

The Stream Polygon: A Technique for 3D Vector Field Visualization

W. J. Schroeder C. R. Volpe W. E. Lorensen
GE Corporate Research & Development
Schenectady, New York 12301

Abstract

We present a new method for the visualization of 3D vector fields called the stream polygon: a regular, n-sided polygon, oriented normal to the local vector. The polygon can represent local deformation due to rigid body rotation and both normal and shear strain. In addition, the effects of translation and scalar functions may be represented by sweeping the stream polygon along the streamline, and by appropriately varying the radius and shading the surface of the resulting streamtube. In this paper we develop a mathematical foundation for the stream polygon, and provide examples with application to velocity field visualization.

1 Introduction

Scientific and engineering analysis programs produce a variety of information that challenges the visualization process. No single representational technique can hope to provide insight into all types of analysis data. In three and four dimensional analysis, both input geometric representation and computed results vary depending on the application and the type of analysis. To enhance the understanding of the results of these analyses, scientific and engineering visualization systems must provide tools that can show scalar and vector quantities in the context of the underlying geometry.

Scalar data, represented as a single value at each computational point, is often represented as a color that varies over some continuous range. Scalars can also be used to warp a geometry along a user-specified vector. Contours, for 2D analyses, and iso-surfaces, for 3D analyses, are other useful graphical representations for scalar information. These facilities are available,

in one form or another, in most visualization systems.

Vector data, with three or more values per computational point, is another common form of engineering and scientific data. Typical examples include the velocity field due to fluid flow, or the displacement field as a structure is loaded.

There are three common techniques for visualizing vector data. These are hedgehogs[Sta89], streamlines[FM73], and domain deformation. Hedgehogs represent the vector field by drawing oriented, scaled lines along the direction of the local vector. Streamlines are paths that are everywhere tangent to the vector field, and are often thought of as representing the path that a massless particle would take in a fluid. Domain deformation represents the vector field by distorting the local geometry according to the vector data. For example, to represent the displacement of a structure, the image would be distorted according to the vector field representing displacement. Each of these techniques has proven its value for visualizing and understanding data. However, these techniques fail to provide an understanding of the local deformations that exist within the vector field.

Non-uniform vector fields give rise to local deformation consisting of both rigid body motion as well as strain. Rigid body motion includes local translation (motion in the direction of the local vector) and rotation. Strain, which can be considered the local deformation not due to rigid body motion, consists of both normal strain (uniform deformation in a particular direction) and shear, or angular deformation.

The three vector field visualization techniques described above generally represent local deformation due only to translation. Another technique proposed by Volpe[Vol89] represents rotation by using an oriented streamribbon (as compared to a streamline). The streamribbon is a strip of polygons, or ribbon,

whose edges lie along two streamlines. Alternatively, the streamribbon can be constructed by placing the strip of polygons along a streamline, and orienting the surface according to the rigid body rotation. None of these techniques directly represents strain, which for many applications can be quite important. For example, many materials cannot effectively support shear strain. Hence when performing structural analysis these areas must be identified and removed if possible.

This paper presents the stream polygon, a new 3D vector visualization technique that can show both local deformation and strain. The stream polygon is a regular n -sided polygon oriented normal to the local 3D vector. The local effects of rigid body motion and strain may be represented by rotating and deforming the polygon. In addition, by sweeping the stream polygon along the streamline, and shading the surface and appropriately varying the radius of the resulting streamtube, the effects of translation, vector magnitude, and scalar functions may be represented as well.

2 Characterizing the Vector Field

Before describing the stream polygon, it is necessary to characterize the vector field. Here we are interested in describing local deformation, as well as techniques for computing the local vector and derivatives at any point in the vector field.

2.1 Local Deformation

Consider the vector field \mathcal{V} consisting of the m local vectors $\mathbf{v} = (u, v, w)$. To examine the local deformation due to \mathcal{V} at the point $\mathbf{x} = (x, y, z)$, a first order Taylor's series expansion of \mathbf{v} about \mathbf{x} is constructed. Then the local deformation e_{ij} is given by [Bat67, Die76]

$$e_{ij} = \varepsilon_{ij} + \omega_{ij}. \quad (1)$$

where ε_{ij} is the local strain tensor, and ω_{ij} is the local rotational tensor.

Here we assume that \mathbf{v} represents a displacement field. For general vector fields, \mathbf{v} may represent many possible data, hence the terms deformation, strain, and rigid body motion must be interpreted accordingly. For example, if \mathcal{V} represents fluid flow, then these terms become deformation rate, strain rate, and velocity, respectively.

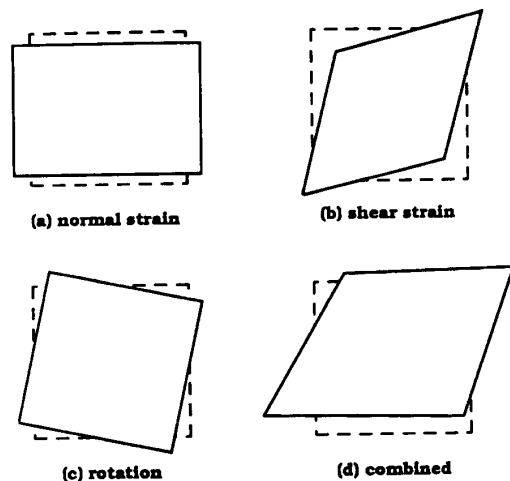


Figure 1: Components of local deformation.

The local strain is given by

$$\varepsilon_{ij} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) & \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) & \frac{\partial v}{\partial y} & \frac{1}{2} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\ \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) & \frac{1}{2} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) & \frac{\partial w}{\partial z} \end{bmatrix} \quad (2)$$

The terms along the diagonal of ε_{ij} are the normal components of strain. Examining a plane oriented in the $x - y$ plane, normal strain causes uniform deformation along the $x - y$ axes as shown in Figure 1(a). The terms off the diagonal are the shear components of strain. Shear strain causes angular deformation as illustrated by Figure 1(b).

The local rigid body rotation is given by

$$\omega_{ij} = \begin{bmatrix} 0 & \frac{1}{2} \left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) & \frac{1}{2} \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) & 0 & \frac{1}{2} \left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y} \right) \\ \frac{1}{2} \left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z} \right) & \frac{1}{2} \left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \right) & 0 \end{bmatrix} \quad (3)$$

The rotation tensor describes the local rigid body rotation as shown in Figure 1(c). By adding the contributions of the normal and shear strain and the rotation, the total deformation can be also represented as illustrated in Figure 1(d).

Note that the local rotation tensor can also be written

$$\omega_{ij} = -\frac{1}{2} \varepsilon_{ijk} \omega \quad (4)$$

where ϵ_{ijk} is the alternating tensor, and ω is the rotation vector

$$\omega = \begin{bmatrix} \frac{\partial v}{\partial z} - \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \end{bmatrix} \quad (5)$$

In fluid flow where \mathcal{V} is the velocity field, ω is the vorticity and represents the rate of angular velocity of the flow at a point. Another important flow parameter is stream vorticity, Ω , or rotation about the local vector \mathbf{v} given by the normalized dot product

$$\Omega = \frac{\mathbf{v}\omega}{|\mathbf{v}||\omega|} \quad (6)$$

2.2 Computing Derivatives

In most applications the vector field \mathcal{V} is known only at m discrete points, and an interpolation function is required to compute the derivatives at the arbitrary point \mathbf{x} in \mathcal{V} . We assume that the geometry can be represented as a union of many non-overlapping elements, or cells, that are simple shapes such as hexahedron or tetrahedron[Sch]. It is only at the vertices, or nodes, of the cells that \mathcal{V} is known. Although many choices of interpolation function are possible, the iso-parametric formulation common to finite element analysis[Zie77] provides many advantages including simplicity of formulation, an abundance of prior work, common use in the analysis process, and the same formulation used regardless of element topology.

The geometry of each element is described by

$$\mathbf{x} = \sum_{i=1}^p N_i \mathbf{x}_i \quad (7)$$

where \mathbf{x}_i are the node points of the element, and N_i are the shape functions, one per element node. The shape functions vary depending upon element topology; for a hexahedron the shape functions are given in terms of the element coordinates $\xi = (\xi, \eta, \zeta)$

$$N_i = \frac{1}{8}(1 - \xi_i)(1 - \eta_i)(1 - \zeta_i) \quad (8)$$

with $\xi_i, \eta_i, \zeta_i = \pm 1$ at the element nodes, and $-1 \leq \xi, \eta, \zeta \leq 1$ inside the element. Note that at a particular element node j we choose ξ_j, η_j, ζ_j such that $N_i = 1$ when $i = j$, $N_i = 0$ otherwise.

In an isoparametric formulation, the interpolation functions N_i are the same for the element geometry as

well as the nodal variable, in this case the vectors \mathbf{v}_i . Hence the vector field in the element is given as

$$\mathbf{v} = \sum_{i=1}^p N_i \mathbf{v}_i \quad (9)$$

The local derivatives can be computed from Equations 7,8, and 9 as

$$\partial \mathbf{v} / \partial \mathbf{x} = \mathbf{J}^{-1} \partial \mathbf{v} / \partial \xi \quad (10)$$

with \mathbf{J} the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \quad (11)$$

and

$$\partial \mathbf{x} / \partial \xi = \sum_{i=1}^p (\partial N_i / \partial \xi) \mathbf{x}_i \quad (12)$$

$$\partial \mathbf{v} / \partial \xi = \sum_{i=1}^p (\partial N_i / \partial \xi) \mathbf{v}_i \quad (13)$$

2.3 Computing the Stream Line

Computing the streamline is straightforward. The basic approach is to integrate the equation

$$\mathbf{x}(t) = \int_{t_0}^t \mathbf{v}(s) ds. \quad (14)$$

In fluid flow, $\mathbf{v}(t)$ is the velocity and t is time, and the path generated can be considered the motion of a massless particle in the velocity field.

Generally the integration is performed numerically using interpolation functions such as those described in the previous section, and a numerical integration scheme such as the Euler or Runge-Kutta methods[Cd72]. As the integration proceeds, it is necessary to track the streamline as it moves through the cells, requiring repeated transformation from global to local coordinates. This transformation is performed by solving Equation 7 explicitly for $\xi(\mathbf{x})$ [LA90] or using a numerical technique such as Newton's method[Cd72].

3 The Stream Polygon

Consider a regular n -sided polygon whose center is located at position \mathbf{x} and normal to the local vector

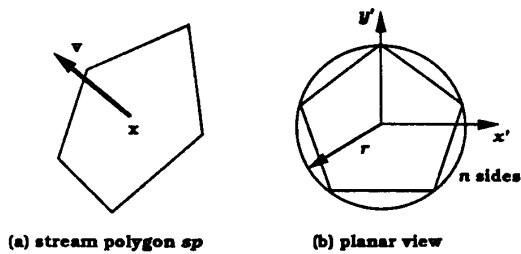


Figure 2: The stream polygon sp .

\mathbf{v} in vector field \mathcal{V} . This polygon is called the *stream polygon* sp as shown in Figure 2(a). The radius of sp is defined as the radius r of the circumscribing circle of the polygon. The parameters r and the number of sides n (Figure 2(b)) are constrained by

$$\begin{aligned} 0 \leq r \leq r_{max} \\ 0 \leq n \leq n_{max} \end{aligned} \quad (15)$$

where r_{max} and n_{max} are arbitrarily chosen finite values and n_{max} is an integer.

The stream polygon provides a number of simple, yet powerful techniques for creating graphical representations of 3D vector fields. The goal is to show the usual translational effects, as well as the effects of local strain and rotation. In addition, we wish to represent scalar information.

The first technique is to deform sp according to the components of local strain and rotation. Equations 2 and 3 can be combined to yield the standard transformation matrix \mathbf{T} containing effects due to scaling (normal strain), shear, and rotation. Then sp is deformed by transformation \mathbf{T} .

It is also possible to project local strain onto the plane defined by the point \mathbf{x} and the local vector \mathbf{v} . For example, if the number of sides is $n = 4$ (i.e., the polygon is a square), the local deformation can be represented exactly as shown in Figure 1. Here the contributions of the strain and rotation are immediately apparent, as well as the combination of all three. Another useful technique described by Equation 6 is to project the vorticity ω onto the local vector \mathbf{v} . The resulting angle Ω is the rotation of sp about \mathbf{v} .

Another effective use of the stream polygon is to vary the radius of the stream polygon according to some prescribed relationship with vector magnitude or other scalar function. For example, if \mathcal{V} represents

the velocity of an incompressible flow with no shear, then the equation

$$r(\mathbf{v}) = r_{max} \sqrt{\frac{|\mathcal{V}_{min}|}{|\mathbf{v}|}} \quad (16)$$

represents an area of constant mass flow. Here r_{max} is a user specified radius at the minimum flow velocity v_{min} .

Probably the most striking use of the stream polygon is to sweep it along a streamline to produce the *streamtube*. The streamtube is composed of n sides corresponding to the n sides of the stream polygon. Each side is a surface that can be colored according to a specified scalar function. By combining the effects noted previously, namely varying radius and transforming sp , a large amount of information can be represented in a simple, intuitive manner.

As a final note, if $n = 0$ or $r = 0$, sp is a point and if swept in \mathcal{V} results in a streamline. When $n = 1$, sp is a line and if swept in \mathcal{V} results in a streamribbon. Hence the stream polygon is a generalization of the streamline and streamribbon.

4 Application

One application of the stream polygon is to place multiple instances of the stream polygon along a streamline, as shown in Figure 3. The stream polygon may represent a single component of deformation or possibly combinations of deformation mode. It is also possible to locate the stream polygon along the streamline, and generate the 2D strain projection as described in the previous paragraph. By allowing the user to position the stream polygon interactively, it is possible to rapidly move along the streamline, viewing the local strain at any point.

Our primary use of the stream polygon is to generate streamtubes using the techniques described in the previous section. In our implementation of the swept stream polygon in the LYMB/VISAGE[SLY⁺] visualization system, we implemented the streamtube using offset ribbons. These ribbons can be placed at a specified radius (possibly variable) from an arbitrary line (generally a streamline). The width and location of the ribbon around the line is controlled by specifying a chord angle and start angle, and the ribbon rotates with the normal to the line. Also, when creating a streamtube, if the number of sides $n = 0$ a streamline is generated, when $n = 1$ a ribbon is generated, and when $n = 2$ two ribbons are generated, each perpendicular to the other in a + cross section.

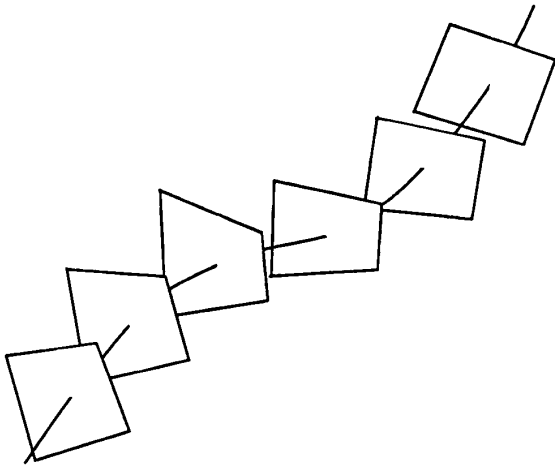


Figure 3: Deformed stream polygons along streamline.

A simple example of the streamtube is shown in Figure 4. Here a closed loop has been constructed with a regularly varying vorticity vector to generate the twisted ring shown. The stream polygon is a hexagon, and each of the six sides of the tube are colored with the same regularly varying scalar function. The effects of local strain are not shown.

A variant of the previous example is shown in Figure 5. Here a stream polygon with $n = 49$ generates the streamtube shown. Every seventh side of the tube is colored with a different scalar function to create the stripes on the tube.

We have applied these techniques to many practical data sets. For example, in Figure 6 the structured grid *blunifin*¹, is used to generate the streamline, streamribbon, and streamtube shown. Again the stream polygon is a hexagon, but in this case alternate faces of the streamtube are colored with the two scalar functions flow temperature and flow pressure. We have found that using more than three scalar functions simultaneously is generally not effective, since too much information is present, and much is hidden from the viewer. Also notice in this example that the radius of the tube varies with the magnitude of the flow velocity. This is clearly seen as a bulge in the tube as the flow impinges on the fin.

Figure 7 is an example of streamtubes in a section

¹Widely available structured grid data from NASA Ames.

of an annular combustor. The data was generated using finite difference techniques. The geometry of the combustor is shown as a wire mesh. A hexagonal stream polygon is used to generate the tube shaded with flow density, and with varying radius. It is interesting to note how the shorter streamtube opens into a trumpet shape as it approaches the surface of the combustor where the flow velocity approaches zero.

As a final example, consider Figures 8 and 9. The analysis is of the airflow a room containing some bookcases and desks. The room contains a heat and pollution source (i.e., cigarette) near the closer of the two desks, and has an air inlet duct (black square, lower left) and exhaust duct (black square, upper left). On the wall opposite the vents is a window. The purpose of the analysis is to determine the effectiveness of the ventilation system.

In Figure 8, a streamtube is started in the vicinity of the inlet, travels through the room, is exits through the exhaust vent. The tube is colored with room pressure, and the radius varies according to Equation 16. In Figure 9, another streamtube is started in a slightly different location, but this time meanders throughout the room. Eventually, the tube ends because the velocity of the air is reduced below a specified terminal velocity. In this figure the radius of the tube is not varied.

5 Conclusions

The stream polygon is a powerful display technique for visualizing 3D vector fields. It can be used to display local deformation including normal and shear strain and rigid body rotation. Streamtubes may also be generated by sweeping the stream polygon along the streamline. Rotating the stream polygon, varying the radius, and shading the surface of the streamtube with scalar functions provide other simple but powerful visualization techniques.

Acknowledgements

Scott Lamson and Graham Holmes of GE CR&D provided some physical insight into vector field analysis in computational fluid dynamics.

References

- [Bat67] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, England, 1967.

- [Cd72] S. D. Conte and C. de Boor. *Elementary Numerical Analysis*. McGraw-Hill, New York, 1972.
- [Die76] G. E. Dieter. *Mechanical Metallurgy*. McGraw-Hill, Inc., New York, 1976.
- [FM73] R. W. Fox and A. T. McDonald. *Introduction to Fluid Mechanics*. John Wiley and Sons, Inc., New York, 1973.
- [LA90] R. Löhner and J. Ambrosiano. A vectorized particle tracer for unstructured grids. *Journal of Computational Physics*, 91:22-31, 1990.
- [Sch] W. J. Schroeder. Cell abstraction for visualization of diverse data types. In preparation.
- [SLY+] W. J. Schroeder, W. E. Lorensen, B. Yamrom, G. D. Montanaro, and C. R. Volpe. An object-oriented, interpreted environment for visualization, animation, and graphics. In preparation.
- [Sta89] Stardent Computer Inc., 95 Wells Avenue. *Application Visualization System User's Guide*, 1989.
- [Vol89] G. Volpe. Streamlines and streamribbons in aerodynamics. Technical Report AIAA-89-0140, AIAA, 1989.
- [Zie77] O. C. Zienkiewicz. *The Finite Element Method in Engineering Science*. McGraw-Hill Book Co., London, third edition, 1977.



Figure 4: Stream polygon swept along closed path with uniform rotation.

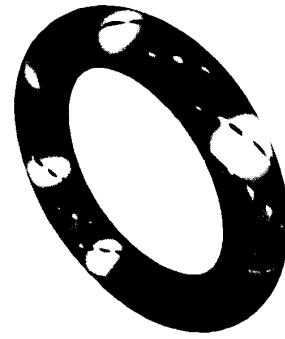


Figure 5: Streamtube with $n = 49$ sides. Every seventh side is colored with a different scalar data.

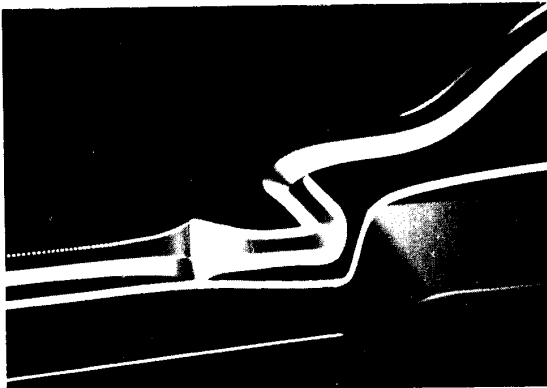


Figure 6: Streamline, streamribbon, and stream tube representing vector field in *bluntnfin*.

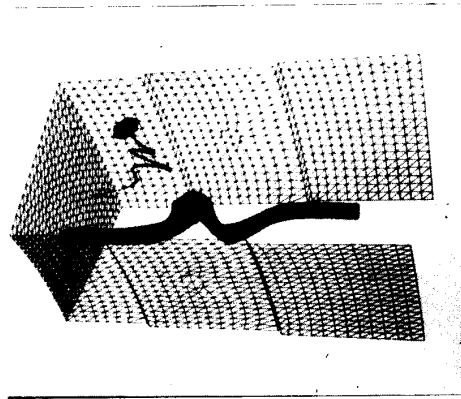


Figure 7: Two stream tubes shown in an annular combustor.

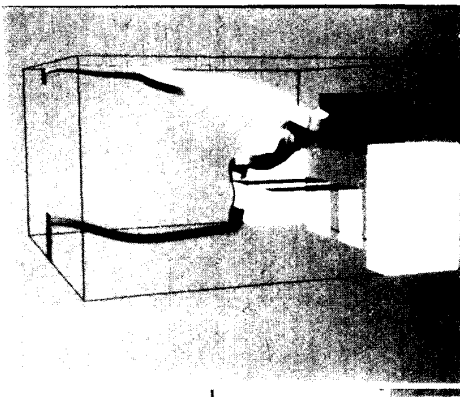


Figure 8: Visualizing air flow in a room.

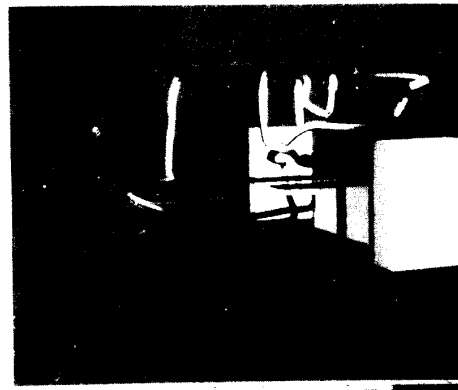


Figure 9: A meandering streamtube.

(See color plates, page 417.)

The Stream Polygon: A Technique for 3D Vector Field Visualization, W.J. Schroeder, C.R. Volpe, and W.E. Lorensen, pp. 126–132.



Figure 4: Stream polygon swept along closed path with uniform rotation.



Figure 5: Streamtube with $n = 49$ sides. Every seventh side is colored with a different scalar data.

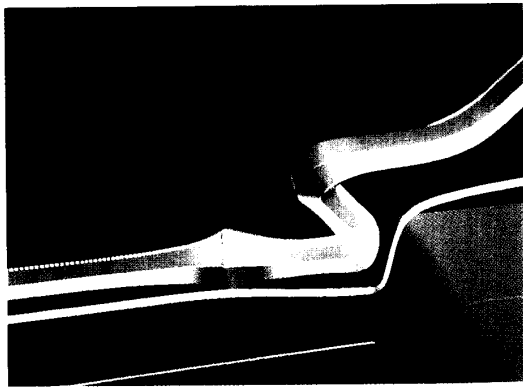


Figure 6: Streamline, streamribbon, and stream tube representing vector field in *blunffin*.

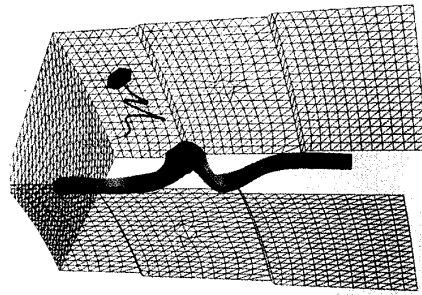


Figure 7: Two stream tubes shown in an annular combustor.

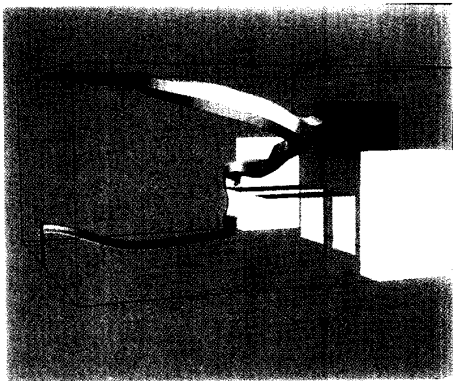


Figure 8: Visualizing air flow in a room.

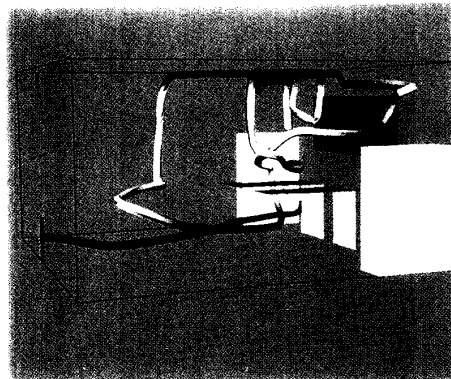


Figure 9: A meandering streamtube.