Principal Stream Surfaces

Wenli Cai*
Fraunhofer Institute for Computer Graphics

Pheng-Ann Heng†
Dept. of Computer Science & Engineering
The Chinese University of Hong Kong

Abstract

The use of stream surfaces and streamlines is well established in vector visualization. However, the proper placement of starting points is critical for these constructs to clearly illustrate the flow topology. In this paper, we present the principal stream surface algorithm, which automatically generates stream surfaces that properly depict the topology of an irrotational flow. For each velocity point in the fluid field, construct normal to the principal stream surface through the point. The set of all such normal vectors is used to construct the principal stream function, which is a scalar field describing the direction of velocity in the fluid field. Volume rendering can then be used to visualize the principal stream function, which is directly related to flow topology. Thus, topology in a fluid field can be easily modeled and rendered.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image generation; I.4.3 [Image Processing]: Enhancement.

Keywords: flow field, visualization, volume rendering, filtering

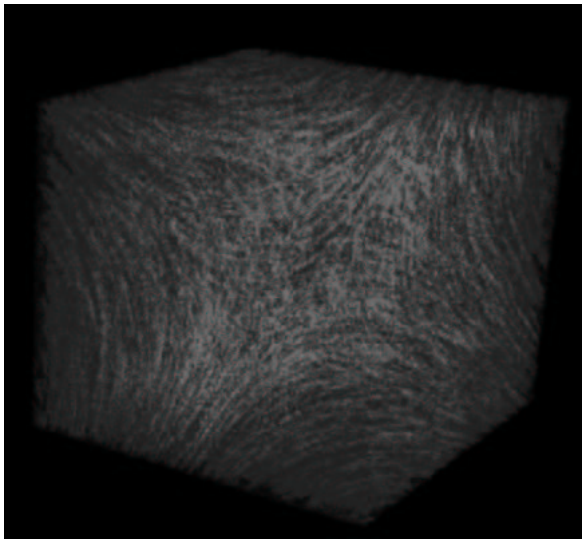

Figure 1: 3D Linear Integral Convolution

1 Introduction

The generation of stream surface [7] and streamline [4], as well as particle tracing [17] and probes [15] are well known techniques for vector visualization, and there are many papers devoted to the subject (see, for example,[1, 9]). A drawback of using stream surfaces and lines is that starting points need to be selected carefully; only certain starting points will trace out stream surfaces that properly describe flow topology. Determining such starting points is very difficult, particularly when dealing with unknown fluid fields. A recent paper[12] discusses an effective method (using an energy function) for addressing this problem in two dimensions.

Several methods have been proposed to visualize the flow topology without tracking stream surfaces from starting points, to avoid the selection of starting points placements. Helman and Hesslink[6] attempted to extract the topology of the fluid field directly using 'separation surfaces', modeled with stream surfaces tracked from critical points. But the construction of such surfaces is neither easy nor fully automatic. Another way to view flow topology is using texture mapping, including the linear integral convolution (LIC) method of Cabral and Leedom[2] and the spot noise[13] method of van Wijk. Applying textures to depict flow orientation is very useful in 2D fields, but not nearly as successful in three dimensions. We have attempted to expand the texture method to 3D fluid fields using 3D linear integral convolutions and volume rendering, but as shown in Figure 1, the results are not very good; too many details obscure the flow topology.

The concept of a stream function[16] is a fundamental concept in computational fluid dynamics (CFD). In a 2D fluid field, streamlines correspond to lines of constant $C = \Psi(X)$, where Ψ is the stream function. In a 3D fluid field, the constant C corresponds to a set of stream surfaces and a streamline corresponds to the intersection of two stream surfaces ($C_1 = \Psi_1, C_2 = \Psi_2$). Kenwright and Mallinson[8] first applied stream function to streamline tracking, and Van Wijk followed with an 'implicit stream surface' way to calculate stream function distributions by simulating the process of placing 'ink' on the inlet of the fluid field. All points in the fluid field are tracked backward to the inlet and the value at the corresponding inlet point is assigned to the tracked point. But the evaluation of a stream function on the inlet boundary and the detection of the inlet boundary are very difficult especially for unknown fluid field. Thus, the difficult problem with implicit stream surfaces is similar to the placement of starting points, due to the fact that it relies on tracking in preprocessing. However, the implicit stream surface introduces a new method to model stream surface and a general way to visualize flow topology directly, using the stream function and volume rendering instead of tracking. Fruhauf[5] proposes a raycasting method for direct rendering of vector fields; his work focused on streamline shading. He proposed a view-dependent stream-line computing method for illuminating streamlines. The rendering result is sensitive to view direction and light direction.

There is only one streamline passing through any particular velocity point in a fluid field, but infinitely stream surfaces that contain it. How may we determine which ones provide the most useful information about flow topology? As illustrated in Figure 2 about a cylindrical flow along the Z axis, we intuitively prefer certain surfaces to others. For example, most people would choose *Surface1* over *Surface2* or *Surface3*, to indicate the flow past point P. We would like to depict the flow topology using stream surfaces that show the flow motion to the greatest extent, that is, we care about the changing of velocity more than the velocity itself. *Surface1* is, in fact, tangent to rectifying planes in the Frenet frame at all passing points. We refer to this kind of stream surface as a principal stream

* wcal@igd.fhg.de
† pheng@cse.cuhk.edu.hk

surface.

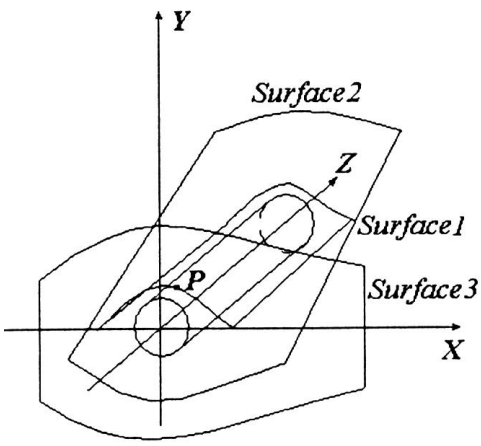

Figure 2: Different stream surfaces passing Point P around a cylindrical flow

The following section describes a way to automatically model the principal stream surface, in order to view flow topology. Briefly, the principal normal vectors in the Frenet frame are used to calculate the principal stream function (the stream function of the principal stream surface). Since these normal vectors constitute, by definition, the gradient of the principal stream function, it is straightforward to construct the scalar field of the principal stream function from them. The process is similar to the reconstruction of a potential field and only a single reference point is required. After the principal stream function is generated, the flow topology can be visualized directly with volume rendering[10, 3] or other iso-surface extraction methods[11].

The next section talks about concepts related to principal stream surfaces and functions. We follow with an algorithm to calculate the principal stream function, and finally a discussion of the implementation and experiments.

2 Related Concepts

In a Frenet frame, two tangent planes passing point P are chosen: the osculating plane and the rectifying plane (Figure 3). These two planes form a basis (via linear combination) for any other tangent plane. The normals of the osculating plane and rectifying plane are, respectively, B and N

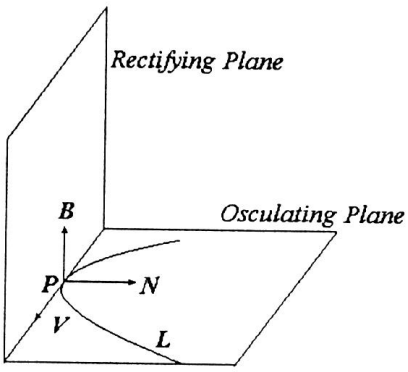

Figure 3: Frenet Frame

$$B = \frac{V \times A}{|V \times A|}, \qquad N = B \times V \tag{1}$$

where V is the first-order derivative of curve L at point P, and A is the second-order derivative. If L is thought of as an integral curve in the velocity field, that is, the unique streamline passing point P, then V is the velocity vector and A is the acceleration.

The osculating and rectifying planes are tangent to different stream surfaces at point P. The osculating plane is the nearest tangent plane to L, and the rectifying plane is the farthest (where nearest and farthest refer to the high order infinitesimal distance). These two planes show different attributes of the velocity at that point; the osculating plane depicts the velocity itself whereas the rectifying plane depicts more information about the changing of the velocity nearby. Other stream surfaces passing through P can be expressed as linear combinations of these two extreme cases.

The placement of stream surfaces passing through P is determined by their starting points. The most meaningful stream surface is the one that depicts the local topology, that is, the primary direction of velocity and its nearby rate of change. The integral curve L lies within the osculating plane, the tangent line to L lies within the intersection of these two planes, and the normal vector to L lies along the direction of curvature of the streamline which is the direction of the changing of the velocity.

As shown in Figure 2, plane XY based on the origin gives no hint about the 3D flow topology although it is a stream surface, since no fluid flows across the XY plane. Plane XY is therefore a typical osculating plane. Surfaces similar to *Surface1* are more helpful in viewing the flow topology, because it depicts the flow topology more clearly than other stream surfaces.

Definition 1 A *principal stream surface* is a stream surface whose tangent planes are the rectifying planes of the streamlines within it. The normal of each point on the principal stream surface has the same direction as the streamline's principal normal vector, and its magnitude is defined to be the magnitude of velocity at that point.

In a 3D fluid field, a stream surface is defined as the set of all points for which a particular stream function is a constant. The velocity at any point is the crossproduct of two stream functions (generated by the intersection of two stream surfaces): see[16, 8, 14]. In the Frenet frame,

$$V = N \times B \tag{2}$$

Definition 2 The principal stream function is the stream function of the principal stream surface, and its gradient is the normal of the principal stream surface.

Existence of Principal Stream Function The relationship between the velocity field $V(u, v, w)$ and two stream functions f, g at a point is[16]:

$$pV = \nabla f \times \nabla g$$

where p is the density value at that point. For an incompressible flow, p is a constant. According to the conservation of mass,

$$\nabla(pV) = \nabla(\nabla f \times \nabla g) = 0$$

f and g must satisfy

$$\begin{aligned} V \cdot \nabla f &= u f_x + v f_y + w f_z = 0 \\ V \cdot \nabla g &= u g_x + v g_y + w g_z = 0 \end{aligned}$$

The constant surfaces of f and g are stream surfaces in the field. There exist a global f and a global g in the whole velocity field,

but normally it is difficult to represent f and g as some elementary functions.

The principal stream function field can also be expressed as a function of f and g. Since f and g exist globally, the total differential of rectifying planes of the field can be expressed as the composite function of ∇f and ∇g. Thus, the principal field is integrable.

3 Principal Stream Surface

In order to model the principal stream surface, we must first compute its normals, the gradients of the corresponding principal stream function. Based on the gradient and a reference point, the principal stream function (a scalar field) can be constructed.

3.1 Normal of a Principal Stream Surface

According to Def. (1) and Eq. (1), the second-order derivative of streamline at point P is:

$$A = \frac{dV}{dl} = \frac{V(P_+) - V(P_-)}{\triangle l} \tag{3}$$

where P_+, P_- are, respectively, the previous and next discretized points on the streamline, parametrically separated from P by distance $\triangle t$.

Using the fixed Euler algorithm for ordinary differential equations, we specify:

$$\begin{aligned} P_+ &= P + \frac{V + V_+}{2} \cdot \triangle t, \\ P_- &= P - \frac{V_- + V}{2} \cdot \triangle t, \\ V_+ &= V(P + V \cdot \triangle t), \\ V_- &= V(P - V \cdot \triangle t) \end{aligned} \tag{4}$$

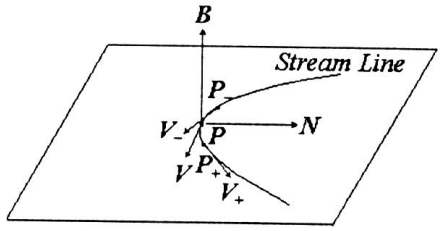

Figure 4: Normal of Principal Stream Surface

We define $\triangle l$ as the absolute distance between P_+ and P_- along the streamline, and approximate it as:

$$\begin{aligned} \triangle l &= \frac{V + V_+}{2} \cdot \triangle t + \frac{V + V_-}{2} \cdot \triangle t \\ &= V \cdot \triangle t + \frac{V_+ + V_-}{2} \cdot \triangle t \end{aligned} \tag{5}$$

3.2 Construction of a Principal Stream Function

By finding a normal to the principal stream surface, we find (up to a scalar multiplier field) the gradient of the principal stream function. Since the stream function is a scalar field, calculating the principal stream function is similar to reconstructing a potential field from its gradients at grid points.

The gradient of the principal stream function is assigned, by definition, the same magnitude as the fluid velocity at any point (see [16] for a similar process in 2D). Similarly, the magnitude of second-order derivatives of the principal stream function equals the second-order derivatives of the velocity field.

The Taylor expansion of the stream function around point P along a line is (this line is not the streamline, but a step along the construction direction):

$$F(P + \triangle P) = F(P) + F'(P) \triangle P + \frac{1}{2!} F''(P) \triangle P^2 + \cdots \tag{6}$$

where $F'(P)$ is the gradient of the stream function at point P, and $F''(P)$ is its Hessian matrix.

Equation (6) allows us to calculate the principal stream function: the only problem is finding a reference point for the fluid field. For simplicity, we select the first point $(t = 0)$ of the principal stream function as the reference point, and set it equal to $(0, 0, 0)$ in Cartesian coordinates.

3.3 Outline of the Algorithm

We use only Cartesian coordinates in this paper. Each element of the fluid field is a cube with edges parallel to the three coordinate axes. The algorithm scans the whole fluid field and constructs the distribution of the principal stream function, thus generate a 3D scalar field (scan volume).

The outline of the algorithm is,

for $k = Z_0$ *to* Z_n *do*
for $j = Y_0$ *to* Y_n *do*
for $i = X_0$ *to* X_n *do*
  $C = Cube(i, j, k)$;
  reconstruct velocity V_f *in* C*:*
  for $kk = 0$ *to* $ScaleZ$ *do*
  for $jj = 0$ *to* $ScaleY$ *do*
  for $ii = 0$ *to* $ScaleX$ *do*
    $P = (i + ii/ScaleX, j + jj/ScaleY, k + kk/ScaleZ)$
    PP = *previous point of* P
    $V = V_f(PP)$ *calculate velocity by interpolation*
    calculate A *according to Eq.(3,4,5)*
    calculate B *and* N *according to Eq.(1)*
    assign $F'(PP) = |V|, F''(PP) = |A|$
    calculate $F(P)$ *according to Eq.(6)*
    Volume$[k * ScaleZ + kk][j * ScaleY + jj]$
    $[i * ScaleX + ii] = F(P)$
END
normalize(Volume)

The reconstruction of the velocity across all the cubic elements can use 3D linear interpolation or any other accurate interpolation scheme that conforms to CFD. For the purpose of reconstruction from Eq. (6), we define the previous point PP as:

$$PP(i, j, k) = \begin{cases} F(PP) = 0 & (i = 0, j = 0, k = 0) \\ (0, 0, k - 1) & (i = 0, j = 0, k \neq 0) \\ (0, j - 1, k) & (i = 0, j \neq 0) \\ (i - 1, j, k) & \text{otherwise} \end{cases} \tag{7}$$

Recall that F is the stream function around point P. It is important to normalize $F(P)$, thus ensuring that the principal stream function lies in the range $[0.0, 1.0]$, for the sake of volume rendering. On the other side, normalization is necessary for cancelling the effect of the selection of reference point and its initial value.

Note that the binormal B to the osculating plane can be approximated by finding the normal to the plane defined by P, P_+, and P_- . Equivalently,

$$B = V_+ \times V_- \tag{8}$$

4 Experiments and Discussion

The goal of our experiments is twofold, to demonstrate the principal stream surface and the principal stream function, and to develop a direct volume rendering method to display flow topology rather than using tracking.

We implemented the principal stream function construction algorithm described above, as well as a ray tracing volume renderer, and performed two experiments with steady irrotational flow: past a circular cylinder, and between two vortices. Volume rendering is done similarly to [10].

4.1 Stream past a Circular Cylinder

We set the center of the circular cylinder to be the line $(5, 5, z)$, with radius $a = 2$, and define r as the distance between (x, y, z) and $(5, 5, z)$, θ is the angle between r and $Xaxis$. Then the velocity data is generated by Eq. 9 at the grid points (see Figure 2).

$$\begin{aligned} V_x &= U_0 \left(\frac{a^2}{r^2} \cos 2\theta - 1 \right) \\ V_y &= U_0 \frac{a^2}{r^2} \sin 2\theta \\ V_z &= 0.0 \end{aligned} \tag{9}$$

Figure 5 shows the results of two volume renderings. The stream is flowing from left to right in both cases. Figure 5a shows the distribution of the principal stream function; color is mapped as a rainbow from $[0..255]$. Different principal stream surfaces have different colors, due to the different principal stream function values. Noted that this method is fully automatic, as opposed to the implicit stream surface method in [14], in which the inlet boundary should first be detected then different source functions are assigned to all different inlets. Figure 5b shows the result of iso-surface rendering, with several surfaces generated from corresponding iso-values. The colors of these values are mapped to their stream surfaces in Figure 5c.

This simple experiment demonstrates the correctness of the principal stream surface and the construction of principal stream function. It also shows the effectiveness of using principal stream surface to display the flow topology.

4.2 Stream between two Vortices

The velocity of this fluid field are calculated according to the following equations:

$$\begin{aligned} V_x &= \frac{1}{2}[\textit{Vortex}_x(x, y, z, -5.5, -5.5, -5.5) + \\ &\qquad \textit{Vortex}_x(x, y, z, 15.0, 15.0, 15.0)] \\ V_y &= \frac{1}{2}[\textit{Vortex}_y(x, y, z, -5.5, -5.5, -5.5) + \\ &\qquad \textit{Vortex}_y(x, y, z, 15.0, 15.0, 15.0)] \\ V_z &= \frac{1}{2}[\textit{Vortex}_z(x, y, z, -5.5, -5.5, -5.5) + \\ &\qquad \textit{Vortex}_z(x, y, z, 15.0, 15.0, 15.0)] \end{aligned}$$

$$\begin{aligned} \textit{Vortex}_x(x, y, z, x_0, y_0, z_0) &= sym(z - z_0) \cdot \\ &\left(-\frac{A}{2\pi} \frac{z - z_0}{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \cdot \right. \\ &\qquad \left. \frac{x - x_0}{\sqrt{(x - x_0)^2 + (y - y_0)^2}} \right) \end{aligned}$$

$$\begin{aligned} \text{where } \textit{Vortex}_y(x, y, z, x_0, y_0, z_0) &= sym(z - z_0) \cdot \\ &\left(-\frac{A}{2\pi} \frac{z - z_0}{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \cdot \right. \\ &\qquad \left. \frac{y - y_0}{\sqrt{(x - x_0)^2 + (y - y_0)^2}} \right) \end{aligned}$$

$$\begin{aligned} \textit{Vortex}_z(x, y, z, x_0, y_0, z_0) &= sym(z - z_0) \cdot \\ &\left(\frac{A}{2\pi} \frac{\sqrt{(x - x_0)^2 + (y - y_0)^2}}{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \right) \end{aligned}$$

$$sym(x) = \begin{cases} 1 & x \le 0 \\ -1 & x > 0 \end{cases} \tag{10}$$

where $A = 720, x = (1..10), y = (1..10), z = (1..10)$

This equation is applied to a $10 \times 10 \times 10$ regular grid. Figure 6 shows the volume-rendered result, using the same rainbow color mapping as in Figure 5. Transparency is set to 0.9 at all points. The value range of the principal stream function is mapped to $[0 \ldots 255]$ for the sake of volume rendering and image processing. It is easy to perform volume cutting to view the inner topology of the flow. Figure 7 shows several different value ranges of principal stream function (generated from their corresponding stream function values). This is useful for visualization; the user may adjust the value range of the principal stream function until satisfied with the topology thus generated. Figure 8 shows the stream surfaces directly modeled by iso-surface modeling, with two iso-values used. After a high-pass filter is applied to the principal stream function data set before volume rendering, implicit stream surfaces are enhanced as shown in Figure 9. The flow topology of the fluid field is displayed more clearly.

4.3 Discussion

A particular principal stream function possesses not only one surface, but a collection of level sets, which may have more than one connected component. Physically, these stream surface components will often meet at some places outside the current domain. In Figure 5, the separation stream surfaces located at the lower part of the box will meet just below the bottom of the box, which we can directly detect from the trend of its neighbors. In Figure 7, since the two vortices belong to two different streams and their inner parts will not meet together, these results are due to the spread of stream function from their intersection region, that is, a saddle point. This phenomenon can be avoided by defining a ranking direction of stream surfaces, if the normal vector of the principal stream surface is opposite to the ranking direction we simply reverse it. Actually, in Figure 5 of the first experiment, the ranking direction is used to attain the different stream surfaces along Y axis. In Figure 7, we notice some isolated stream surfaces in Figure 7(g) and 7(h), which is the result of the saddle point in the middle of the field.

Just as in the construction of a potential field, the position of reference point and its initial value have no effect on the final distribution of the stream function after normalization. Changing the

reference point and its initial value has no effect on the final distribution. Changing the reference point affects only the absolute value of the principal stream function. Normalization requires that we know the minimum and maximum values of the principal stream function, and so normalization is not done until volume rendering begins.

The magnitude of the principal normal vector equals the magnitude of the velocity at any particular point, and so the length of the principal normal vector is proportional to the rate of change of the principal stream function. A region of large velocity values will possess a large range of the stream function. This means that the volume rendering must display more stream surfaces (corresponding to iso-surfaces) and thus more colors in the areas of large velocity, supplying another way to visualize velocity magnitudes from the distribution of principal stream surfaces. This is shown, for example, in the upper-right and lower-left corners of the cube in Figure 7.

Although we can construct iso-surfaces easily, as shown in Figure 8, color mapping is important in viewing the flow topology. Changing the stream function value range will change the mapping density, as specified by the user. In addition, selecting different normalization value ranges also allows the user to view dense distribution in the volume.

5 Conclusions and future work

The generation of principal stream surface is a new and general method for visualization of the topology of irrotational 3D flow. Its advantages include:

- Flow topology is represented by principal stream surfaces.
- Modeling of principal stream surfaces does not require careful placement of starting points.
- The principal stream function is constructed automatically, with no user interaction.
- Volume rendering can be applied directly to the visualization of flow topology.
- Modeling and rendering can be implemented in an efficient and simple fashion.

The following aspects need further work:

- Extending the PSS to numerically simulated data sets, such as bluntfin vector data, to construct the PSS directly from a CFP irregular grid instead of the present regular grid.
- Using a multi-neighbor construction method instead of the present single-neighbor scheme, to increase the stability of construction.
- Determining to what extent high order terms of the Taylor expansion affect the accuracy of the principal stream function, especially near the critical point.
- A more effective volume rendering method to render the topology of stream motions, including segmentation rather than thresholding to extract flow topology from the PSS.
- Currently, the local velocity orientation is not as clear as the result using the linear integral convolution method.

Acknowledgements

We are grateful to Prof. Tim Poston of Institute of Systems Science, National University of Singapore and Mr. Zeba Kimmel for their careful proofreading and helpful suggestions. Thanks also to the reviewers for their valuable suggestions and careful review. This work was supported by the CUHK UGC Research Grant Direct Allocation Program.

References

[1] P. Buning. Sources of Error In The Graphical Analysis Of CFD Results. *Journal of Scientific Computing*, 3(2), 1988.

[2] B. Cabral and L. Leedom. Imaging Vector Fields Using Line Integral Convolution. In Steve Cunningham, editor, *SIGGRAPH 93*, pages 263–270. ACM SIGGRAPH, 1993. ISSN no. 1069-529X.

[3] R. Drebin, L. Carpenter, and P. Hanaran. Volume Rendering. In *Computer Graphics (SIGGRAPH 88 Conference Proceedings)*, pages 65–74. ACM SIGGRAPH, 1988.

[4] P. Eliasson, J. Oppelstrup, and etc. Stream3D:Computer Graphics Programs For Streamline Visualization. *Advanced Engineering Software*, 11(4):162–168, 1989.

[5] T. Fruhauf. Raycasting Vector Fields. In *IEEE Visualization '96*, pages 115–120. IEEE, October 1996.

[6] J. Hemman and L. Hesslink. Visualization Of Vector Field Topology In Fluid Flows. *IEEE Computer Graphics and Application*, 11(3):36–46, 1991.

[7] J. Hultquist. Constructing Stream Surfaces In Steady 3D Vector Fields. In Arie E. Kaufman and Gregory M. Nielson, editors, *IEEE Visualization '92*, pages 171–177. IEEE, October 1992. ISBN 0-8186-2897-9.

[8] D. Kenwright and G. Mallinson. A 3D Streamline Tracking Algorithm Using Dual Stream Function. In Arie E. Kaufman and Gregory M. Nielson, editors, *IEEE Visualization '92*, pages 62–69. IEEE, October 1992. ISBN 0-8186-2897-9.

[9] B. P. Leonard. Stable And Accurate Convective Modeling Procedure Based On A Quadratic Upstream Interpolation. *Computer Methods in Applied Mechanics and Engineering*, 19, 1979.

[10] M. Levoy. Display Of Surface From Volume Data. *IEEE Computer Graphics and Application*, 8(3):29–37, 1988.

[11] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH 87 Conference Proceedings)*, pages 163–169. ACM SIGGRAPH, 1987. ISSN No. 0097-8930.

[12] G. Turk and D. Bank. Imaged-Guided Streamline Placement. In Holly Rushmeier, editor, *SIGGRAPH 96*, pages 453–460. ACM SIGGRAPH, 1996. ISBN 0-201-94800-1.

[13] J. van Wijk. Spot Noise Texture Synthesis For Data Visualization. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH 91 Conference Proceedings)*, pages 309–318. ACM SIGGRAPH, 1991. ISBN 0-201-56291-X.

[14] J. van Wijk. Implicit Stream Surface. In Greory M. Nielson and Dan Bergeron, editors, *IEEE Visualization '93*, pages 245–252. IEEE, October 1993. ISBN 0-8186-3940-7.

[15] J. van Wijk, A. Hin, W. de Leeuw, and F. Post. Three Ways To Show 3D Fluid Flow. *IEEE Computer Graphics and Application*, 14(5):33–39, September 1994.

[16] C. S. Yih. *Fluid Mechanism*. New York McGraw-Hill, 1969.

[17] P. Yueng and S. Pope. An Algorithm For Tracking Fluid Particles In Numerical Simulation Of Homogenous Turbulence. *Journal of Computer Physics*, 79, 1988.

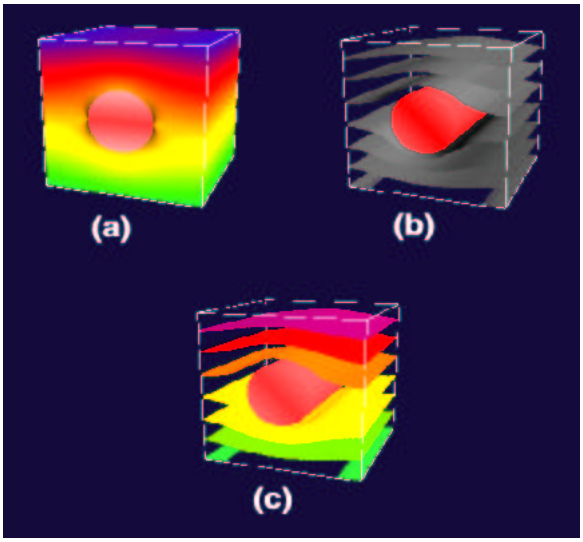

Figure 5: A flow field passing through a circular cylinder. (a) Principal Stream Function (b) Principal Stream Surfaces (c) Color mapped of (b)

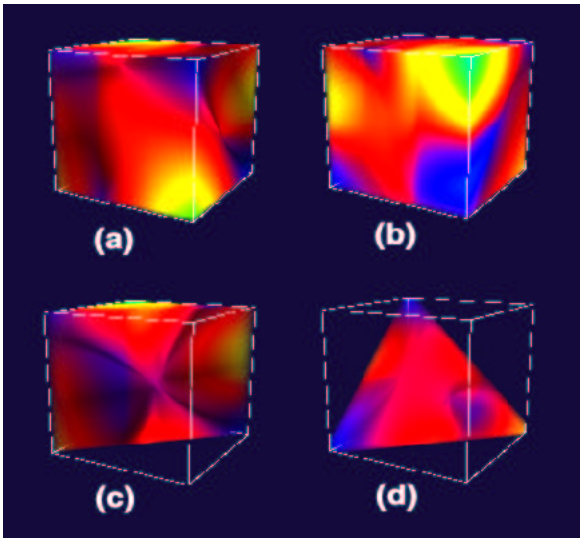

Figure 6: Volume rendering of the Principal Stream Function of Eq.(10) (a) rotate X = 15, Y = 60 (b)rotate 15, Y = -120 (c, d) volume cutting of (a,b) respectively.

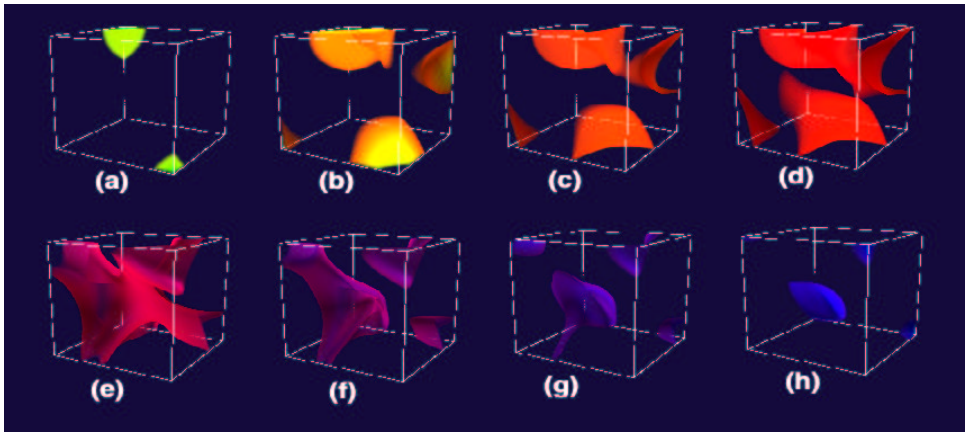

Figure 7: The Principal Stream Function distribution in different value range
(a) [165, 255] (b) [100, 165] (c) [90, 100] (d) [80, 90] (e) [50, 60] (f) [40, 50] (g) [30, 40] (h) [0, 30]

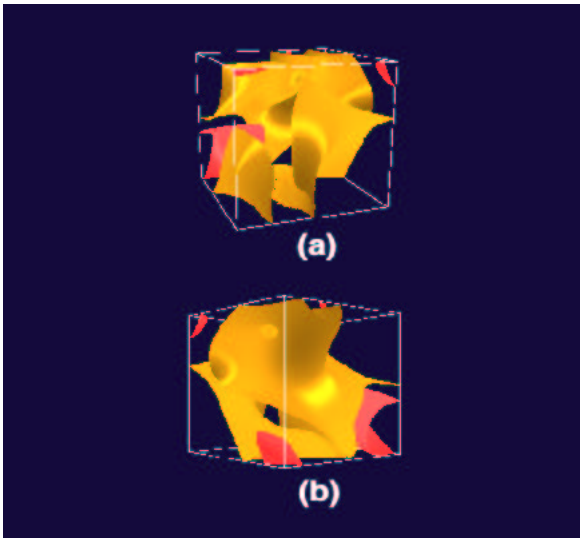

Figure 8: Principal Stream Surfaces, yellow surface iso value = 78, red surface iso value = 95

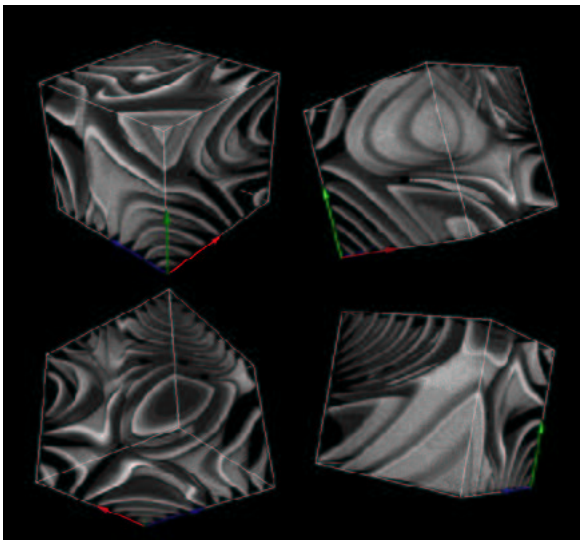

Figure 9: Application of high-pass filter to enhance the principal stream function