the velocity gradient tensor has a complex eigenvalue pair and (2) the reduced velocity, e.g., the velocity in the plane perpendicular to the real eigenvector, is zero. Roth and Peikert<sup>16,17</sup> proposed a different approach for detecting core lines that detects parallel alignment between the velocity vector and the jerk vector, which is the vector time derivative of the acceleration. Other topology-based techniques for visualizing vortices in large-scale simulations have been reported in the literature.<sup>1–3</sup> These extract vortex skeletons and define them in terms of feature-based descriptors.

Levy et al.<sup>18</sup> developed a region-type method that assumes that a vortex core is located in a region where the normalized helicity approaches  $\pm 1$ . Berdahl and Thompson<sup>19</sup> use the fact that two of the eigenvalues of the velocity gradient tensor are a complex conjugate pair in regions of swirling flow to define a scalar swirl field using the magnitude of the imaginary part of the conjugate pair and the reduced velocity. Jeong and Hussain<sup>20</sup> define a vortex using the symmetric deformation tensor  $S$  and the antisymmetric spin tensor  $\Omega$ . Their criterion is that a point is located in a vortex if  $\lambda_2$ , the second largest eigenvalue of  $S^2 + \Omega^2$ , is negative at that point. Jiang et al.<sup>21</sup> employed a combinatorial labeling scheme based on Sperner's Lemma applied to the velocity vector field. An objective definition of a vortex has been proposed by Haller<sup>22</sup> for unsteady, three-dimensional incompressible flows. Haller defines a vortex as a set of fluid trajectories along which the the Cotter-Rivlin derivative of the rate of strain is indefinite over directions of zero strain, which Haller terms the  $\mathbf{M}_Z$  criterion. This approach has the advantage that it is Galilean invariant.

Both types of methods have shortcomings, however. Bauer and Peikert<sup>23</sup> proposed a preprocessing step for the parallel vectors operator<sup>13</sup> to reduce artifacts occurring due to numerical approximation of higher order derivatives. Jiang et al.<sup>24</sup> proposed a postprocessing step for automatically verifying the results from the detection algorithm. The work presented by Stegmaier et al.<sup>25</sup> attempted to address robustness issues associated with schemes of this type by combining the method of Jeong and Hussain<sup>20</sup> with that of Banks and Singer.<sup>14</sup> Rütten<sup>26</sup> addresses these issues by using a combination of the various methods to narrow the selection of candidate cells. Then the parallel vectors operator is employed to determine a seed point and streamline integration is used to represent the vortex axis.

Several approaches for computing the physical extent of a vortex, which is an ambiguous quantity, have been proposed in the literature. Banks and Singer<sup>14</sup> presented an approach that samples pressure and vorticity along radial lines emanating from the detected core line, until user defined thresholds are exceeded. Roth<sup>13</sup> proposed a similar approach using a fraction of the value at the core position for each cross section as the threshold. Recently, Garth et al.<sup>27</sup> proposed an improvement to the above approach by approximating the physical extent that eliminates user-defined thresholds by using the local maximum as the termination criterion.

## III. Characterization of Vortical Flows

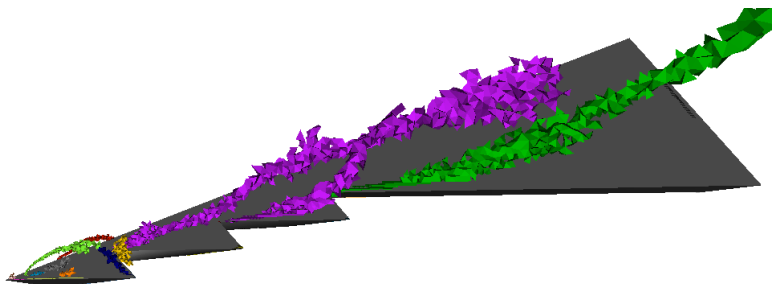
In this section, we provide a brief description of our vortex characterization algorithm. For a detailed explanation, we refer the reader to our previous work.<sup>4,5</sup> We employ the serrated wing data set<sup>10</sup> to illustrate the steps in the process.

## A. Core Line Extraction

The primary component of the characterization algorithm is extraction of the vortex core line. Our technique employs a predictor-corrector algorithm to extract the location of a vortex core line from a scalar field in which a line-type extrema coincides with the vortex core line. Here, we employ the swirl parameter of Berdahl and Thompson.<sup>19</sup> Candidate cells are selected based on whether or not the eigenvalues of the velocity gradient tensor contain a complex conjugate pair,<sup>28</sup> and then whether or not the cell lies on a local line-type maximum of the swirl parameter field. Contiguous collections of the candidate cells are then aggregated to form regions that may contain a vortex core. Next, a  $k$ -means<sup>7,8</sup>-based algorithm is employed to identify the aggregate topologies and segment them appropriately. Subcell resolution of the vortex core line position is achieved by assuming a conical variation of the scalar field in the region near the core line.

### 1. Identify Candidate Cells and Aggregate

The local extrema method (LEM)<sup>4,5</sup> exploits the fact that, for certain scalar fields, a vortex core line coincides with a line-type extremum in the field variable. Therefore, a vortex core line can be extracted from an appropriate scalar field  $f$  by locating local extrema in a series of properly oriented planes and connecting the location of the extreme values with line segments to form a curve. A five-point stencil is employed to determine if the node under consideration is a locally extreme value. The central point of the stencil lies in the cell currently under consideration, while the other four points lie outside the cell along perpendicular rays in a planar local neighborhood. In this case, we employed the eigenvector corresponding to the real eigenvector to define the normal to the plane. When the value of  $f$  is largest at the central point, the current cell lies on a local maximum curve. The cells containing local extrema are marked by the LEM as candidate cells. Since we want to work at the feature level rather than the cell level, we group contiguous candidate cells into aggregates. Small aggregates can be eliminated by unmarking aggregates that contain less than a certain number of cells. The result for this step is shown in Figure 1.



**Figure 1.** Contiguous candidate cells are aggregated. Aggregates with less than a specified minimum number of cells are removed.

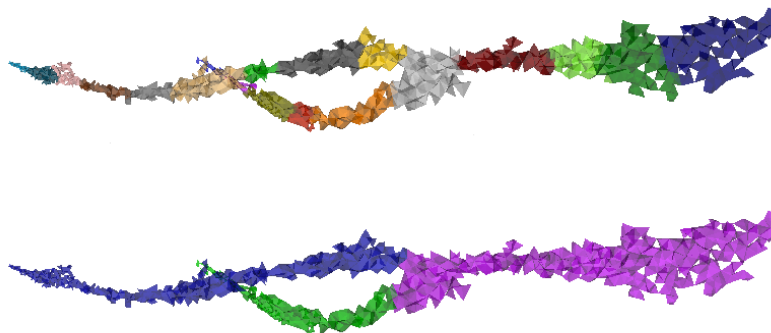
Occasionally, an aggregate produced may contain holes in its interior. One approach is to employ hole filling. Hole filling marks cells as candidate cells if they are surrounded by a specified number of neighboring cells that are candidates. A second option is directional growth, which grows candidate cell regions along the core line direction, as approximated by the swirl vector.

### 2. Identify Aggregate Topology and Segment

One of the problems with existing vortex detection algorithms is that they have difficulty distinguishing vortices that are merging. The LEM may also mark cells in a manner that causes distinct vortices in close proximity to one another to appear to be merging. We would like to be able to distinguish the individual vortices that are merging from the resulting merged vortex. The serrated wing data set exhibits this phenomenon, which is shown in Figure 1. There are two distinct vortices created by the first and second serrations, as numbered from the left. They begin to merge slightly to the right of the third serration, forming a Y-shaped aggregate, which is colored purple. This topology presents challenges for the core line extraction algorithm because only one vortex core line is found per aggregate. The branching aggregates need to be subdivided into component non-branching aggregates before core line extraction can be applied.

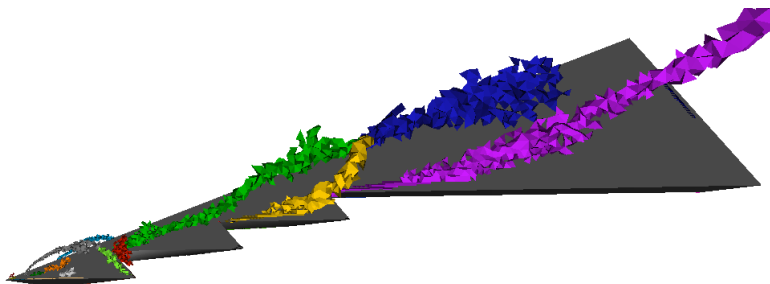
In order to correctly identify the vortex topology, we developed a clustering algorithm to identify if and where two vortices merge – in other words, where the branching of the aggregate occurs. This clustering algorithm is a variation of the  $k$ -means clustering algorithm,<sup>7,8</sup> designed specifically for unstructured meshes. The idea is to decompose the entire aggregate into small segments (clusters) and determine if branching occurs. To create the initial set of clusters, we apply the standard region-growing algorithm for unstructured meshes, but we only grow each cluster to a fixed neighborhood size. Each neighborhood iteration consists of adding the connected neighbors that are within the aggregate to the cluster. For the serrated wing data set, we used a neighborhood size of four.

Once the entire aggregate is decomposed, we winnow out the tiny clusters by redistributing the cells within the tiny clusters to the closest connected cluster. For the serrated wing data set, we used a minimum cluster size of 16. Next, we compute the centroids of all the remaining clusters. At this point, we apply the standard  $k$ -means clustering algorithm,<sup>7,8</sup> with the only difference being that we do not use any prescribed value for  $k$ . The algorithm terminates once the same sets of cells have been identified to be the centroids. Figure 2 shows the individual clusters resulting from our algorithm for the merging vortices.



**Figure 2.** Zoomed in view of the merging vortices. The top image shows the clusters from the modified  $k$ -means algorithm. The bottom image shows the merged clusters.

After the clusters have been computed, we proceed to identify branching clusters. In this case, a *simple* cluster should have at most two neighboring clusters and only one if it is at a terminus. A *non-simple*, or branching, cluster will have more than two neighboring clusters. We identify branching clusters and merge the clusters in the branching regions separately. Figure 3 shows the result from the clustering and merging step of our algorithm. Notice that we have correctly distinguished the merging vortices from the merged vortex.



**Figure 3.** Aggregates after clustering. Notice that the single purple aggregate shown in Figure 1 has been divided into three constituent aggregates.

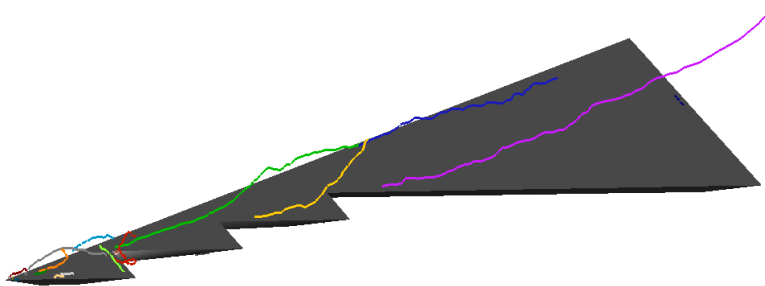
### 3. Extract Vortex Core Line

We start by defining a seed point for the core line extraction in each aggregate. We locate the point on the aggregate's bisecting plane with the largest swirl value and use this as the seed point. We then correct the position of this point using function fitting applied to the scalar field in the swirl plane. Our objective is to find the location of the extreme value in the local swirl plane. It is highly unlikely that the extreme value falls exactly on a cell center. Thus, this extreme value cannot be found with linear interpolation of the values at the cell centers. If we use a higher-order polynomial interpolation, unrealistic oscillations may occur. What we do know about the field is that it attains a local maximum (or minimum) at the vortex core line in the swirl plane perpendicular to the vortex core line. Therefore, in the local neighborhood of the vortex core line, we can search for the location of the local maximum (or minimum) by fitting a conical function to the data in the swirl plane. We chose a conical function because, like the field in the local neighborhood of the core line, it has a single extreme value.

Function fitting consists of several steps. First, we select the fitting data. The data points are the center of the cell in which the core line currently resides, as well as the centers of the neighboring cells (face, edge and node neighbors). Next, we take the cell centers and project their locations onto the swirl plane. We can justify using the values at the cell centers as follows. In the local neighborhood near a vortex core line, the isosurfaces of the scalar field appear as tubes around the vortex core. Thus, the original cell center and its projection onto the swirl plane lie on the same "isotube" and have the same value of the scalar field.

Now that we have two-dimensional data in the swirl plane, we can fit a conical function to it. Fitting involves moving the center of the cone, the location of the function's extreme value, to different positions in the local neighborhood, computing the standard deviation of the fitting error, and selecting the location with the smallest standard deviation of the fitting error. Our local neighborhood is the size of a two-dimensional bounding box that contains all the projected data points in the swirl plane.

Given a current point on the core line, we wish to find the next point along the core line. We predict where the next point will lie using the swirl vector, which is an approximate tangent to the vortex core line. A ray is shot from the current point in the direction of the swirl vector. The intersection of the ray with a cell face is found and is taken to be a predicted point on the curve. The correction algorithm is applied and the process is repeated. The core line extraction algorithm terminates when it hits an aggregate other than the one in which it started, when it hits a domain boundary, or when it has traversed more than a certain number of contiguous unmarked cells. Core line ends that lie in unmarked cells are removed. The result of this step is shown in Figure 4.



**Figure 4.** Vortex core lines are extracted, one line per aggregate. Core lines have been colored by the aggregate ID of the aggregate they were extracted from (Figure 3)

### B. Extent Surface Computation

Once the core line is extracted, other vortex characteristics can be computed. One characteristic of interest is the extent of the vortex. Unfortunately, the extent of a vortex is an ambiguous concept. We employ the procedure outlined by Garth et al.<sup>27</sup> who compute an extent curve in the plane containing the swirling motion by determining a set of radial positions at which the tangential velocity is a maximum. A surface is developed by lofting between each of the two-dimensional extent curves as shown in Figure 5.

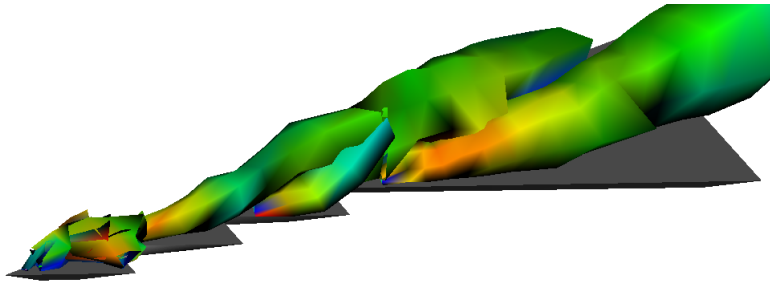


Figure 5. Vortex characteristic visualization: Vortex extent shaded by maximum tangential velocity.

### C. Shortcomings of Existing Method

We now briefly describe some of the shortcomings of the existing method.

- Missed vortices: In some cases, the topology identification and segmentation algorithm fails. Generally this occurs because the topology of the aggregate has been incorrectly identified.
- Discontinuous core lines: In some cases, the core lines that are extracted are not  $C^0$  continuous. A core line extracted in its entirety but in several pieces has breaks. A core line with segments missing has gaps. Gaps occur when core line extraction terminates prematurely and indicate a need for greater robustness of the core line extraction algorithm.
- Nonsmooth core lines: In many instances, the core lines that are extracted, while  $C^0$  continuous, are not smooth, i.e., not  $C^1$  continuous. This may be due to a number of factors. Among them are the discrete nature of the data and the increased error due to computation of derived fields necessary for core line extraction.
- Nonsmooth extent surfaces: The extent surface extraction technique is based on the premise that there is a well-defined maximum tangential velocity along each radial line emanating from the vortex core in the plane containing the swirling motion. The locus of these points forms a closed curve that describes the extent of the vortex. Essentially, this is a template that is employed to define the extent of the vortex. In some cases, such as two vortices in close proximity to one another or when a vortex is interacting with other flow features, this template is not appropriate.

## IV. Enhanced Topology-Based Identification and Segmentation

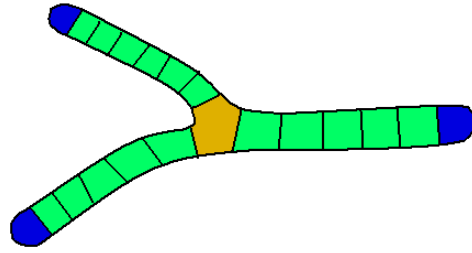
In this section, we present enhancements to the aggregate topology identification and segmentation algorithm. To illustrate some of the issues associated with complex vortex topologies, we again consider the flow over the serrated wing<sup>10</sup> shown in Figure 1. This flow contains multiple merging vortices. While the individual vortices in these flows may be readily apparent to the human observer, they are not to a machine. Thus, there is a need for an algorithm that identifies individual vortices in complex flows.

### A. Aggregate Topology Identification and Segmentation

We have briefly presented a method for identifying individual vortices in complex flows based on vortex topology<sup>5</sup> in Section III. We now focus on the topology identification and segmentation component of this algorithm.

In the aggregate topology identification step, illustrated in Figure 6, we look for branch points where two aggregate branches merge. At a branch point, we expect to see a cluster with three neighbors. At the ends of the branches, we expect to see clusters with one neighbor. Between the terminal points and branch points, we expect to see clusters with two neighbors. The baseline method could identify individual, line shaped aggregates and aggregates with two merging branches, but could not handle aggregates with diverging branches nor multiple merging branches.





**Figure 6.** Aggregate of candidate cells colored by cluster neighbor count: blue – one neighbor (branch termination), green – two neighbors (part of an individual branch), and orange – three neighbors (branch point/point where two branches merge)

In order to extract individual vortices from complex vortical flows, we first need to classify the aggregates according to their topology. Our topology classification system is shown in Figure 7. The simplest case is the isolated non-branching linear aggregate (see Figure 7(a)). Loops are another example of a non-branching but topologically distinct aggregate (see Figure 7(b)). The other cases contain multiple branches and / or loops. The simplest of the multi-branch cases are shown in Figures 7(c) and 7(e). Two branches may originate near each other, then diverge (see Figure 7(c)). Two branches may merge (see Figure 7(e)). Initially diverging branches may re-merge (Figure 7(d)). Multiple branches can merge (see Figures 7(f) and 7(g)). While all aggregates may not fit neatly into any of these cases (see Figure 7(h)), they may be broken up into pieces that do.

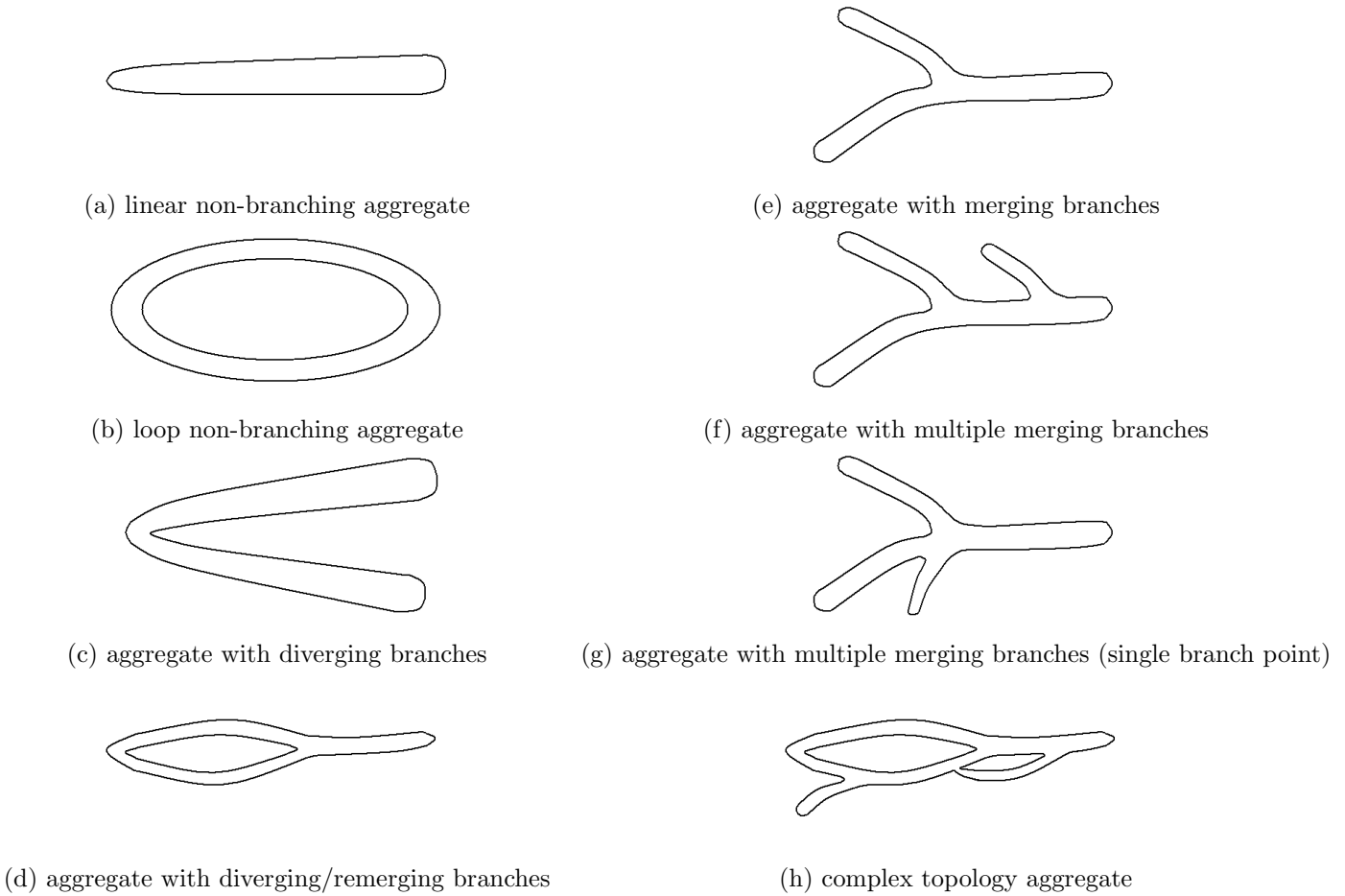
## B. Implementation Challenges

Our vortex topology identification method presents us with two main implementation challenges. The first is creation of candidate cells clusters with the following properties:

- Cluster boundaries are perpendicular to the vortex core line
- Clusters at vortex origin and ending points have one neighbor
- Clusters at branch points have three or more neighbors
- Clusters between origin, ending, and branch points have two neighbors

These requirements must be met in order to apply our vortex topology algorithm which is based on cluster neighbor counts. We have found that the standard  $k$ -means clustering algorithm does not always produce candidate cell clusters that meet these requirements. We have made three improvements to the clustering algorithm.

- The baseline  $k$ -means algorithm is based on distance between cells and thus produces clusters whose boundaries may not be perpendicular to the vortex core line. For this reason, we use the distance between cells projected onto the vortex core line rather than regular Cartesian distance. This results in cylindrical shaped clusters whose cluster-cluster boundaries are perpendicular to the core line. Since the vortex core line has not yet been extracted when this is being done, we approximate it by using the swirl vector at the current cluster center.
- In our previous work, the  $k$ -means clustering algorithm used a fixed minimum cluster size. This size was chosen to work well with the data set we were currently working on. However, in other data sets, it produces too many small clusters. When clusters are too many and too small, their neighbor count goes up and their boundaries may not be perpendicular to the core line. We now use a minimum cluster size related to aggregate size (the number of cells in the aggregate divided by 20).
- Our previous clustering algorithm would consider clusters to be neighbors if they had even one pair of neighboring cells. We now use a more stringent neighbor definition: neighbors must have a *sufficient number* of neighboring cells.



**Figure 7. Aggregate topology classification system**

- The aforementioned hole filling directional growth enhancements to the cell marking and aggregation stage improved the quality of the input to the clustering algorithm. The clustering algorithm has difficulty when many holes are present in the marked cell region. Since hole filling and directional growth greatly reduce the number of holes, the clustering algorithm’s output quality improves.

The second implementation challenge is correct subdivision of all the aggregate topology cases shown in Figure 7, such as diverging branches aggregate and multiple merging branches aggregate. The goal of topology based subdivision is to end up with non-branching subaggregates (straight or loop). In order to do this, we propose the following improvements to our method:

- Handle multiple merging branches
  - Handle multiple branch points rather than stopping after the first
  - Handle branch points joining three or more branches rather than just three
- Handle the two diverging branches case (Figure 7(c))
- Handle loops
- Differentiate between loops and diverging / remerging

The multiple branching cases require two changes. First, the aggregate is subdivided into all branch points, not just the first one found. Secondly, the definition of branch point is expanded from a three neighbor cluster to three *or more* neighbor cluster, allowing for more than two merging branches. This process is illustrated in Figure 8. First, branch points are identified. They aggregate is then split at the branch points by allocating their cells to neighboring clusters and adjusting the cluster neighbor count. The result should be non-branching subaggregates composed of two terminal points at either end of a chain of two neighbor segments. Neighboring clusters are then reaggregated into non-branching aggregates from which core lines can be extracted.

### C. Future Modifications

Loop aggregates can be identified by their lack of branch points and terminal points. They are composed of segments with two neighbors each. This is a special case where there is no branch point and no terminal points.

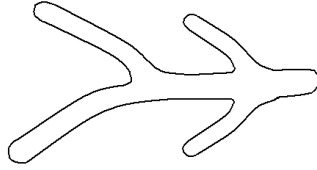
Diverging branches likely indicate two counter-rotating vortices that originate near each other. We cannot use the neighbor count definition of a branch point in this case. Thus, we must expand or change the definition of branch point. Counter-rotating vortices have opposite signs of normalized helicity at their cores. Thus, we can consider the normalized helicity of the clusters and pick the one where helicity changes sign as the branch point, as shown in Figure 9. We choose not to use the geometric position of the clusters relative to each to select the branch point because a single hairpin vortex could give rise to an aggregate with a “diverging branch” shape, but it should not be subdivided. A hairpin vortex would have the same sign of normalized helicity in all the clusters of its aggregate. This definition of branch point could also be used in the diverging / remerging case.

## V. Results

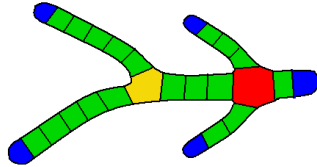
In this section, we will demonstrate the effectiveness of our algorithm by applying it to three different simulation data sets. We first discuss the inherent difficulties of quantifying improvement on the characterizations and present our strategy and the rationale for using it.

### A. Quantification of Improvements

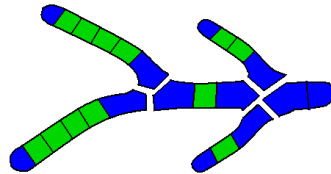
Quantifying improvement in vortex characterizations is a difficult proposition. It is difficult to develop an objective measure of quality for these algorithms – what exactly constitutes “better” is sometimes subjective. The characteristics of vortices in the simulation data are not usually known *a priori* except in the case of synthetic data. This makes it difficult to assess the error in the characterization. Additionally, due to error



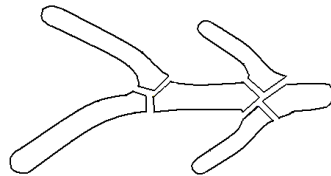
(a) A multi branch aggregate requires subdivision into non-branching aggregates before core lines can be extracted.



(b) The aggregate is split into clusters with one (blue), two (green), three (orange) or more (red) neighbors. Branch points are identified as clusters with three or more neighbors.

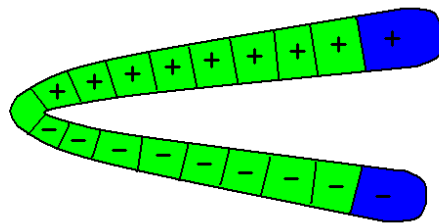


(c) Branch point clusters are split among their neighbors and neighbor count is adjusted accordingly.



(d) One and two neighbor clusters are left. They are merged into non-branching aggregates. Vortex core line extraction can now proceed.

**Figure 8.** A multi-branch aggregate is split at multiple branch points. The neighbor count is adjusted. Only non-branching subaggregates remain.



**Figure 9.** Diverging branch aggregates do not have a branch point with three or more neighbors, but may have a branch point where normalized helicity changes sign.

from multiple sources in the simulation itself, comparison of derived vortex characteristics with experimental data may also be problematic.

In some cases, visual assessment is a possible strategy. However, this technique is usually only effective in situations in which the vortex characterization algorithm fails to properly define an entity as a vortex. In our algorithm, this happens most frequently when the topology identification component fails to properly segment candidate cell aggregates and no core line is extracted from a portion of an aggregate. This error cascades since it is necessary to compute a core line before the extent can be computed. This may produce gaps or may miss some vortices completely. Additionally, a vortex is sometimes not extracted as a single entity. That is, it may be extracted in several segments. Therefore, using a simple count of the vortices present as a quality metric may not be effective strategy. In these cases, maximizing the vortex length is probably the most effective metric. However, this metric is not without problems because it rewards the detection of false positives. In the sections that follow, we will use visual inspection to quantify the quality of our vortex characterizations since our primary objective is to eliminate missed vortices.

## B. Isolated Wing

We first consider the wing flow field simulation described by Luke et al.,<sup>9</sup> which is dominated by a wing tip vortex (Figure 10 top). Upon closer inspection, two merging vortices can be noted (Figure 10 bottom). The cell marking and topology identification algorithms struggle in the region where the two vortices come in close proximity to one another. This is partly due to the coarseness of the mesh there, and also due to the multiple clusters with three or more neighbors there. The branch points produce core lines in little pieces. However, the topology identification method does pick up both of the merging vortices, and has little trouble with the large wingtip vortex. There is likely an interaction between the primary vortex and the secondary vortices that is not captured in the relatively coarse mesh downstream of the wing.

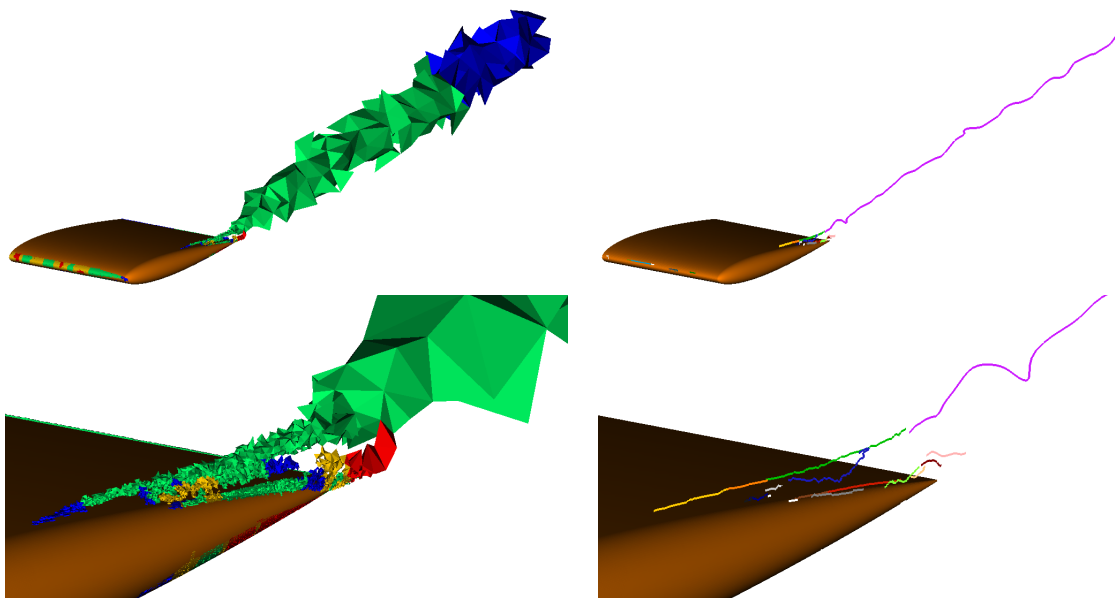


Figure 10. Isolated wing: wingtip vortex correctly identified as a single, complete, straight vortex (top). The two vortices that merge near the wing tip were also detected, although the method had difficulty in the merge region itself (bottom).

## C. Serrated Wing

The serrated wing, described by Hammons and Thompson,<sup>10</sup> gives rise to merging vortices, as well as vortices that originate in close proximity to one another. In the past we have had difficulty robustly detecting the four largest vortices. We have also had difficulty with the flow features near the nose due to their multi-branching nature. The problems were traced to aggregates with multiple branches and branch points with more than three branches. With the improvements to algorithm, we are now able to detect the four large vortices much

more robustly with less need for parameter tweaking (see Figure 11). We are also able to extract the vortex core line near the nose, seen in Figure 13. The nose vortex, which originates at the apex and terminates near the tip of the first serration, is still not complete. However, this is an improvement over parts of it being missing.

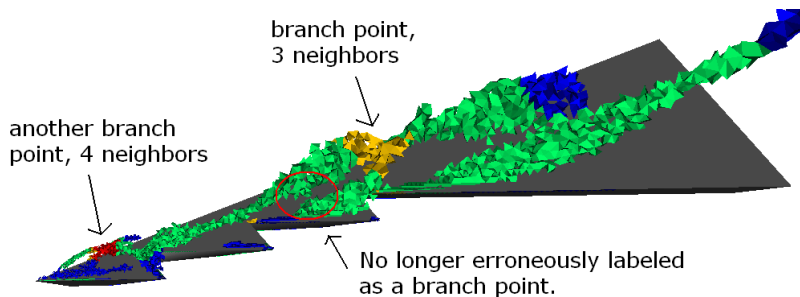


Figure 11. Improved performance of the clustering algorithm, part of vortex topology identification.

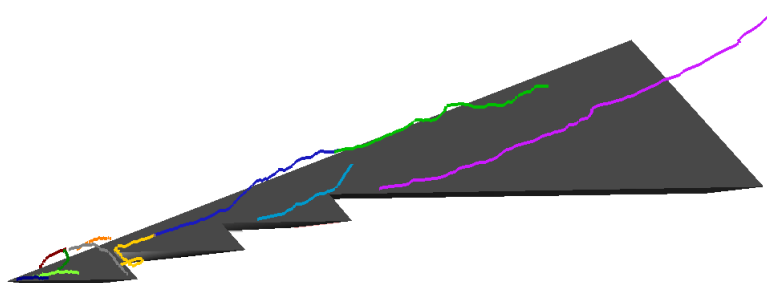


Figure 12. Complete vortex core lines extracted for four large vortices. Nose vortex extracted in pieces, but without missing pieces.

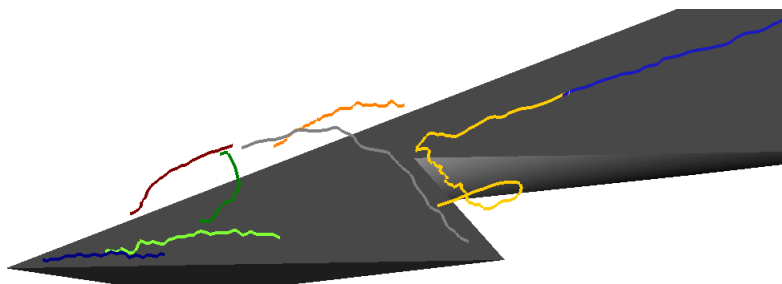


Figure 13. Closeup of the nose of the serrated wing. The vortex originating at the nose tip and ending at the serration tip has been extracted, although it is in two pieces.

#### D. CAWAPI

The F-16XL wing data, generated by Görtz and A. Jirásek,<sup>11</sup> has proven quite challenging. It contains many vortices that originate near each other. It also have vortices whose core lines are in close proximity to each other. This challenges both the cell marking and clustering algorithms. Note that while in parts of the data, cluster quality is high and neighbor counts are as we would expect, in other parts (colored red), the topology identification algorithm struggles (Figures 14 and 15). This results in vortex cores lines that are in little

pieces or have gaps (see Figure 16). Note that in the regions that were green in Figure 15, indicating that clustering worked correctly, we have long, continuous core lines.

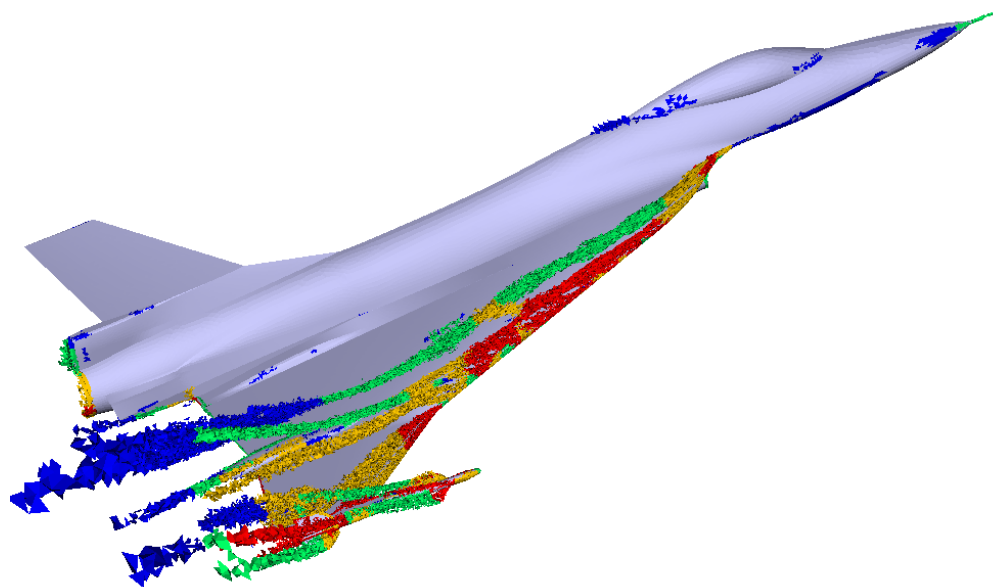


Figure 14. CAWAPI topology identification clusters. High quality clusters are perpendicular to vortex core lines, and have two neighbors unless they are at the end of a vortex or at a branch point.

## VI. Conclusions and Future Directions

The vortex topology identification technique still needs significant work, but shows promise. The improvements we have made to candidate cell marking (directional growth) and clustering (distance along vortex core line, larger clusters, correct neighbor counts) have improved the topology identification performance. The enhancement we've made to the topology algorithm itself, such as handling multiple branch points, have also improved our results in measurable ways. We now see more complete vortices, less vortices in pieces, less missing pieces, less gaps.

Even if the topology-challenges described in this paper were completely surmounted, we would still be faced with the pipeline challenge. Our vortex detection and characterization method is a pipeline composed of several stages. The input to the current stage is the output of the preceding stage. The vortex topology stage is preceded by the candidate cell marking and aggregation stages. Therefore, if those stages produce cells that are difficult to cluster, there is little the clustering and topology identification stages can do to produce high quality results.

One way to address this problem is to change the parameters that control candidate cell marking and aggregation. However, finding just the right set of parameters that give ideal candidate cell aggregates is tedious and is contrary to the objective of this effort: automatic characterization of the data. Sometimes, adequate cell marking is not possible due to the quality of the CFD data itself or simply the nature of the cell marking algorithm. We have made improvements to the cell marking and aggregation stages. Chief among them were hole filling and directional growth. The  $k$ -means clustering method performs better on cell aggregates without holes, or with very low porosity. Hole filling and directional aggregate growth both reduce porosity. The latter does so while preserving aggregate topology. Even with these changes, the clustering algorithm can still have difficulty. Another approach is to rethink the cell marking algorithm, for example, try a combination of scalar fields rather than just the swirl parameter.

Topology identification challenges still remain. We are currently unable to correctly subdivide diverging aggregates, diverging / remerging aggregates, and to recognize loops. Loop recognition is just an implementation matter, the others are a bit more difficult. We would like to expand the definition of "branch point" to include clusters where the sign of the normalized helicity changes. This would help us identify individual vortices in a counter-rotating pair.

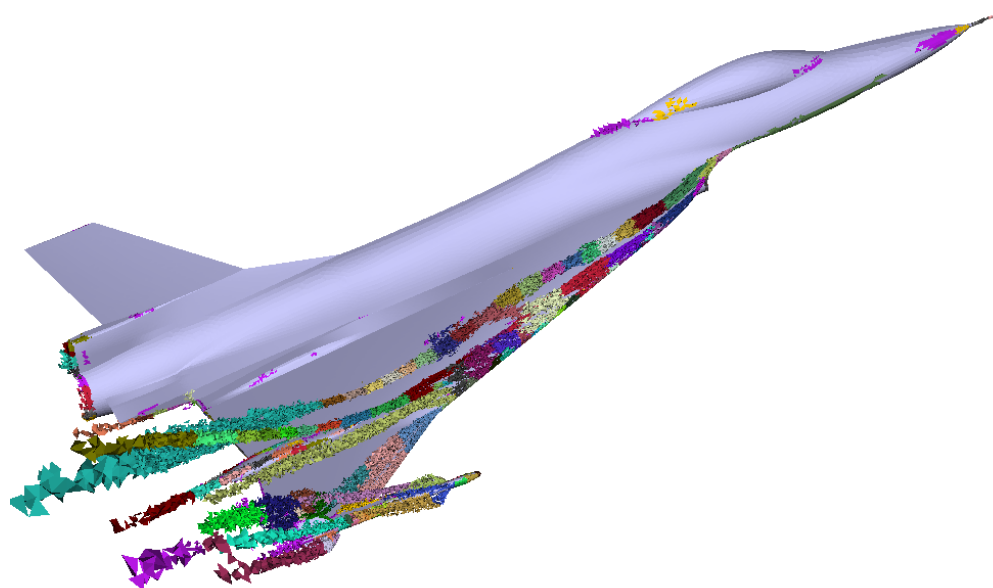


Figure 15. CAWAPI clusters colored by neighbor count. Note the large red area. This area does not get aggregated properly and produces core lines in pieces or with gaps.

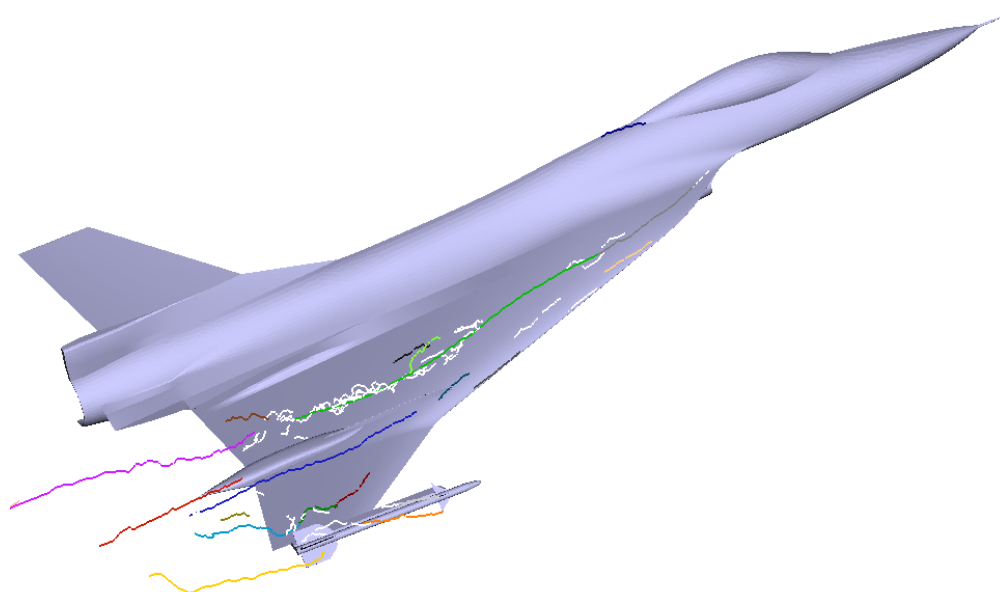


Figure 16. In the region where clustering and topology identification had difficulty, we see lots of little pieces of core lines, or missing lines. However, note that in the regions that were green in Figure 15, we see complete core lines (the long green, blue, magenta lines).



## VII. Acknowledgements

The effort is partially funded by the National Science Foundation and by the Department of Defense. We want to thank Adam Jirásek of the Swedish Defence Research Agency for providing the CAWAPI data.

## References

- <sup>1</sup>Silver, D. and Wang, X., "Tracking Scalar Features in Unstructured Datasets," *IEEE Visualization '98*, 1998, pp. 79–86.
- <sup>2</sup>Ma, K.-L. and Interrante, V., "Extracting Feature Lines from 3D Unstructured Grids," *IEEE Visualization '97*, 1997, pp. 285–292.
- <sup>3</sup>van Walsum, T., Post, F. H., Silver, D., and Post, F. J., "Feature Extraction and Iconic Visualization," *IEEE Trans. Visualization and Computer Graphics*, Vol. 2, No. 2, 1996, pp. 111–119.
- <sup>4</sup>Jankun-Kelly, M., Thompson, D., Brewer, W., Jiang, M., and Machiraju, R., "A Multistage Vortex Visualization Algorithm," *AIAA 44th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 9-12, 2006*, 2006.
- <sup>5</sup>Jankun-Kelly, M., Jiang, M., Thompson, D., and Machiraju, R., "Vortex Visualization for Practical Engineering Applications," *IEEE Trans. Vis. Comp. Graphics*, Vol. 12, No. 5, September / October 2006.
- <sup>6</sup>"Guide for the Verification and Validation of Computational Fluid Dynamics Simulations," Tech. Rep. AIAA G-077-1998, The American Institute of Aeronautics and Astronautics, 1998.
- <sup>7</sup>Kanungo, T., Mount, D. M., Netanyahu, N., Piatko, C., Silverman, R., and Wu, A. Y., "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, 2002, pp. 881–892.
- <sup>8</sup>MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations," *Fifth Berkeley Symp. Math. Statistics and Probability*, Vol. 1, 1967, pp. 281–296.
- <sup>9</sup>Luke, E., Hebert, S., and Thompson, D., "Theoretical and Practical Evaluation of Solver-Specific Mesh Quality (Invited)," *AIAA 46th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 7-10, 2008*, Jan. 2008.
- <sup>10</sup>Hammons, C. and Thompson, D., "A Numerical Investigation of Novel Planforms for Micro UAVs," *AIAA 44th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 9-12, 2006*, Jan. 2006.
- <sup>11</sup>Görtz, S. and Jirásek, A., "Steady and Unsteady CFD Analysis of the F-16XL using the Unstructured Edge Code," *AIAA 45th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 8-11, 2007*, Jan. 2007.
- <sup>12</sup>Jiang, M., Machiraju, R., and Thompson, D. S., "Detection and Visualization of Vortices," *Visualization Handbook*, Academic Press, 2004, pp. 287–301.
- <sup>13</sup>Roth, M., *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*, Ph.D. thesis, Swiss Federal Institute of Technology Zürich, 2000.
- <sup>14</sup>Banks, D. C. and Singer, B. A., "A Predictor-Corrector Technique for Visualizing Unsteady Flow," *IEEE Trans. Visualization and Computer Graphics*, Vol. 1, No. 2, 1995, pp. 151–163.
- <sup>15</sup>Sujudi, D. and Haines, R., "Identification of Swirling Flow in 3D Vector Fields," *AIAA 12th Computational Fluid Dynamics Conf., Paper 95-1715*, 1995.
- <sup>16</sup>Peikert, R. and Roth, M., "The "Parallel Vectors" Operator—A Vector Field Visualization Primitive," *IEEE Visualization '99*, 1999, pp. 263–270.
- <sup>17</sup>Roth, M. and Peikert, R., "A Higher-Order Method for Finding Vortex Core Lines," *IEEE Visualization '98*, 1998, pp. 143–150.
- <sup>18</sup>Levy, Y., Degani, D., and Seginer, A., "Graphical Visualization of Vortical Flows by Means of Helicity," *AIAA J.*, Vol. 28, No. 8, 1990, pp. 1347–1352.
- <sup>19</sup>Berdahl, C. H. and Thompson, D. S., "Eduction of Swirling Structure using the Velocity Gradient Tensor," *AIAA J.*, Vol. 31, No. 1, 1993, pp. 97–103.
- <sup>20</sup>Jeong, J. and Hussain, F., "On the Identification of a Vortex," *J. Fluid Mechanics*, Vol. 285, 1995, pp. 69–94.
- <sup>21</sup>Jiang, M., Machiraju, R., and Thompson, D. S., "A Novel Approach to Vortex Core Region Detection," *Joint Eurographics-IEEE TCVG Sym. Visualization*, 2002, pp. 217–225.
- <sup>22</sup>Haller, G., "An Objective Definition of a Vortex," *J. Fluid Mech.*, Vol. 525, 2005, pp. 1–26.
- <sup>23</sup>Bauer, D. and Peikert, R., "Vortex Tracking in Scale-Space," *Joint Eurographics-IEEE TCVG Sym. Visualization*, 2002, pp. 233–240.
- <sup>24</sup>Jiang, M., Machiraju, R., and Thompson, D. S., "Geometric Verification of Swirling Features in Flow Fields," *IEEE Visualization '02*, 2002, pp. 307–314.
- <sup>25</sup>Stegnaier, S., Rist, U., and Ertl, T., "Opening the Can of Worms: An Exploration Tool for Vortical Flows," *IEEE Visualization '05*, 2005, pp. 463–470.
- <sup>26</sup>Rütten, M., "Vortex Axis Calculation by Using Vortex Features," *AIAA 34th Fluid Dynamics Conference and Exhibit, Paper 2004-2353*, 2004.
- <sup>27</sup>Garth, C., Tricoche, X., Salzbrunn, T., and Scheuermann, G., "Surface Techniques for Vortex Visualization," *Joint Eurographics-IEEE TCVG Sym. Visualization*, 2004, pp. 155–164.
- <sup>28</sup>Perry, A. E. and Chong, M. S., "A Description of Eddy Motions and Flow Patterns using Critical Point Concepts," *Ann. Rev. Fluid Mechanics*, Vol. 19, 1987, pp. 125–155.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.