

Automated Extraction of Secondary Flow Features

Suzanne M Dorney*

NASA Marshall Space Flight Center, MSFC, AL 35812

suzanne.m.dorney@nasa.gov

Robert Haimes⁺

Department of Aeronautics and Astronautics

Massachusetts Institute of Technology, Cambridge, MA 02139

haimes@mit.edu

ABSTRACT

The use of Computational Fluid Dynamics (CFD) has become standard practice in the design and development of the major components used for air and space propulsion. To aid in the post-processing and analysis phase of CFD many researchers now use automated feature extraction utilities. These tools can be used to detect the existence of such features as shocks, vortex cores and separation and re-attachment lines. The existence of secondary flow is another feature of significant importance to CFD engineers. Although the concept of secondary flow is relatively understood there is no commonly accepted mathematical definition for secondary flow. This paper will present a definition for secondary flow and one approach for automatically detecting and visualizing secondary flow.

INTRODUCTION

The use of Computational Fluid Dynamics (CFD) has become standard practice in the design and development of the major components used for air and space propulsion. Many of today's advanced computer simulations create datasets containing as much as a billion pieces of information for the results of a single steady-state run. Clearly, transient simulations of the same spatial fidelity stress the

available computer resources. The sheer size of this data results in an exceedingly difficult and time-consuming analysis process.

The task of interrogation and interpretation of this information is required so that the knowledge contained within the simulation can be extracted. The problem is becoming more significant as improvements in computational performance result in these large-scale simulations becoming more commonplace. Today, computational performance is increasing an order of magnitude every 3.5 years. Simulations once used only for exploration are now available more frequently during design and parametric studies.

Traditional interactive visualization is used to probe the data in order to locate and identify physical phenomena, or to identify limitations in the simulation process. However as the frequency of the large-scale simulations increase in the design process, new approaches must be developed to enable the design engineer to process the information in a timely fashion. Specifically there needs to be closer integration of the traditional analysis stages (pre-processing, solver and post-processing). One scenario is to employ visualization throughout the simulation process.

* Computer Scientist.

⁺ Principal Research Engineer.

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

- **Exploration Time** -- Interactive exploration of large-scale CFD datasets is laborious and consumes hours or days of the scientists/engineers time.
- **Field Coverage** -- Interactive visualization techniques produce output based on local sample points in the grid or solution data. Important features may be missed if the user does not exhaustively search the dataset.
- **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.

It is clear that these tools do not directly answer the questions of the CFD investigator. An expert is required to infer the underlying fluid field topology from the imagery supplied. Getting a more specific answer is required. Direct fluid feature extraction has the following advantages over these exploratory visualization tools:

- **Deterministic Algorithms** -- If there are no "parameters" that the user need adjust, then no intervention is required.
- **Fully Automated** -- The analysis can be done off-line in a batch computation. It can be used directly by a solver to adapt the mesh to better resolve the feature.
- **Local Analysis** -- These schemes, where possible, perform only local operations. Therefore, the computations for each cell are independent of any other cell and may be

performed in parallel. This is clearly advantageous in distributed memory compute arenas.

- **Data Reduction** -- The output geometry is several orders of magnitude smaller than the input dataset. This is an important characteristic for the size of a resultant output. High fidelity spatial and temporal results of the feature extraction can be stored on disk for non-interactive co-processing environments. This is usually not possible for the entire transient simulation. The side effect is that playback is rapid (the extraction process is done and the data has been distilled to salient information). Also, better than a movie, viewpoint can be changed during playback.
- **Quantitative Information** -- Precise locations for the flow features are extracted. Also, classification and measures of strength can be reported.

The results of feature extraction can be viewed in a 3D interactive visualization environment and can be used in conjunction with interactive visualization tools. Feature extraction tools are now being used in parametric studies where tens or hundreds of simulations are run for the same design with subtle changes to the structure or flight conditions. These tools can be used to detect the existence of such features as shocks [1], vortex cores [2], recirculation zones [3], boundary layers [4] and separation and re-attachment lines [5]. All of these feature extraction algorithms have been collected together into a single software toolkit: **FX** [6].

The existence of secondary flow is another feature of significant importance to CFD turbomachinery engineers. The identification of secondary flow can highlight areas of stress and loss. Although the concept of secondary flow is relatively understood there is no commonly accepted mathematical definition for secondary flow. Because of this it has been extremely difficult to develop an automated feature capability for the identification of secondary flow. This paper will present a formal definition for secondary flow and one approach for automatically detecting and visualizing secondary flow. In addition to the definition of secondary flow this paper will discuss how such an automated feature extraction utility was developed and used in the post-processing analysis of CFD simulations. Of particular interest will be how this tool was used to add insight to the final analysis.

BACKGROUND

The concept of secondary flow in turbomachinery is generally thought of as any flow that is not in the direction of the primary flow. An example of this vague definition is shown in fig 1. The vortices shown in the figure are examples of secondary flow as the primary flow direction is directly between the blades. Although the concept of secondary flow is generally understood it is not well defined. Some attempts at rigorously defining this idea include:

- ...component of absolute vorticity in the direction of the relative streamline [7].
- Secondary flow in broad terms means flow at right angles to intended primary flow [8].
- Due to viscous effects, end walls divert primary flow produced by blades and vanes; to give rise to what has come to be called secondary flow [9].

Of the three definitions listed above only [8] provides a definition that could be made operational. What is required is the notion of primary flow, which we can define. Unfortunately by the time we get a CFD solution the notion of *intended* is lost.

The desire to view perturbations from the *primary* flow direction can give insight into the viscous, reverse flow and vertical effects that deviate from the design. To this end it is obviously desirable to be able to generate two-dimensional vector plots that display the secondary flow given a traditional CFD solution.

Secondary flow plots are usually displayed in a passage between blades or just downstream from the trailing edge. The arrows are generated from a frame of reference that is relative to the passage in question (i.e. absolute for fixed rows and moving for rotors). This points to a difficulty in areas between stators and rotors: what is the appropriate frame of reference? Because of this ambiguity the frame of reference is specified by the user in the final implementation.

ALGORITHM

It would clearly be desirable to have a scheme that could maximize the primary flow through a constructed surface. This could be done by defining a *pivot* point in the channel that reflects the centroid of the passage or flow. A surface that goes through the point can be generated. By adjusting the position of this surface the *best* fit can be found. This surface can then be used to view the secondary flow by projecting the vector field data onto the surface.

The goal here is to calculate the mass-averaged quantities in the channel. This should be done on a grid plane or a cut through the passage that is orientated so that all bounds of the cut are walls (if possible). The following can be done with either a plane (all surface facets have the same normal \vec{n}) or an analytical surface where the normal for cut facets can change.

Compute surface integrals:

$$\begin{aligned}\bar{A} &\equiv \iint \vec{n} dA \\ \bar{M} &\equiv \iint [\rho \vec{q} \cdot \vec{n}] dA \\ q_0 &\equiv \bar{M} / \bar{A} \\ \bar{X}_0 &\equiv \iint [\vec{X} \rho \vec{q} \cdot \vec{n}] dA / q_0\end{aligned}$$

where \bar{M} is the mass-averaged flux, q_0 is the mean velocity and \bar{X}_0 is the mass-averaged center of the flow.

Newton-like Iteration to Maximize Primary Flow

By selecting various cuts that pass through \bar{X}_0 we can adjust the normal (in the case of a simple plane) in an iterative loop so that we maximize q_0 (the velocity perpendicular to the plane):

$$\vec{n} = |q_0|$$

Note that a new set of surface integrals is computed during each iteration. This can also change the position \bar{X}_0 .

Using a planar cut this technique takes about 3 to 4 iterations to converge (i.e. the normals returned differ by some suitably small factor). This Newton-like convergence is most always seen unless the planar cut is adjusted so that a new portion of the flow field is exposed.

When converged, this provides a view of the data that displays secondary flow when the normal velocity component is removed.

DISCUSSION

In practice this algorithm works well but did require a number of operational adjustments. These included:

Passage of Interest

The fast cut algorithms are based on Marching Cubes and are performed on a single 3D element at a time. The result is a set of disjoint polygons that reflect the portion of the surface that cuts through the cell. The notion of where in the domain the fragments come from is usually lost. So if the simulation contains more than a single passage the cut data can easily contain fragments from elsewhere in the simulation. This will corrupt the primary flow calculation in that we are no longer focused on a single passage.

The solution is to reconnect the fragments into complete (and bounded) surfaces. Once this is done a seed point can be located within the bounded surfaces so that one can be selected. Only those polygons that are within the selected region are used in the calculations.

The cut algorithm used constructs the surface in a Finite-Element sense (that is, a list of nodes that reflect the 3D edges being cut is constructed and the polygons refer to indices in that list). The reconnection is performed via a polygon side-matching algorithm based on the indices (not floating point locations). This is unique and robust. Any side that is seen by two polygons is interior to the region. A side with only a single polygon is bounding the region.

Multi-block simulations

In multi-block simulations the volumes represented by the blocks can abut or overlap. The individual cell definitions are usually block specific so that even if the blocks maintain a larger contiguous volume, it is usually not apparent by the time one looks at the fragments from the Marching Cubes results. When reconnecting the regions the results will reflect the block boundaries and not the actual bounds of the cut. The regions need to be placed back together.

When performing streamlining, it is traditional to use the "IBlack" data to inform the software how the blocks are connected. When one pierces a cell on a face where the "IBlack" data indicates that a jump to another block is required, the "IBlack" index contains the accepting block. Initially, this data was used to attempt to flood the region from the target surface fragments to connecting blocks. This was found to be unreliable.

A much more expensive technique was developed. This involved producing a bounding-box around each region as a first step. All regions (that have not been included) and have bounding boxes that overlap the start region are examined. Each point on the exterior

of the start region is compared to all fragments of the candidate regions. If it is found that any point is interior, then the new region is considered part of the calculation and this process is then recursively applied where this candidate becomes the start region.

In this way the seed point fills all connecting and overlapping areas and the calculation can proceed on that "passage".

Tip leakage simulations

When performing the secondary flow algorithm on a simulation that displays tip leakage there is a natural connection between passages. With the algorithm described above there will be flooding into other passages. This will corrupt the primary flow calculation.

This problem has been taken care of if the simulation is multi-block and there are individual blocks that represent the tip flow regions. The flooding can be "dammed" by informing the technique not to use certain blocks as candidates.

Frame of reference

In multi-stage calculations care needs to be taken so that the algorithm sees data in a consistent frame of reference. This means that when looking at the secondary flow in a rotor, all velocity field values should be in the rotating frame. It is important that the data in the stators be transformed so that the technique does not see any discontinuities in the velocity field.

This then means that if one were to traverse the machine from upstream to down that there will be a number of changes of reference. These should be done while the resultant planar cut is in the zone between blades.

IMPLEMENTATION

The algorithms for the detection of secondary flow features were implemented in C in the Feature eXtraction toolkit FX [6]. This toolkit, unlike most visualization systems, is lightweight because no drawing and/or GUI functions are supported. In general, the input is the CFD solution and output is various forms of geometry. The following entry point has been added:

```
FX_MEANFLOW(XPOS, VNORM, DAM)
```

float XPOS[3] -- On input the position that sets the plane given the normal VNORM. On output,

the mass averaged position on the planar cut is returned.

float VNORM[3] -- On input the normal that sets the family of planes to use to produce the cut. On output VNORM is filled with the mass averaged velocity through the cut.

int *DAM -- Pointer to the status of each block (for multi-block cases) to act as a "dam" for the flooding procedure. Zero indicates that flooding through the block is OK, a one is the flag to NOT use this block. NOTE: may be NULL to specify no "damming".

This subroutine, given the start position and plane normal, computes the mass-averaged "center" of flow and the mass averaged velocity. This is essentially a single iteration in the maximization of the primary flow through the surface. Fixing the pivot point is controlled by the position specified in XPOS. The position can be made stationary by resetting the values to the fixed position after each iteration. This will allow for the examination of secondary flow in many contexts (for example, picking points along a streamline, one can get the sense of the secondary flow as seen by a traveling fluid "particle").

A GUI was then developed using C++, OpenGL, and QT to interact with FX for the specification and eventual visualization of the secondary flow features. This system was used to generate all of the images shown in this paper.

Fig 2 shows the initial window that is displayed upon reading in the input file. The input file contains the names of the geometry and solution files as well as specifications on which surfaces to display. The figure shows not only the specified surfaces but also the outline of each grid in the system. Fig 3 shows the initial specification of the secondary flow plane. The plane is displayed in blue and the yellow cross hair indicates the initial pivot point. Fig 4 then shows the calculated secondary flow plane in its position relative to the two blades. Fig 5 shows a close up view of the calculated secondary flow plane. The contours shown on the plane are those of density. The specifications on what to display are made using the panel shown in fig 6.

RESULTS

This system was first tested on the results of an analysis for an injector chamber and then used to analyze the results of a simulation of an axial turbine from a turbopump. Fig 7 shows the initial geometry of the injector chamber and the injector holes. This simulation was done on one quarter of the complete

chamber. Figures 8 and 9 show the secondary flow vectors projected onto a density contour surface near the injectors. An animation was generated that showed how the secondary flow diminished as the flow moved down the chamber. The application of this system to the injector chamber is important because the amount of secondary flow and swirl (i.e., flow mixing) within the chamber controls the burning characteristics within the chamber.

The system was also used on a distinctly different geometry, that of an axial-flow turbine stage from a notional rocket turbopump. Fig 10 shows three images that highlight the flow through one rotor passage of the turbine. The image on the left is a vector plot that shows the direction of the primary, or core flow. The image in the middle is the result of calculating the plane that accounts for the primary flow at a location near midchord of the rotor. This plane contains the projected vectors (overlaid on contours of density) that represent the secondary flow in this region. The vectors are anchored at the black tuft locations and the direction of flow is away from the tuft. The image on the right is a close up view of the secondary flow. The flow in this image is consistent with the expected results found in the diagram shown in Fig 1. Both figures show a horseshoe vortex which emanates from the hub endwall. In addition, Fig. 10 highlights the tip clearance leakage vortex. Figure 11 shows a similar set of results further downstream. The position of the plane is nearly perpendicular to the surface of the rotor. The secondary flow vectors shown in the right image clearly show the tip clearance leakage vortex and movement of the endwall vortices towards midspan. An animation was also created that shows the secondary flow for a series of pivot locations along the suction side of the rotor blade. Figure 12 shows one frame from this animation. The image on the left shows both a stator and a rotor blade to establish the location of the primary flow plane within the simulation. The image on the right is a close up view of the resulting primary flow plane displaying secondary flow vectors.

CONCLUSIONS

A set of tools has been developed to aid in the detection and visualization of secondary flow features in CFD results. The interactive tool provides for a platform to display several aspects of the geometry and the flow solution. The ability to determine the primary flow direction and then display secondary flow features has been found to be helpful in the analysis of a turbo-pump test chamber.

REFERENCES

1. Lovely, D. and Haimes, R., "Shock Detection from Computational Fluid Dynamics Results", AIAA Paper 99-3285, June 1999.
2. Sujudi, D. and Haimes, R., "Identification of Swirling Flow in 3-D Vector Fields", AIAA Paper 95-1715, June 1995.
3. Haimes, R., "Using Residence Time for the Extraction of Recirculation Regions", AIAA Paper 99-3291, June 1999.
4. Baskett, L. and Haimes, R., "Feature Extraction of Shear Layers", AIAA Paper 2001-2665, June 2001.
5. Kenwright, R., "Automatic Detection of Open and Closed Separation and Attachment Lines", Proceedings of IEEE Visualization '98, October 1998.
6. Haimes, R., "FX - Fluid feature eXtraction toolkit", <http://raphael.mit.edu/fx>.
7. Hawthorne, W. R., Cambridge University CUED/A-Turbo/TR 63, 1974.
8. Cumpsty, N. A., 'Compressor aerodynamics' Longman ISBN 0-582-01364-X page 316, 1989.
9. Bradshaw, P., 1996, The Bradshaw quote is from the paper by L.S. Langston (reference 2) 'Secondary Flows in Axial Turbines -A Review' annals New York Academy of Sciences May 2001.

CASCADE ENDWALL FLOW STRUCTURE

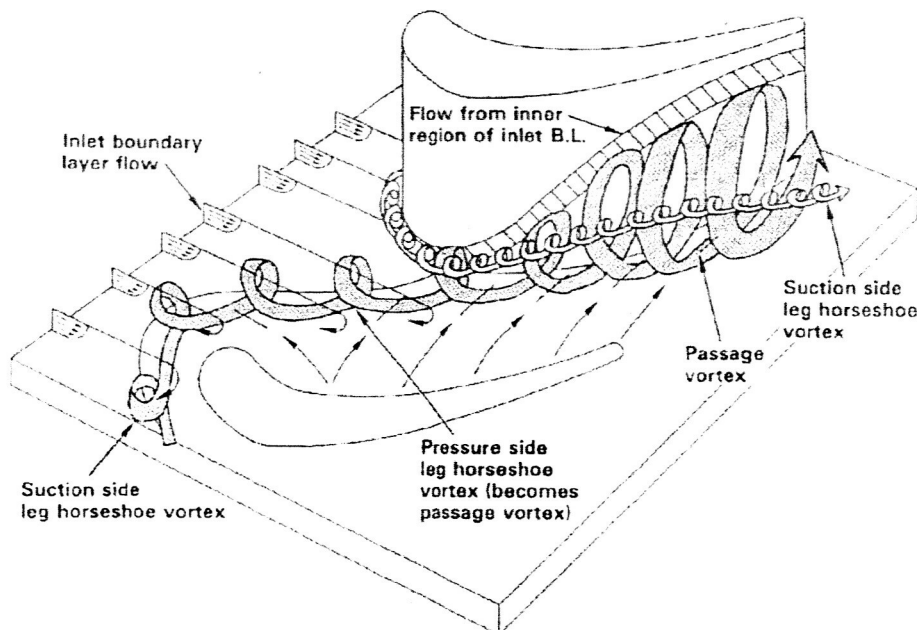


Figure 1: Example of Secondary Flow

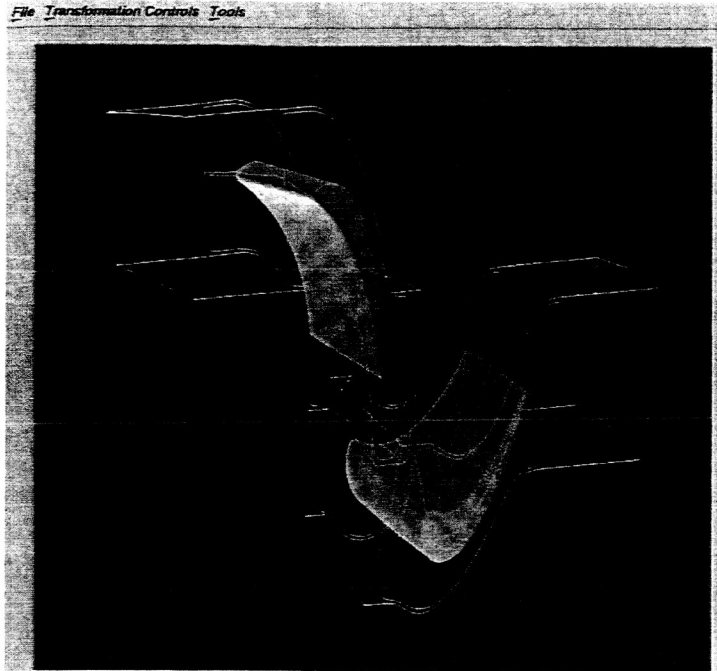


Figure 2: Initial Image

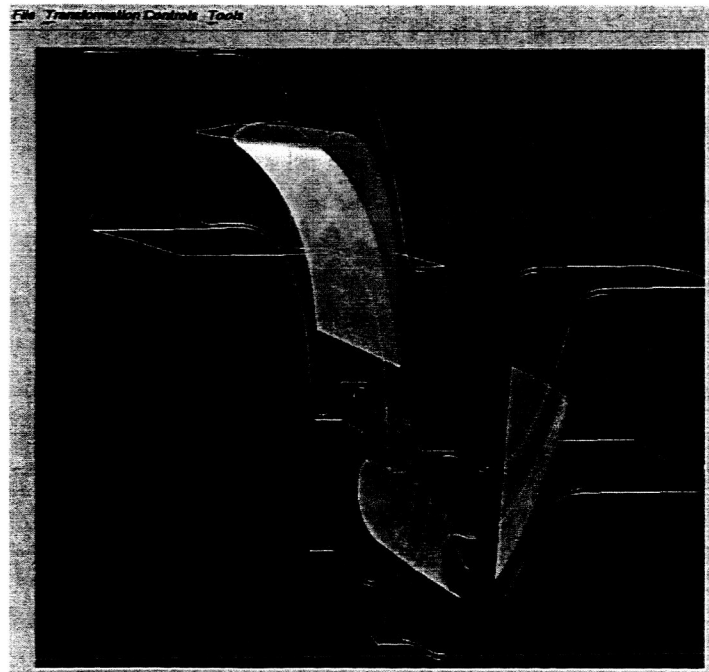


Figure 3: Initial Position and Pivot Point of Secondary Flow Plane

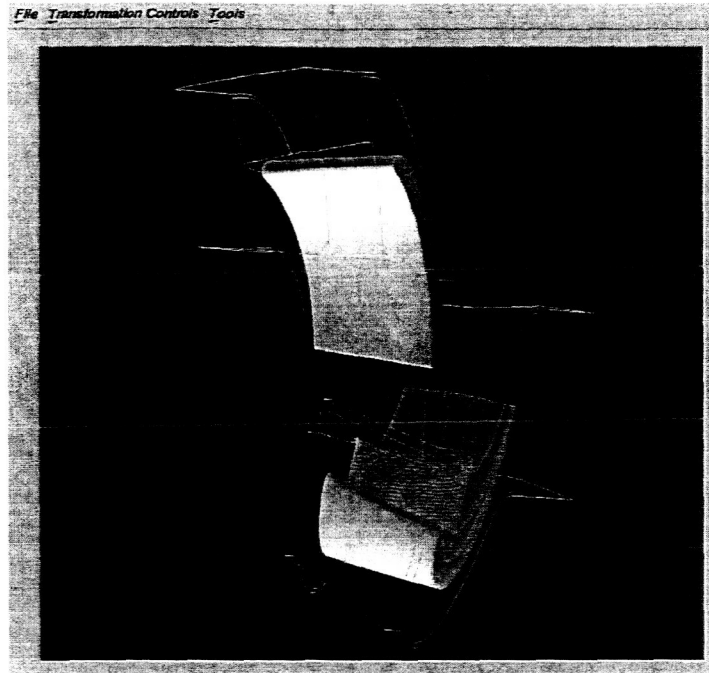


Figure 4: Calculated Secondary Flow Plane

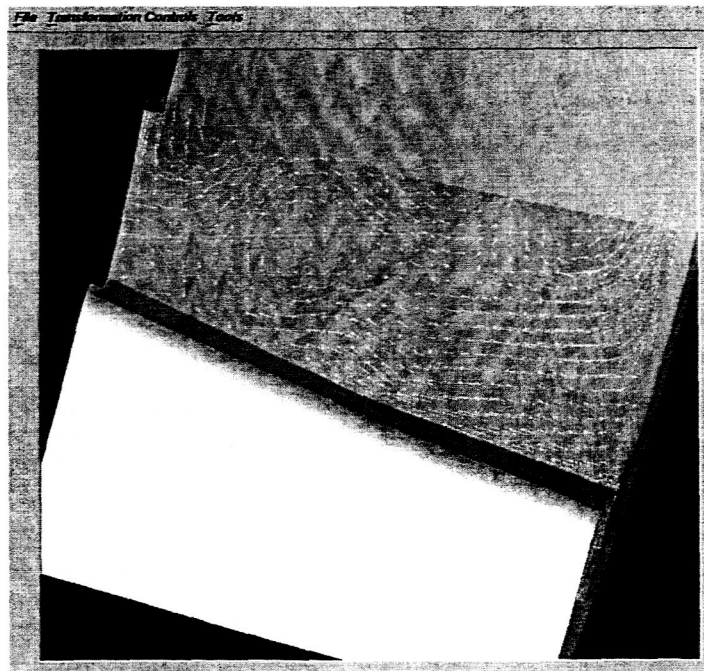


Figure 5: Close Up View of Vortices

Secondary Flow Plane Definition Panel

Secondary Flow Plane Panel

Current Plane
 1

Plane Location

Plane Normal Vector

Grid Size

Number of Grid Points

Vector Length Factor

Display Options

☒ Show Plane

☒ Show Point

☒ Show Flow

☒ Show Contours

Type of Flow Vectors
☒ Projected ☐ 3 Dimensional

Color of Tuft Base
☒ Black ☐ Gray ☐ White

Figure 6: Control Panel

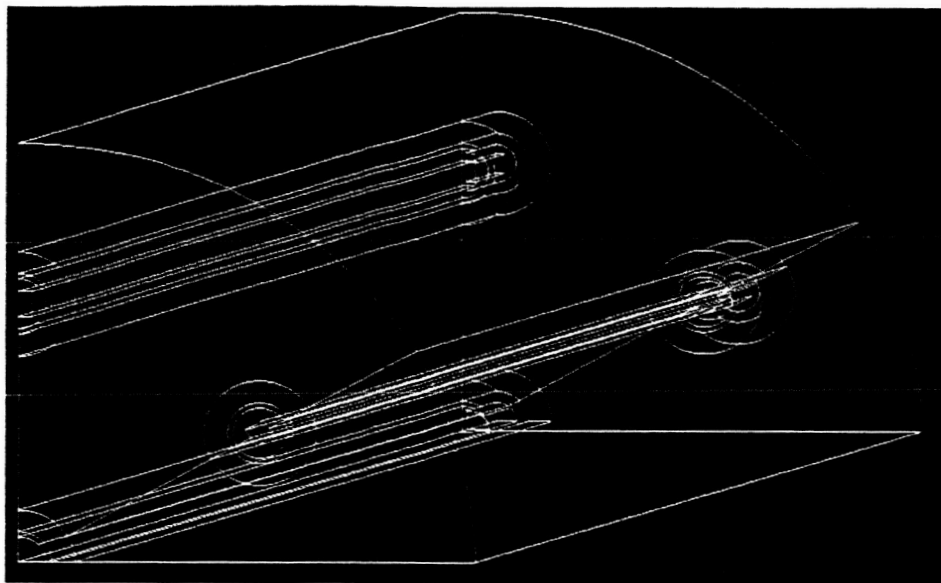


Figure 7: Geometry of Injector Chamber

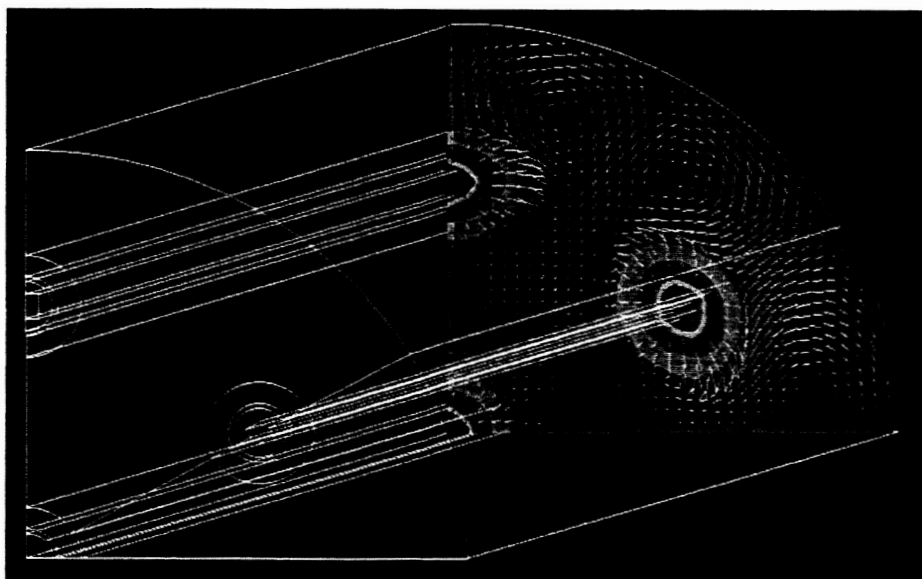


Figure 8: Secondary Flow Vectors Projected onto Density Contour Surface

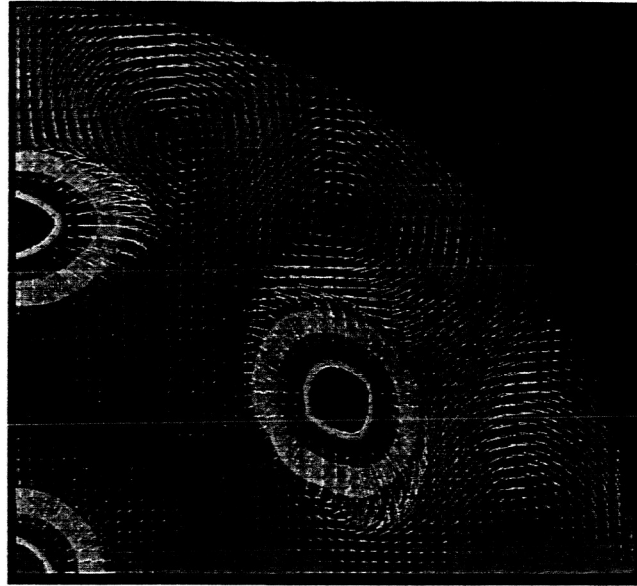


Figure 9: Full Image of Secondary Flow Vectors

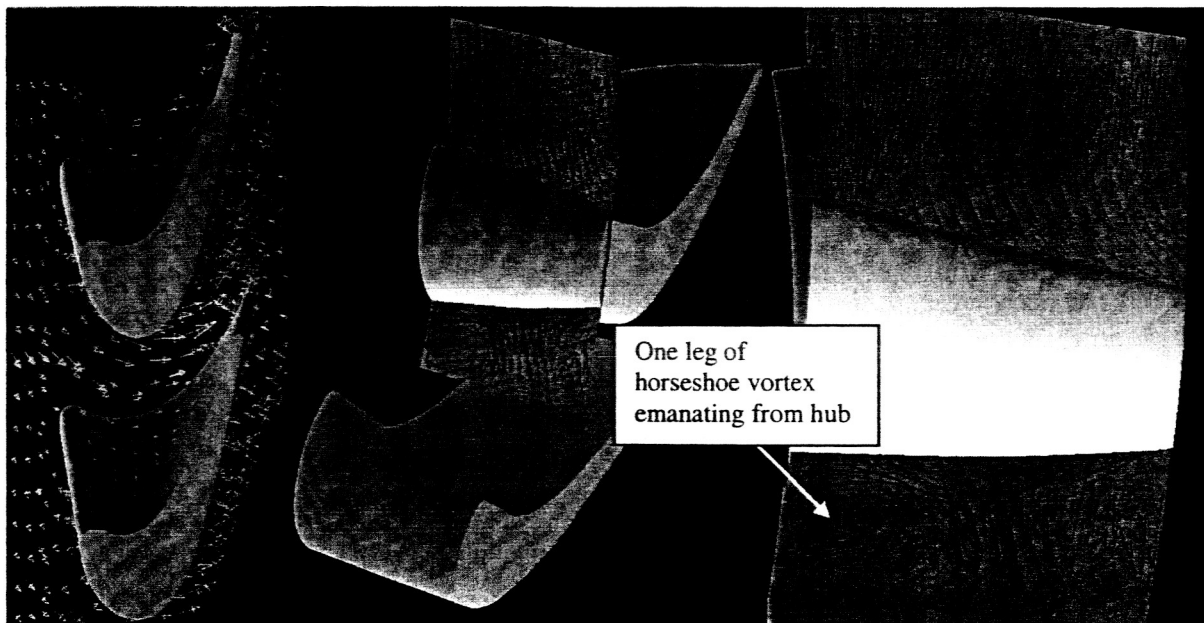


Figure 10: Vectors Showing the Primary flow Direction (left), Position of Secondary Flow Plane (middle) Secondary Flow Image (right)

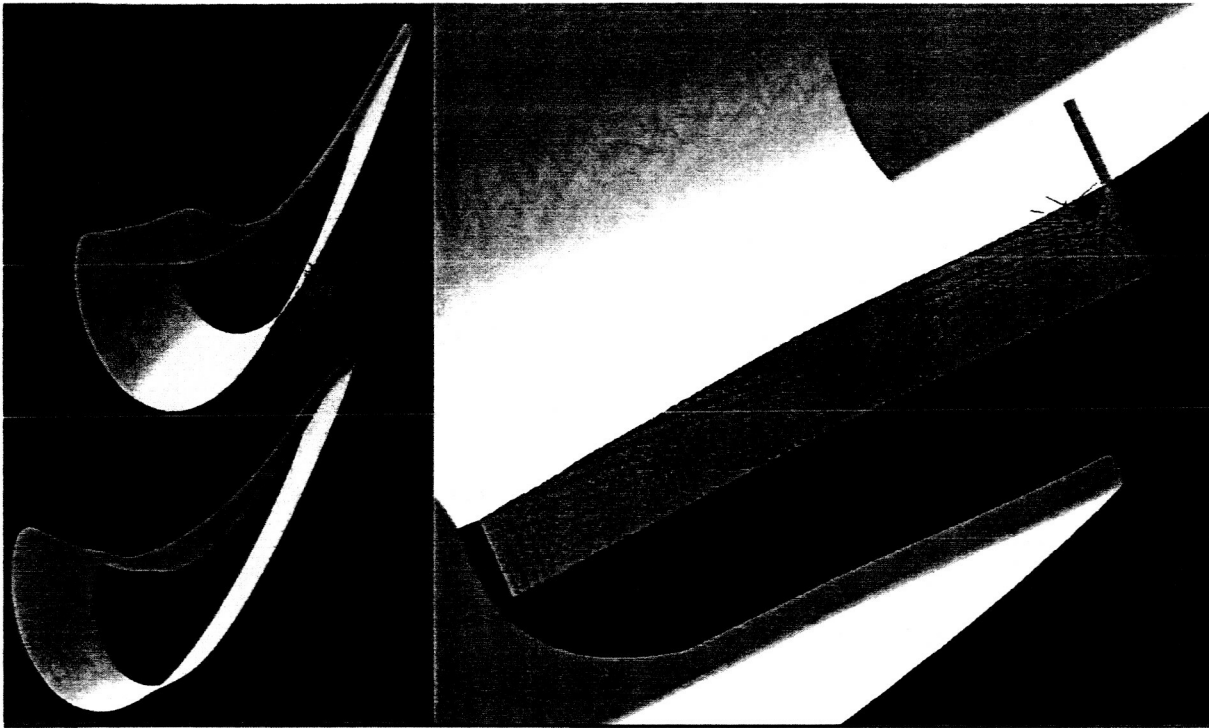


Figure 11: Position of Primary Flow Plane (left) and Close Up View of Secondary Flow Results (right)

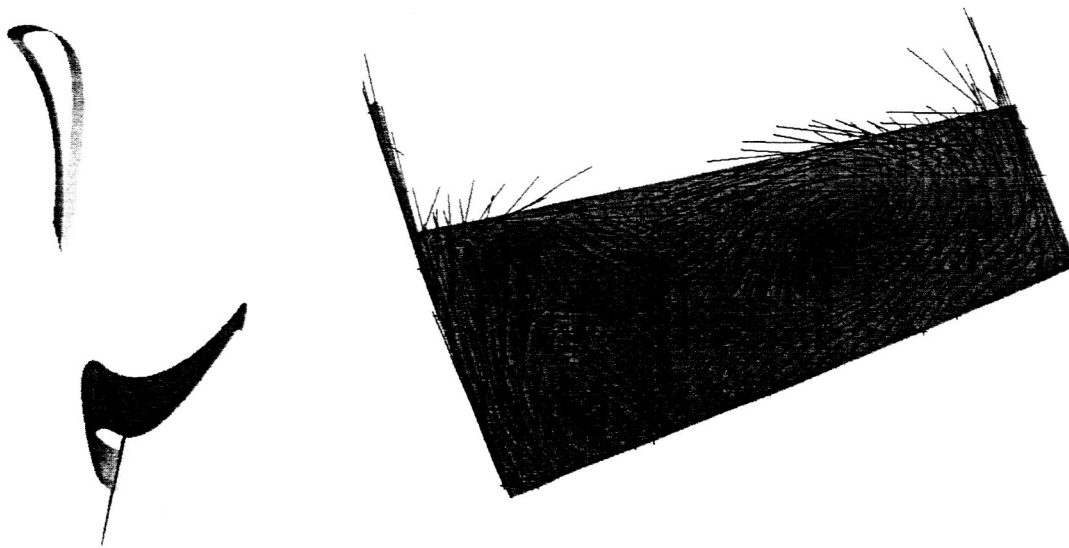


Figure 12: Image from Secondary Flow Animation