Constructing Stream Surfaces in Steady 3D Vector Fields

J.P.M. Hultquist
hultquist@nas.nasa.gov

July 1992
Report RNR-92-025

Numerical Aerodynamic Simulation Systems Division
NASA Ames Research Center, Mail Stop T045-1
Moffett Field, CA 94035-1000

```
@inproceedings{xxx,
 author    = "J.P.M. Hultquist",
 title     = "Constructing Stream Surfaces in Steady
              3D Vector Fields",
 booktitle = "Proceedings of Visualization '92",
 address   = "Boston, MA",
 month     = oct,
 year      = 1992,
 pages     = "171--178",
 note      = "(also RNR Tech Report 92-025)"}
```

Abstract

A *streamline* is a curve which is everywhere tangent to a steady fluid velocity field. A *stream surface* is the locus of an infinite set of such curves, rooted at every point along a continuous originating line segment, or *rake*. A stream surface may be approximated by the triangular tiling of adjacent pairs of integrated streamlines. Such a surface may be refined by repeatedly splitting the widest of the ribbons by the insertion of new curves.

A more efficient method begins with a discretization of the rake. These particles are repeatedly advanced a short distance through the flow field. New polygons are appended to the downstream edge of the surface. The spacing of the particles is adjusted to maintain an adequate sampling across the width of the growing surface. This approach more efficiently accesses the flow field volume data and produces a better distribution of points over the two dimensions of the stream surface.

1 Introduction

Tangent curves and narrow ribbons are often used to depict the velocity field of numerically simulated fluid flows, but the visual interpretation of such images is often difficult. More recent efforts have generalized the technique to construct a two-dimensional tangent surface. These sheets are more visually effective, but their construction is much more time-consuming. The curve and ribbon methods are described here, followed by a description of some earlier methods for stream surface construction. These ideas provide the context for the improved surface construction algorithm described in the remaining sections.

1.1 Curves and Ribbons

Many methods of vector field visualization depend on the calculation of tangent curves through the field. When the field is represented by a grid of sample points, the calculation of these curves is generally performed using numerical integration over some piecewise interpolant. If the field is the velocity of a fluid, each curve defines the path traveled by a massless particle advected through the flow; such curves are called *streamlines*.

Most flow visualization packages allow the user to specify a set of initial points from which a set of these curves is computed. This set of points is often called a *rake*, a name taken from the array of tubing used in wind tunnels to introduce multiple streams of smoke into the flow.

Careful use of depth cues can aid in the interpretation of such images. Intensity cueing, stereopsis, and motion are perhaps the most commonly applied methods. Another aid is the representation of each curve as a narrow ribbon (*e.g.* Belie [1], Kerlick [9]) or a space-filling and shaded cylindrical tube (*e.g.* Dickinson [2], Haimes [5]).

Schroeder and his colleagues [13] describe a model called the *stream polygon*. The deformation of an infinitesimal fluid element is tracked along the length of a streamline. This deformation is depicted graphically by the changing shape of a sequence of polygons placed at regular intervals along the computed curve.

1.2 Stream Surfaces

Just as a streamline is the locus of a single point advected over time, a *stream surface* is the two-dimensional locus of an advected infinitely-malleable initial seed curve. Alternatively, the sheet is the locus of all the streamlines with a seed point on the initial line. The initial curve is a one-dimensional analogue of the initial seed point used in the placement of a single streamline.

A few carefully-placed rakes can create surfaces which clearly convey the shape of a complicated flow field. The shading and obscuration of a surface provides visual cues which help one to interpret two-dimensional pictures of this three-dimensional model. Surfaces curve both across and down their length, and widen in divergent flow. Surfaces can be overlaid with texture to represent additional measures of the flow, can be rendered with variable transparency to mimic the appearance of empirical smoke injections.

Krueger [11] implemented a proof-of-concept system for the visualization of flow fields using a novel user interface called *VideoDesk*. A video camera was used to merge an outline of the user's hands into the three-dimensional space of the data. One of the visualization models supported in this environment was a simple stream surface, constructed with a polygonal tiling of adjacent pairs of streamlines. Ribbons which exceeded a specified width were truncated, with the latter portion represented by two narrower ribbons.

Helman [6, 7] has implemented a system which identifies the topologically significant curves on the solid boundaries of a vehicle. The software then attempts to construct the topological separatrices, that is, the stream surfaces which emanate from these lines and extend into the surrounding flow. Helman used a ribbon tiling approach to construct these surfaces, splitting the widest ribbons down their entire length.

Eder [3] developed a distributed system for computing stream surfaces. After the user specified the rake for a new surface, the system computed between fifty and two hundred curves using a vectorized code running on a Siemens-Nixdorf VP-200. These curves were copied into the memory of a workstation, and the surface so described resampled into a rectangular array of points. A two-pass filtering marked some of these points as extraneous, and the points which remained were tiled with polygons to represent the surface.

1.3 Problems

Typical flow fields diverge greatly and the surfaces embedded in such flows twist and fold with wild abandon. This deformation complicates the polygonal approximation of the surface. Divergence of the flow causes adjacent streamlines to separate, while shearing tends to widen the gap between points on the same time-line. The two-phase approach taken by Eder can therefore leave sparsely sampled regions in the finished polygonal sheet.

The adaptive ribbon-splitting methods of Helman and Kreuger produce a better distribution of sample points over the two dimensions of a diverging surface; however, the complete tracing of individual curves results in scattered access patterns to the memory pages and cache lines of the sampled flow data.

An improved algorithm, described here, advances a a row (or *front*) of particles in a tightly-clustered group. This approach more efficiently accesses the sampled field data and provides better control over the sampling density across the width of the evolving surface representation. An implementation of these ideas was built using the SuperGlue programming environment [8].

Section 2 describes some of the mathematics of the stream surface concept. Section 3 surveys some methods for the polygonal tiling of ribbons, one of which is used in the advancing front algorithm described in section 4. Methods for controlling the density of particles across the advancing front are presented in section 5. This paper concludes with a review of the advantages and the potential pitfalls of this new technique.

2 Mathematics

A stream surface is a two-dimensional parametric surface embedded in the three-dimensional domain of the flow field. Constructing the surface may be viewed as the problem of generating a well-distributed collection of sample points over the two dimensions of that surface, followed by a polygonal tiling of those points.

2.1 Parameterization

An ideal stream surface is the locus of an infinite number of streamlines. Each one may be uniquely identified by the fractional displacement ($s \in [0, 1]$) of its seed point along the originating rake. The surface supports a second set of curves which are the advected images of the rake at an infinite sequence of downstream displacements. These curves, called *time-lines* in fluid flow, may be labelled by a parameter ($t \in [0, N]$). These two sets of curves define a two-dimensional coordinate space over the surface, illustrated in figure 1.

The surface is bounded on one edge by the rake itself, which defines the time-line (t_0). The edges of the surface are marked by streamlines (s_0) and (s_1). These are joined by the sequence of time-lines (t_k), which are increasingly distorted by the flow.

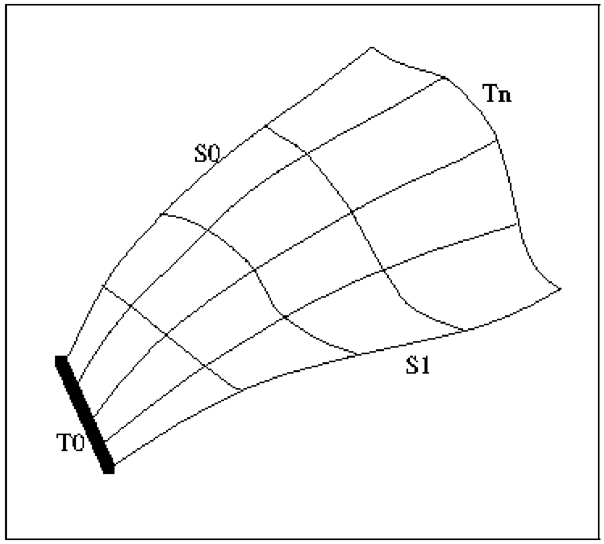

Figure 1: Parametric space over a stream surface.

2.2 Polygonal approximation

This continuous two-dimensional sheet may be numerically approximated and depicted using polygons. The most straightforward implementation creates a rectangular mesh of points at regular intervals in (s, t) space. For each point (s_i, t_j), the first coordinate defines the displacement along the rake of the containing streamline, while the second value defines the accumulated integration stepsize downstream along that curve.

Each streamline (s_i) may be computed by numerical integration using a fixed stepsize to produce a sequence of points (s_i, t_0) through (s_i, t_n). An adaptive stepsizing method could be used, if followed by a resampling of the curve at fixed intervals. Normals may then be computed over this rectangular mesh and the result rendered.

As described above, a simple rectangular mesh cannot adequately represent a highly convoluted surface without a massive number of sample points. Efficient, yet accurate, representation of the surface demands adaptive sampling over its two dimensions.

The sampling density along the length of each streamline may be controlled by adaptively adjusting the integration stepsize so that regions of high curvature are captured. These methods have been carefully analyzed and are trusted by the numerical scientific community.

The time-lines, which span the surface in the crossflow direction, also exhibit curvature; therefore, the approximation of the surface must be augmented by a varying number of samples along that dimension. The splitting of ribbons increases the density of samples across the polygonal surface. The surface construction methods of Helman and Kreuger, and the method presented here, all are based on the splitting of ribbons. The polygonal tiling of such ribbons is discussed in the next section.

3 Tiling of ribbons

A number of different methods can create a ribbon by constructing polygons which span the gap between a pair of curves. All of these methods create triangles which connect the seed points of the two curves and extend downstream. Each new triangle shares a common edge with its predecessor, and consumes a third point from one of the two bordering curves.

The entire ribbon is constructed as the result of a sequence of choices which advance along the two curves, producing a new triangle at each step. Two curves with $(N+1)$ points in each can therefore be tiled with triangles in *(2N choose N)* different ways. Any one of several algorithms may be used to select a "reasonable" tiling from among these possibilities. Regular and globally-minimal tiling, shown in figure 2, have been used in the past. The present algorithm uses a locally-greedy algorithm which is depicted in figure 3.

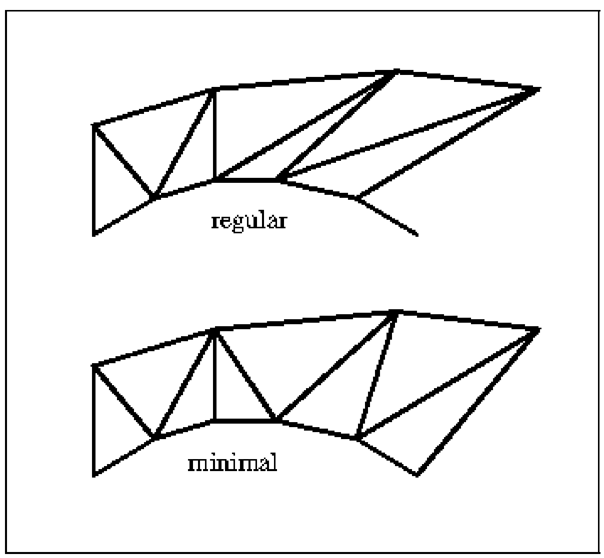

Figure 2: Regular and globally-minimal tiling.

3.1 Regular Tiling

The simplest method alternates sides, thereby consuming point values on the two curves at an equal rate. Krueger [10] used this method in the VideoDesk application. It works well when the flow field is well behaved; in highly sheared flow, lockstep connection of these points produces long skinny triangles. This tiling may also exhaust the points on one curve before reaching the end of the other.

Siclari [14] proposed normalizing the curves by arc length to overcome this difficulty, but this raises similar problems in helical flow. The curve which lies closest to the axis will be straighter than its partner, and triangles will once again span an increasingly greater distance.

3.2 Minimal tiling

The problems caused by shear can be resolved by an irregular tiling of each ribbon. The construction and the display of such a tiling is more difficult, but the shape of these triangles is closer to isosceles.

The polygonal tiling which is globally minimal in some measure can be obtained by the method of Fuchs, Kedem and Uselton[4]. This algorithm uses a dynamic programming approach to find the minimally-measured tiling which connects two closed loops of points. In this application the initial points of the two adjacent curves must be connected to one another, so a simpler version of the algorithm may be used. Helman [6, 7] used this method to construct surfaces of minimum total surface area.

3.3 Greedy tiling

The globally-minimal algorithm requires all of the points on the borders of each ribbon. A greedy tiling method produces similar results with less computational cost, and can begin producing triangles for display before the integration of the curves has been completed.

The tiling begins at the rake, and continues down the length of a ribbon. The two possible triangles at each iteration are compared; the one with the shortest leading edge is appended to the growing ribbon. In practice, when the curves are close together and well sampled, this greedy method produces results quite similar to those of the globally-minimal method.

Figure 3 illustrates the implementation used in the current system, in which two *tracer* structures are used to generate the sequence of points which define each curve. Each tracer maintains the context required to advance a particle through a sampled vector field. Multi-step methods are supported by the storage of a short list of previously-computed points. Integration may be carried out in either the physical or the grid-local computational coordinates. Transition across the boundaries of abutting or overlapping blocks of sample points is

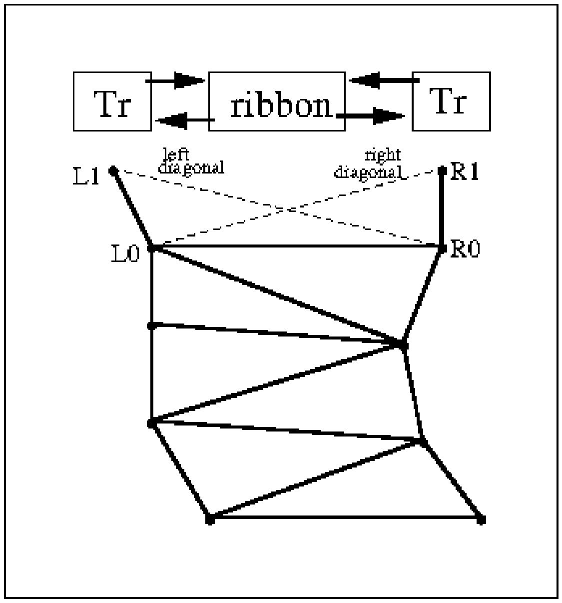

Figure 3: Greedy minimal-width tiling of a ribbon.

always carried out in physical space, since the computational coordinate-spaces of different blocks are incompatible.

A *ribbon* structure is used to connect these points with a sequence of triangles. At each iteration, the two most recently connected points $(L0, R0)$ and their successors $(L1, R1)$ form a quadrilateral which must be split along one of its two diagonals. The shortest of the two diagonals ($(L0, R1)$ or $(L1, R0)$) forms the new leading edge of the ribbon and the base of the next quadrilateral.

The straightforward ribbon-splitting construction algorithm retains an entire curve until the neighboring curves become available; only then may the polygonal tiling commence. The greedy minimal-width algorithm maintains only a few points along each curve, and it outputs point and tiling information in a steady progression from the start of each ribbon to the far edge of the sheet.

The position of each point is recorded in physical space, computational space, and the surface parametric space. Each point record also carries an integer tag which indicates the index of its data written to the output buffers. A triangle is represented by the integer indices for the data of its three vertices. Each vertex is written only once, but its index usually appears in the triple for several triangles.

4 Advancing front

By advancing a set of tracers as a group, we can produce a well-sampled surface while maintaining good locality of access into the flow data. Points are easily inserted into this advancing *front* of points when it is becomes too sparsely sampled, and points can be removed when the sampling becomes needlessly dense.

4.1 Sweeping

The advancing front is represented by a linked-list of alternating *tracer* and *ribbon* structures. Each ribbon holds pointers to its lefthand and righthand tracers. Each tracer, except those which create the streamlines (s_0) and (s_1), produces points which are shared by two ribbons.

The labels "left" and "right" are conventions which define the relative ordering of the tracers across the width of the surface. In fact, the actual implementation performs the sweeping advances in alternating directions across the front.

Each iteration across the front begins with the "leftmost" ribbon, and advances the particles on its two neighboring curves some short distance. The greedy minimal-width tiling algorithm generates each triangle, selecting whichever point produces the triangle with the shortest leading edge.

When the diagonal $(L0, R1)$ is selected, the righthand neighbor of the current ribbon is advanced until the two ribbons are once again the same length along the shared curve. If necessary, the third ribbon is brought abreast with the second, and so on. Each ribbon is advanced along its righthand side as long as the leading edge of each new triangle is shorter than that of it predecessor. This method keeps the front locally orthogonal to streamlines, tends to reduce the overall length of the front, and thereby provides better control over the sampling density.

```
advance_ribbon(self) {
  caught_up = FALSE;
  while (1) {
    L0,L1, R0,R1 ←  next quad
    left_dg   = length(L1,R0);
    right_dg  = length(L0,R1);
    min_dg    = MIN(left_dg, right_dg);
    advance_on_left = (left_dg == min_dg);

    if (caught_up &&
        ((advance_on_left) ||
         (right_dg > prev_dg))) return;

    if (advance_on_left) {
      write_triangle(L0,R0,L1);
      free_point(L0);
      caught_up = TRUE;
    } else {
      write_triangle(L0,R0,R1)
      free_point(R0);
      advance_ribbon(right_neighbor(self));
    }
    prev_dg = min_dg;
  }
}
```

5 Controlling the density

The advancing front of points forms a discretization of a cross-flow line on the ideal surface. Particles may be added to or removed from the front to maintain an acceptable sampling density across this line. Ribbons which encounter massive divergence of the flow may be truncated, thus tearing the surface around this obstacle.

5.1 Adding a particle

A ribbon can be split by the insertion of a new tracer structure and a new ribbon structure into the linked-list which represents the front. One new triangle, shown in figure 4, must be added to the surface to make the transition from one ribbon to its two narrower replacements.

The test which determines when to insert a point must be efficient, since it is evaluated at every iteration of the ribbon advancement. High-order methods which measure the curvature of the front might be valuable in some applications, but simpler methods have proven themselves acceptable for most cases. The current implementation splits a ribbon when the width of the current quadrilateral grows to more than twice its height.

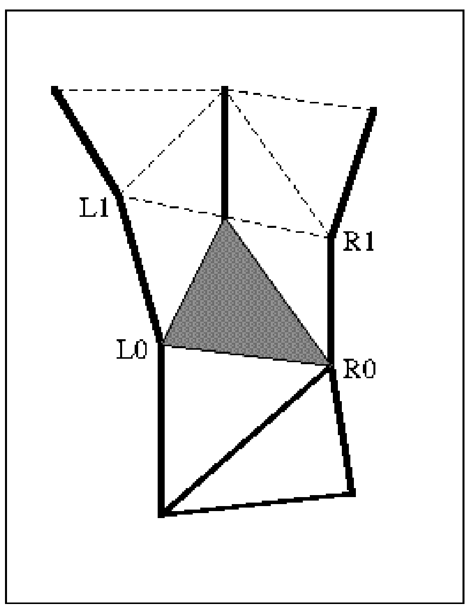

Figure 4: Splitting a ribbon.

When a new particle must be added, it is identified by its location in the parametric coordinate-space of the surface. This new particle must have an

(s, t) location between those of the two particles $(L1, R1)$ which have drifted apart. The three-dimensional physical and computational positions of this new particle must then be determined.

The simplest method merely interpolates in physical coordinates midway between the most recent points on the bordering curves. A more accurate method is to solve the initial value problem starting at (s_{mid}, t_0), but this requires considerable expense. A compromise solution periodically saves an interpolated point in the middle of each ribbon, perhaps every ten iterations. When a ribbon then needs to be split, one of these recent positions can be advanced a short distance to where the ribbon has exceeded acceptable limits.

5.2 Removing a particle

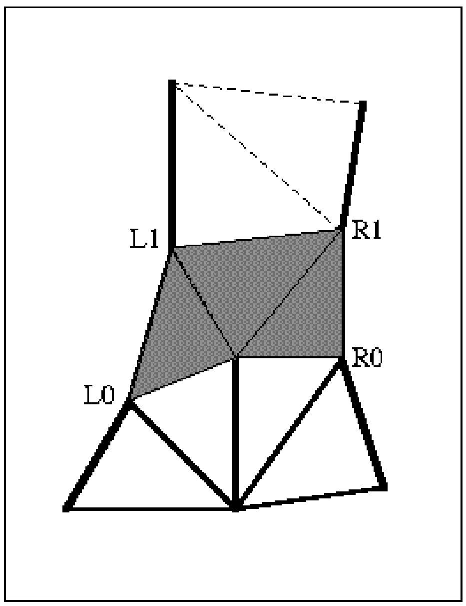

Figure 5: Merging two ribbons.

When a particle is deemed unnecessary, it is removed from the front. The adjoining ribbons are merged into a single, wider replacement. This requires the creation of three new triangles which make the transition between the leading edges of the two ribbons and the single leading edge of the new wider replacement. This transition is shown in figure 5, which also demonstrates that the merging of ribbons advances the new ribbon by one step along both its neighboring curves, and thus requires the subsequent advancement of the righthand neighbor as described above.

The test for when to merge ribbons must consider the abutting quadrilaterals of two neighboring ribbons. In the current implementation, two ribbons are merged if these six points are roughly coplanar and if their height is greater than their combined width. Note that failure to merge two ribbons produces needless additional polygons in the final result, while failure to split a ribbons can produce an inaccurate representation of the true surface.

5.3 Ripping

When a portion of the front is stretched rapidly, it is sometimes preferable to sever the front and to continue the independent advancement of these two portions. The software detects this condition by comparing the relative direction travelled by the pair $(L0, R0)$ of adjacent particles. When these are very close and headed in almost opposite directions, then an obstacle to the flow is assumed to be near. The current ribbon is truncated, and its neighboring curves become bordering edges of two separate fronts.

5.4 Examples

This ripping of the surface is depicted in the color images, which depict a flow field computed by Rogers [12]. The grid defines a single cylindrical coordinate-space with 38 points in the vertical direction, 38 points radially, and 76 points about the circumference. This gives a total of 109,744 grid points, with position and velocity recorded at each point for a memory size of $(N \times (3 + 3) \times 4) =$ 2.6 Mbytes. This is a moderately sized solution, with present-day datasets now ranging in the one to two "Mpoint" range. Each surface was fully computed in about three wall-clock seconds on an SGI VGX-320, and each contains roughly 2500 triangles.

This solution represents an incompressible flow impinging a vertical post mounted on a flat plate. Twin vortices are shed from this obstacle and the entire flow is reflected about a horizontal symmetry plane across the top of the data depicted here. Figure 6 is a side view with the flow passing from left to right; figure 7 shows the same surfaces from a three-quarter angle of view.

Figures 8 and 9 show three-quarter and top views of a stream surface which impinges the base of the post on the upstream face. The flow reverses direction in this region, and the surface tears in the presence of the incoming flow. Note the attenuation of the textured time-lines, and the increasingly acute angles these make with the streamlines.

Figure 10 demonstrates the maintenance of the front orthogonal to the local flow direction. Figure 11 highlights the triangles which were inserted when ribbons were split in this highly divergent region.

6 Conclusions

This algorithm has been implemented and used by the author to illustrate a number of flow field solutions. The code is currently being ruggedized for production use.

6.1 Advantages

This new algorithm provides a method for the interactive exploration of flow field data. It offers increased performance, produces improved sampling densities over the constructed surface, and allows the rapid display of interim results.

performance The ribbon-based methods are most easily implemented by calculating the full length of each new curve. This causes repeated access to the same vector samples, and this can increase the number of page and cache faults. The advancing front method uses spatial coherence to improve the memory access patterns.

sampling quality The advancing front method begins at the rake and works downstream across the width of the surface. The sampling density across this line can be continually adjusted to meet the demands of the local curvature of the field.

interaction The ribbon splitting methods repeatedly replace a single previously-computed ribbon with two narrower ribbons. If the user interface is designed to continually update an image of the refining surface, then that image must be entirely redrawn. Furthermore, the surface normals are thrown away and new normals must be computed. The advancing front method produces no interim results which would require a full redisplay. Newly-computed triangles are simply drawn into the Z-buffer and color-planes of an evolving image.

6.2 Pitfalls

These surfaces can be a useful tool for the investigation of flow field data. Like any other visualization tool, they can be dangerous to the unwary and misused by the unscrupulous.

accuracy of curves The representation of the surface cannot be any more accurate than the curves on which it is constructed. These curves are numerically integrated though a field which is defined by a piecewise interpolation over a set of numerically calculated sample values. Many opportunities arise for the introduction of error, and the gee-whiz factor of nicely rendered surfaces increases the risk that these considerations will be forgotten, ignored, or swept under the rug.

insufficient splitting If the test for splitting is insufficiently rigorous, then a lengthwise fold in the surface can develop between two adjacent points on the front. This crease can continue to grow in height, and will not be represented in the final result. Unfortunately, such folds are quite common in fluid velocity fields. This problem can be avoided by maintaining a sampling density on the front which is comparable to the local sampling density of the original 3D flow field data.

overeager merging If ribbons are merged excessively, the true shape of the ideal surface will be coarsely described by distracting facets. These artifacts may obscure important information in the underlying data.

6.3 Summary

Maintenance of a front of particles is an efficient method of generating a set of sample points over a two-dimensional stream surface. The resulting triangles are approximately equilateral and of a size adapted to the local curvature of the surface. These surfaces have been used to depict structures in complicated flow field data.

Acknowledgements

I wish to thank Tom Lasinski for his patient support of this work. I have benefited from the advice of Eric Raible, Fred Brooks, Jim Coggins, and Henry Fuchs. Thanks also go to Stuart Rogers for the use of his flow data. Dan Asimov, Al Globus, Kris Miceli, and Mary Hultquist found numerous gaps in logic and spelling.

References

[1] R.G. Belie. Some advances in digital flow visualization. In *AIAA Aerospace Sciences Meeting*, Reno, NV, January 1987. AIAA Paper 87-1179.

[2] R.R. Dickinson. A unified approach to the design of visualization software for the analysis of field problems. In *Three-dimensional Visualization and Display Technologies*, volume 1083, pages 173–180. SPIE, 1989.

[3] Erich Eder. Visualisierung von Teilchenstroemungun mit Hilfe eines Vektorrechners. Master's thesis, Fachberiech Informatik, Fachhochschule Muenchen, Munich, February 1991. (Visualization of Flow Fields Using Vector Computers (*diplomarbeit*)).

[4] Henry Fuchs, Zvi M. Kedem, and Samuel P. Uselton. Optimal surface reconstructions from planar contours. *Communications of the ACM*, 20(10):693–702, October 1977.

[5] Robert Haimes and Dave Darmofal. Visualization in computational fluid dynamics: A case study. In *Proceedings of Visualization '91*, pages 392–397, San Diego, CA, October 1991.

[6] James L. Helman and Lambertus Hesselink. Representation and display of vector field topology in fluid flow data sets. *IEEE Computer*, pages 27–36, August 1989.

[7] James L. Helman and Lambertus Hesselink. Surface representations of two- and three-dimensional fluid flow topology. In *Proceedings of Visualization '91*, pages 6–13, San Diego, CA, October 1991.

[8] J.P.M. Hultquist and Eric Raible. Superglue: A programming environment for scientific visualization. In *Proceedings of Visualization '92*, pages 243–250, Boston, MA, October 1992.

[9] G. David Kerlick. Moving iconic objects in scientific visualization. In *Proceedings of Visualization '91*, pages 124–129, San Francisco, CA, October 1990.

[10] Myron W. Krueger. Private communication, February 26, 1992.

[11] Myron W. Krueger. *Artificial Reality*. Addison-Wesley, second edition, 1991. pages 175-176.

[12] S. Rogers, D. Kwak, and U. Kaul. A numerical study of three-dimensional incompressible flow around multiple posts. In *AIAA Aerospace Sciences Conference*, Reno, NV, January 1986. AIAA Paper 86-0353.

[13] W.J. Schroeder, C.R. Volpe, and W.E. Lorenson. The stream polygon: A technique for 3d vector field visualization. In *Proceedings of Visualization '91*, pages 126–131, San Diego, CA, October 1991.

[14] M. Siclari. Private communication regarding the picture on the cover of *Science*, 245(28), July 1989.

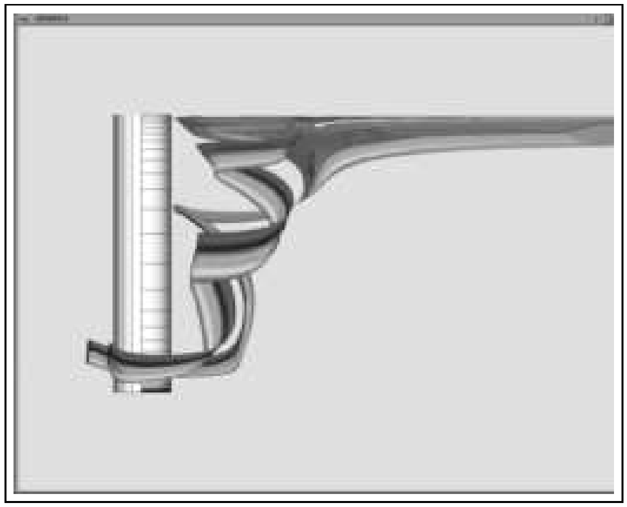

Figure 6: Two stream surfaces shed from a vertical post in an incompressible flow field.

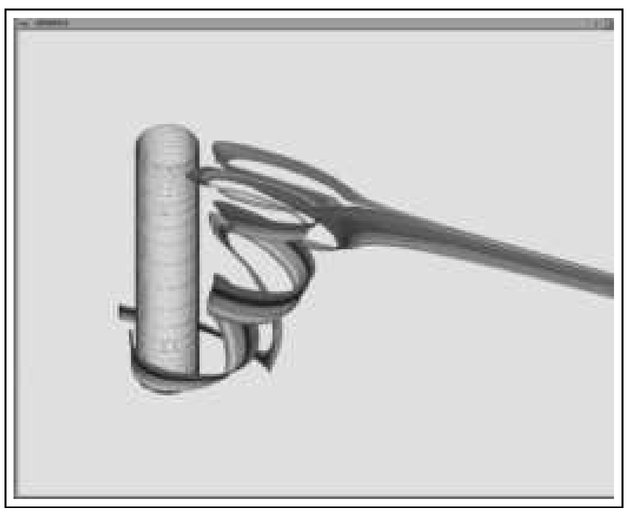

Figure 7: Three-quarter view of two stream surfaces, from above and leeward.

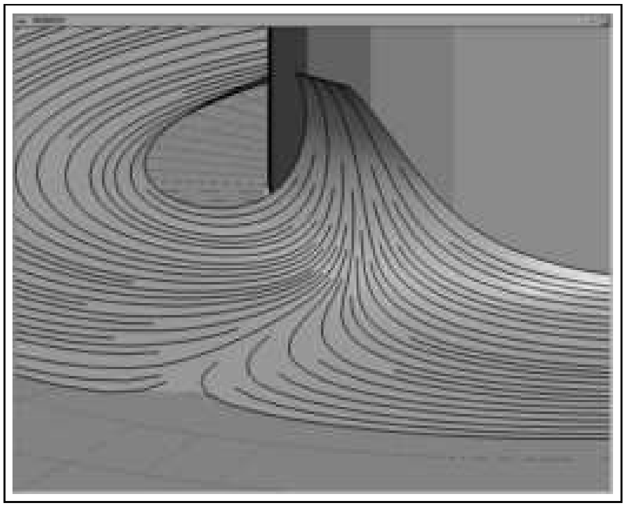

Figure 8: Stream surface in the flow-reversal region near the base of the post on the windward side.

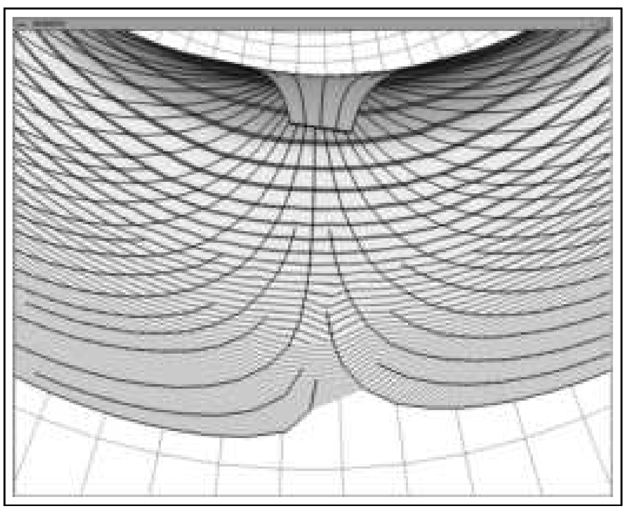

Figure 9: The same stream surface viewed from above and textured with time-lines.

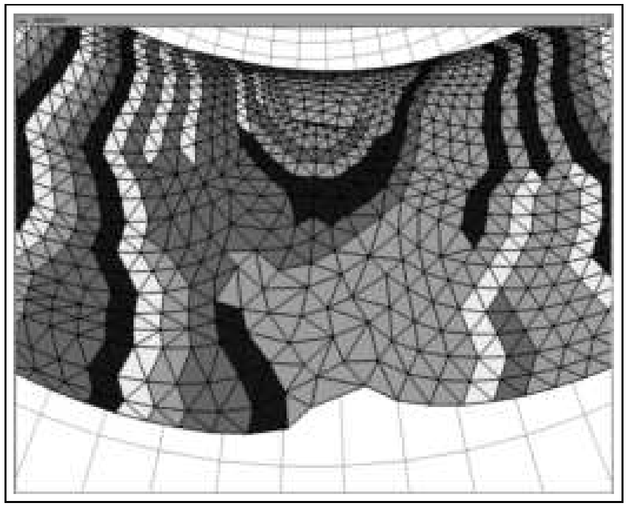

Figure 10: Color-coded representation of the separate sweeps across the expanding sheet.

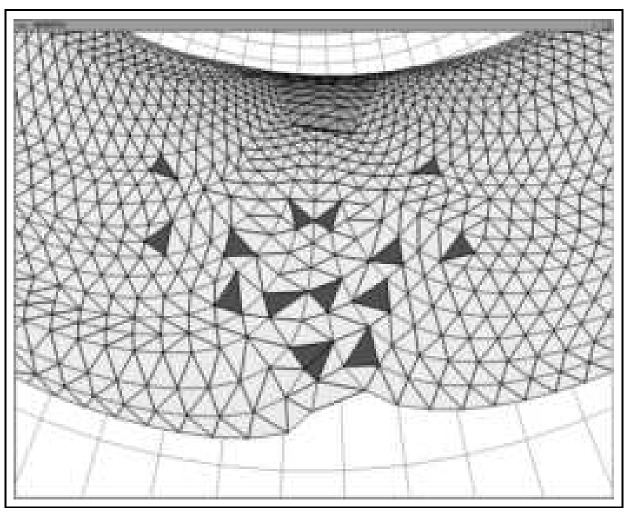

Figure 11: The expanding surface with the splitting of ribbons marked by dark triangles.