FAST METHODS FOR COMPUTING ISOSURFACE TOPOLOGY WITH BETTI NUMBERS

Shirley F. Konkle
University of California, Davis
konkle@cs.ucdavis.edu

Patrick J. Moran
NASA Ames Research Center
pmoran@nas.nasa.gov

Bernd Hamann
University of California, Davis
hamann@cs.ucdavis.edu

Kenneth I. Joy
University of California, Davis
joy@cs.ucdavis.edu

Keywords: Betti Numbers, Isosurface topology.

Abstract Betti numbers can be used as a means for feature detection to aid in the exploration of complex large-scale data sets. We present a fast algorithm for the calculation of Betti numbers for triangulated isosurfaces, along with examples of their use. Once an isosurface is extracted from a data set, calculating Betti numbers only requires time and space proportional to the isosurfaces, not the data set. Because the overhead of obtaining Betti numbers is small, our algorithm can be used with large data.

1. INTRODUCTION

Topology can be used as a means of automated feature detection. One example is determining whether molecules have bonded, which is done by calculating the isosurface of the electron density for the outer valence. When the two surfaces of the molecules become one, then the molecules have bonded. Betti numbers can be used in a different manner. If the desired topology is known in advance and the value to define the isosurface is not, then a range of values that define isosurfaces with the correct topology can be computed as a "batch job." This approach is more convenient than having a user manually guess which isovalues are correct.

Rather than using an ad hoc method, we build on results from algebraic topology. More specifically, we use the Betti numbers of homology groups to classify the topology of isosurfaces. Betti numbers have intuitive physical interpretations: Betti zero, β_0, is the number of connected components, Betti one, β_1, is the number of independent tunnels, and Betti two, β_2 is the number of closed regions in space. Betti numbers provide an intuitive understanding of the topology of isosurfaces.

As part of their work on the topology of alpha shapes [2], Delfinado and Edelsbrunner give a proof for incrementally computing the Betti numbers β_0, β_1, and β_{d-1} for a simplicial complex embedded in $\mathcal{S}_d$. They use the results of the proof to calculate β_0, β_1, and β_2 for tetrahedral meshes in $O(n\alpha(n))$ time and $O(n)$ storage, where n is the number of elements in the simplicial complex. One of the open questions in their work was calculating β_i, $i <= d$ for the general case.

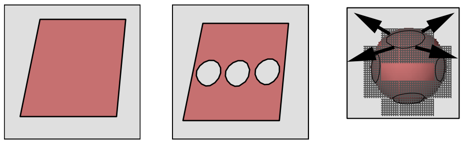
In our work, we restrict ourselves to a study of the topology of isosurfaces. This is more efficient than analyzing the topology of volumes, because we need not store the tetrahedra that represent a volume, but only the triangles defining the isosurface. The time and space bounds are $O(n\alpha(n))$ and $O(n)$, where n is the number of triangles on the surface, a small number compared to the number of tetrahedra and triangles that define a volumetric mesh. One of the main contributions of this paper is calculating β_2 without using tetrahedra.

We also extend our work to hexahedral meshes. In this case, we can directly calculate Betti numbers without splitting the mesh into many tetrahedra.

Section 2 reviews Betti numbers and topology. It covers the intuition for Betti numbers, and introduction to the terminology, our solution, and its mathematical correctness. Section 3 discusses the methods of calculating Betti numbers for isosurfaces. It describes the implementation and supporting data structures, as well as the methods used to extend the algorithm to a hexahedral mesh. Examples are provided in Section 4.

2. BETTI NUMBERS

There are simple physical interpretations of Betti numbers β_0 to β_2. Loosely defined, β_0 is the number of connected components, β_1 is the number of independent tunnels, and β_2 is the number of closed regions defined by a two-manifold surface in 3-dimensional space.



(a) (b) (c)

Figure 1 Betti 1 is the number of holes in a sheet; (a) $\beta_1 = 0$, (b) $\beta_1 = 3$, (c) $\beta_1 = 3$.

Figure 1 shows several examples of β_1. In Figure 1a a flat sheet is shown that has no holes in it, and therefore no tunnels. Figure 1b shows a three-holed example, where the number of tunnels is also three. Figure 1c illustrates a surface also having three tunnels. To determine how many tunnels there exist, the surface must be "stretched flat." To stretch the geometry of Figure 1c flat, one uses the points on the surface where the arrows touch and stretches them in the direction of the arrow. The resulting geometry looks like the one shown in Figure 1b. If a surface can be morphed to look like a flat sheet with n holes, then β_1 will be equal to n.

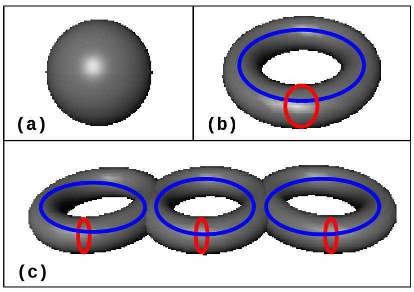


Figure 2 For orientable, closed surfaces, β_1 is two times the number of handles in a sphere: (a) $\beta_1 = 0$, (b) $\beta_1 = 2$, (c) $\beta_1 = 6$.

Some surfaces are closed and cannot be morphed into a flat sheet, as is shown in Figure 2. These surfaces have special cycles shown in red and blue, which show the properties of Betti numbers. Cutting along the red or blue cycles will not divide the torus into two pieces, and neither cycle can be shrunk to a single point. Also, the red and blue cycles cannot be morphed into each other, and no cycles are a combination of other cycles. Closed, orientable surfaces have another nice property. They are all homologous to a sphere with h handles. The analogy is the isosurface of a coffee mug, which has one handle. Thus, β_1 is always 2*h. This is of importance for isosurfaces that do not hit the boundary of the volumetric mesh, because they are closed and homeomorphic to a sphere with h handles.

The Betti number β_2 is more meaningful than β_1, it is equal to the number of closed regions in space. A sphere has β_2 equal to one, two spheres have β_2 equal to two, etc. For general surfaces, two spheres can meet at one point. In that case, there is one connected component but two closed regions in space, *i.e.*, β_0 is one and β_2 is two. This does not occur when isosurfaces are extracted from a tetrahedral mesh or for isosurfaces computed with the "ambiguity decider" of Nielson and Hamann [6].

2.1. BETTI NUMBERS AND THEIR CALCULATION

A *k-simplex* σ is the convex hull of a set T of $k+1$ affinely independent points. A *vertex* is a 0-simplex, an *edge* is a 1-simplex, a *triangle* is a 2-simplex, and a *tetrahedron* is a 3-simplex. A proper *face* σ' of σ is any simplex σ' consisting of a non-empty set U of points, where $U \subset T$. A simplex $\sigma = [u_0, u_1, \ldots, u_k]$ can be oriented by defining an order for its vertices.

A *simplicial complex* $\mathcal{K}$ is a collection of simplices such that these two conditions hold: (1) if $\sigma \in \mathcal{K}$, then any face of σ is also contained in $\mathcal{K}$, and (2) two simplices contained in $\mathcal{K}$ are either disjoint or intersect in a face of both. The k^{th} Betti number β_k of a simplicial complex $\mathcal{K}$ is the rank of the k^{th} homology group of $\mathcal{K}$, see [7].

The *boundary* of a simplex σ is defined as

$$\partial_k(\sigma) = \sum_{i=0}^{k} (-1)^i [u_0, u_1, \ldots, \hat{u}_i, \ldots, u_k],$$

where $u_0, u_1, \ldots, u_k$ are the points defining the simplex. For example, the boundary of a triangle is the set of its edges, and the boundary of a tetrahedron is the set of its bounding triangular faces. The boundary of a complex $\mathcal{K}$ uses the *boundary homomorphism*

$$\partial_k \left(\sum_j a_j \sigma_j \right) = \sum_j a_j \partial_k \sigma_j,$$

where a_j are integers and σ_j are the simplices in $\mathcal{K}$. For an oriented planar mesh of triangles, the boundary is the set of outer edges, since the inner edges cancel. In other words, each “double” inner edge has a positive weight for one direction and a negative weight for the other one. The boundary of a closed manifold triangular mesh has a boundary of zero, since all edges cancel.

Betti numbers have mathematical meaning and properties. Betti number β_0 is the number of connected components of a simplicial complex $\mathcal{K}$. Betti number β_1 is the number of independent tunnels in $\mathcal{K}$. For example, a torus has two tunnels: one that goes through the center and one that goes around the center. A nice property of all closed, orientable surfaces is that they are all homotopic to a sphere with h handles, and β_1 is always $2 * h$, see Figure 3.

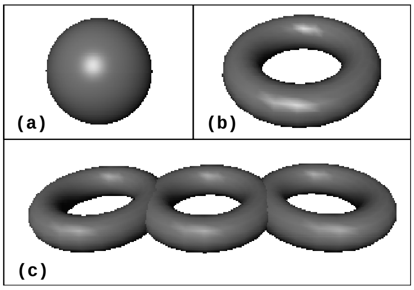


Figure 3 Orientable, closed surfaces: β_1 is two times the number of handles; (a) $\beta_1 = 0$, (b) $\beta_1 = 2$, (c) $\beta_1 = 6$.

Betti number β_2 is the number of closed regions in a simplicial complex $\mathcal{K}$. A regular triangulation of a sphere has $\beta_2 = 1$. Triangulations of two disjoint spheres have Betti numbers $\beta_2 = 2$. If two triangulated spheres intersect at a point, then there is one connected component, i.e., $\beta_0 = 1$, but two closed regions exist, which implies that $\beta_2 = 2$.

3. COMPUTING β_0, β_1, AND β_2

A triangulated isosurface is a simplicial complex $\mathcal{K}$ of dimension two. In the following, we provide algorithms to effectively calculate the Betti numbers β_0, β_1, and β_2. We use the *Euler number* (or *Euler-Poincaré* *characteristic*) χ, which is defined as

$$\chi = \beta_0 - \beta_1 + \beta_2$$

or, equivalently, as

$$\chi = v - e + t,$$

where v, e, and t are the number of vertices, edges, and triangles, respectively, in $\mathcal{K}$.

3.1. COMPUTING β_0

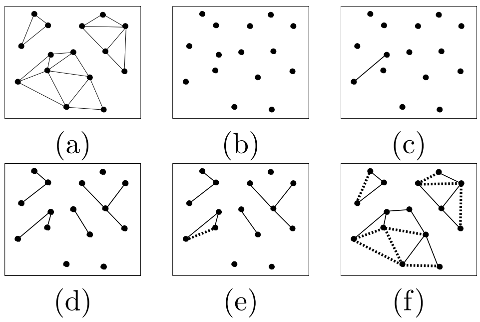


Figure 4 Detecting 1-cycles with incremental method: (a) original triangulation; (b) $\beta_0 = v$, (c) $\beta_0 = v - 1$, (d) $\beta_0 = v - 8$, (e) $\beta_0 = v - 8$, (f) $\beta_0 = v - e' = 3$.

The value of β_0 is calculated using Delfinado and Edelsbrunner's incremental approach [2]. Given an initial triangulation, see Figure 4a, we consider vertices only and ignore all edges and triangles, see Figure 4b. Betti number β_0 is initially set to be identical to the number of vertices. Adding the first edge connects two components, see Figure 4c, and β_0 is decremented by one. More edges that connect components are shown in Figure 4d. Other edges, shown as dotted lines in Figures 4d and 4e, do not connect two components since the components are already connected; instead, they form a cycle. The final number of components, shown in Figure 4f, is the number of vertices v minus the number of edges e' that do not form cycles. In our example, the number of components is 15-12=3.

The implementation is based on the union-find data structure, see Cormen et al. [1]. The initial set of vertices is used to initialize the data structure. Adding edges corresponds to *Union* and *Find* operations. First, a *Find* operation is performed to see whether an edge's two endpoints are already in the same set. If they are not in the same set, the *Union* operation is applied to the endpoints, and e' is incremented by one. If the endpoints are in the same set, no further action is needed. The total run times for the *Union* and *Find* operations is $O(n\alpha(n))$, where α is the inverse of the Ackermann function.

The most difficult part of the algorithm is the construction of exactly one copy of each vertex and edge in the isosurface triangulation to define the input for the Union-Find data structure.

Many isosurface rendering algorithms generate a set of triangles where certain vertices and edges are represented multiple times. One method of avoiding this problem is to compute and represent an isosurface directly with a topological data structure such as the specific quad-edge data structure, using, for example, the quad-edge data structure described by Guibas and Stolfi [3]. However, we hash the vertices and edges, which is straightforward operation in the *NASA FEL Library* [5].

Since each isosurface vertex is obtained from linearly interpolating function values along mesh edges, an isosurface vertex can be hashed on the associated mesh vertex, which has a unique tag in the FEL Library. Topologically, it does not matter where an isosurface point is located on an edge; it only matters that a point exists.

Assuming that an isosurface is generated using a tetrahedral-based isosurface algorithm [8], edges in the isosurface triangulation are only slightly more difficult. Most edges lie on the faces of tetrahedra, see Figure 5a and 5c. These edges use the FEL tag for a triangle as a hash value. Edges lying in the interior of tetrahedra, see Figure 5b, do not need to be stored in the hash table. The reason is this: when a Union operation is first applied to the four edges of the isosurface that lie on faces of the tetrahedra, the interior edge can never connect two components. Therefore, the interior edge does not need to be stored.

3.2. COMPUTING β_2

Betti number β_2 is the number of closed regions in space, and it can be calculated by checking whether all components are closed. The easiest way to determine whether a component is closed is to perform a depth-first search from each triangle to each of its neighboring triangles. If one of the triangle's edges does not have a neighbor, then the isosurface touches the boundary of the underlying volumetric mesh, and the surface is not closed. The total run time is $O(e)$.

There is a mathematical basis for the correctness of the algorithm. For a complex of triangles, β_2 is the number of 2-cycles, or components, whose boundary is zero. Since all isosurface triangulations are orientable–the normal always facing "outward"– a boundary can be computed as a depth-first search is performed. The first time an edge is encountered in the search it will have either a positive or negative weight and is incrementally added to the value of the boundary. When the search reaches that triangle's neighbor, that triangle will use the opposite weight. As a result, the weights will always cancel, unless the surface intersects the boundary of the volumetric mesh. Any surface that stays within the data set has a boundary of zero and is a 2-cycle.

For the algorithm to lead to correct results, an edge cannot be part of more than two triangles. To understand the algorithm's correctness, it is necessary to see how edges are generated and degenerate cases are handled. If we assume that the desired isosurface has isovalue zero, then Figure 5 shows that there are only two symmetric cases for a surface passing through a tetrahedron and two types of edges in those surfaces. The edge type lies completely inside the tetrahedron and is obviously part of only one surface. The second type lies on a triangle face of the tetrahedron. Only one isosurface triangle touches it in the tetrahedron. By symmetry, the neighboring tetrahedron will also have one isosurface triangle at the edge. As a result, there can never be more than two triangles meeting at any edge in an isosurface generated by a tetrahedral mesh.

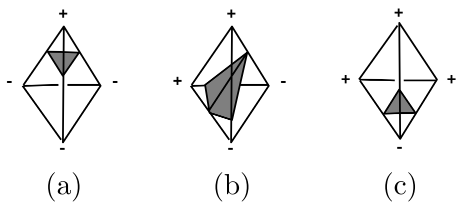


Figure 5 Two cases to be considered for an isosurface passing through a tetrahedron: (a) and (c) are symmetrical cases of a one-triangle configuration, and (b) shows a two-triangle configuration.

The algorithm requires time $O(n)$ time and space, where n is the number of triangles in the isosurface triangulation.

3.3. CALCULATING β_1

The value of β_1 is calculated from the Euler number χ, and the Betti numbers β_0, and β_2. Knowing the values of β_0, β_2, and χ, β_1 is given by

$$\beta_1 = \beta_0 + \beta_2 - \chi.$$

3.4. EXTENSION TO HEXAHEDRAL MESHES

Isosurfaces extracted from hexahedral meshes are more complex than isosurfaces extracted from tetrahedral meshes [4]. Difficulties arise from ambiguous faces and multiple isosurfaces components in a single hexahedral cell. Ambiguous faces, shown in Figure 6, have to be resolved consistently to preserve topology: if on one cell face the "+"'s are separated and the neighboring cell sharing this face separates the "-"'s, then there would be a crack in the isosurface triangulation. Our implementation uses the ambiguity decider from Nielson and Hamann to produce crack-free triangulations [6]. Even though the connectivity on faces should be uniquely decided with this method, floating point errors can potentially cause different results when the same shared face is calculated from the two hexahedral cells that share it. A simple solution to prevent cracking is to hash each ambiguous face first, apply the ambiguity decider once, and reference the result from the hash table.

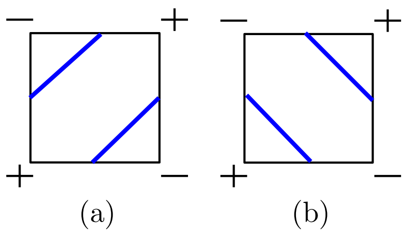


Figure 6 Example of an ambiguous face: (a) positive points are separated, (b) negative points are separated.

One also has to deal with multiple isosurfaces components in a single hexahedral cell. The depth-first search for calculating β_2 is more complicated, because there are two isosurfaces that pass through an ambiguous face. When calculating β_2, extra bookkeeping must be done to ensure that the depth-first search chooses the correct edge.

4. RESULTS

We have found that one can determine relevant isosurfaces of interest by examining β_0, β_1, and β_2 for various isosurface threshold values.

Figure 7 shows isosurfaces of the Laplacian of the electron density field of adenine. Here, $\beta_1 = 20$, which indicates the existence of more than the visible ten components. Clipping these isosurfaces reveals the additional components inside the visible parts. Clearly, the Betti number reveals additional information about the isosurface.

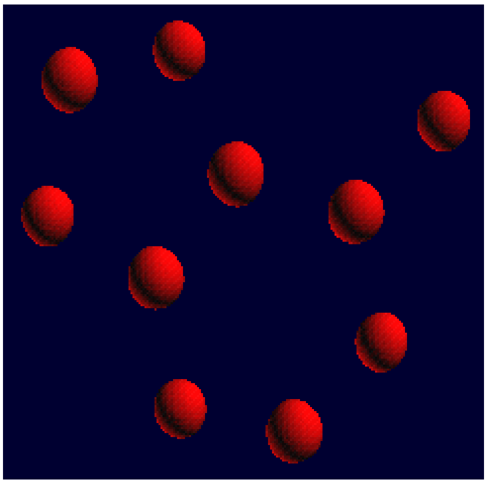


Figure 7 Isosurfaces of the Laplacian of the electron density field of adenine. Here, $\beta_0 = 20$, which is larger than the number of visible objects.

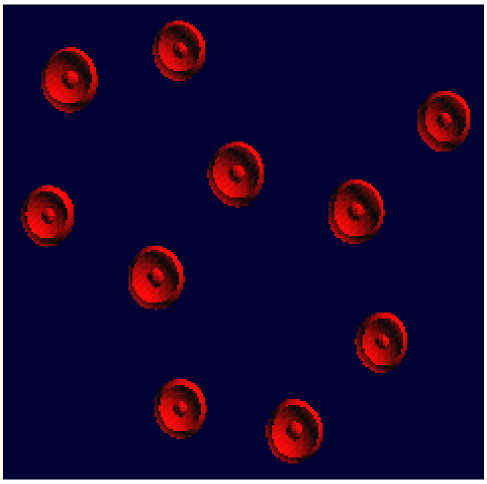


Figure 8 Isosurfaces of the Laplacian of the electron density field of adenine. Clipping the isosurface components visible in Figure 7 reveals the additional, (previously invisible) components.

Figures 9–11 show isosurfaces of the Laplacian of water. Figure 9 shows the isosurface at isovalue -0.931, and the associated Betti numbers are $\beta_0 = 3$, $\beta_1 = 1$, and $\beta_2 = 2$. This implies that the isosurface has three connected components, one tunnel, and two closed regions. Figure

10 shows the isosurface for isovalue -0.935, and the Betti numbers are $\beta_0 = 5$, $\beta_1 = 0$, and $\beta_2 = 3$. They indicate that this isosurface has two additional components and one additional closed region. In this case, one original component has "split" into three components. Figure 11 shows the components in the interior of the isosurfaces. (This image was generated by clipping the isosurface shown in Figure 10.)

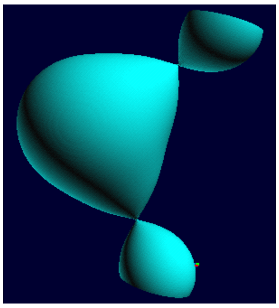


Figure 9 Isosurfaces of the Laplacian of water. For the isovalue of -0.931, three components exist.

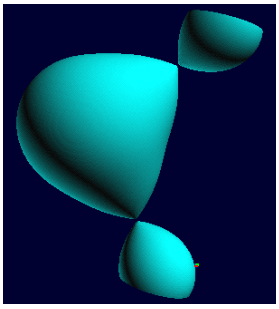


Figure 10 Isosurfaces of the Laplacian of water. For the isovalue of -0.935, five components exist.

One of the features of the Blunt Fin data set, shown in Figures 12 and 13, is the presence of vortices near the base of the fin. These vortices are detected by sweeping a range of isovalues, see Table 1, and observing that β_2, the number of tunnels, is equal to two for a wide range of isovalues. Since tunnels in the isosurface of the velocity magnitude are a possible indicator for vortices in the data set, a range of interesting isovalues is identified. By visually inspecting some of the resulting isosurfaces for tunnels, the region in space for the potential vortices is identified, see

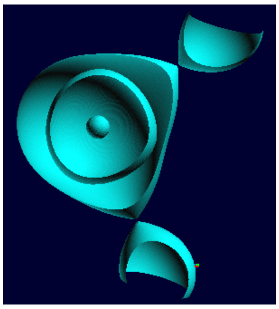


Figure 11 Isosurfaces of the Laplacian of water. Clipping the isosurface shown in Figure 10 reveals full structure of the isosurface.

Figure 12. Figure 13 shows where one tunnel split off into a separate component.

5. CONCLUSIONS

There are a number of applications where Betti numbers provide meaningful information about complex, large-scale time-varying data sets. We have presented a fast algorithm for the calculation of Betti numbers for isosurfaces and have illustrated a set of techniques that use Betti numbers to provide information about isosurface topology. Calculating Betti numbers for isosurfaces requires time and space proportional to the size of the isosurfaces.

There are other applications where Betti numbers provide valuable information about the data, such as detecting when molecules bond in a time-varying data set. In addition, Betti numbers can be used to determine the accuracy of mesh reduction algorithms. Reducing a mesh must not change the topology of the extracted isosurfaces.

We plan extend our algorithms to more general non-manifold surfaces.

Acknowledgments

We would like to thank Chris Henze for help acquiring data sets to test our algorithm.

This work was supported by the National Science Foundation under contracts ACI 9624034 (CAREER Award), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI); the Office of Naval Research under contract N00014-97-1-0222; the Army Research Office under contract ARO 36598-MA-RIP; the NASA Ames Research Center through an NRA award under contract NAG2-1216; the Lawrence Livermore National Labora-

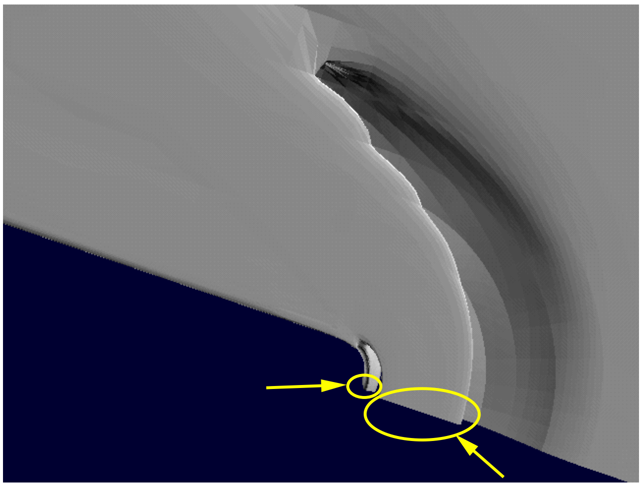


Figure 12 Isosurfaces of velocity magnitude of the the Blunt Fin data set, courtesy of NASA/Ames Research Center. Shown is the isosurface for the isovalue 1.3. Two tunnels are seen near the base of the fin. The yellow circles show the base of the tunnels.

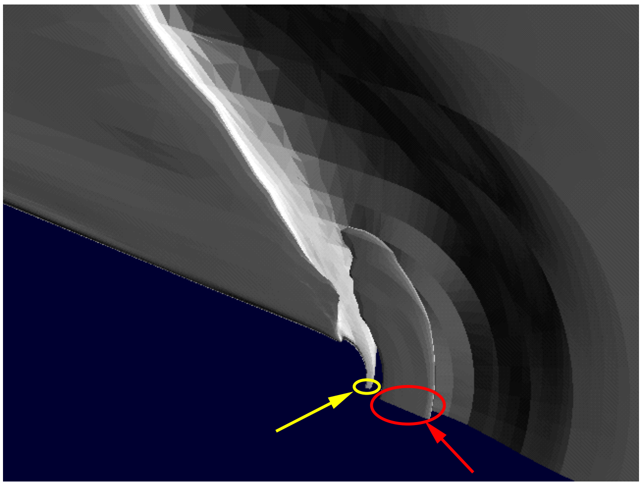


Figure 13 Isosurfaces of velocity magnitude of the Blunt Fin data set. Shown is the isosurface for the isovalue 1.9. The yellow circle indicates the base of the remaining tunnel. The red circle shows the region that had been the base of a tunnel for a lower isovalue, but the region split into a separate component in the current isovalue.

Isovalue	β_0	β_1	β_2	Meaning
0	1	1	0	
0.75	3	0	1	
0.8	1	1	0	
0.85	1	2	0	two stable tunnels appear indicating vortices near the base of the fin
0.9	1	2	0	
0.95	1	2	0	
1	1	2	0	
1.05	1	2	0	
1.1	1	2	0	
1.15	1	2	0	
1.2	1	2	0	
1.25	1	2	0	
1.3	1	2	0	
1.35	1	2	0	
1.4	1	2	0	
1.45	1	2	0	
1.5	1	2	0	
1.55	1	2	0	
1.6	1	2	0	
1.65	2	2	1	another component appears briefly
1.7	2	2	1	
1.75	3	1	1	one of the stable "tunnels" splits off into another component
1.8	2	1	0	
1.85	4	1	2	topology becomes less stable and meaningful
1.9	2	2	0	
1.95	3	1	1	
2	3	1	1	

Table 1 Isovalues and Betti numbers for the Blunt Fin data set.

tory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159; the Lawrence Berkeley National Laboratory; the Los Alamos National Laboratory; and the North Atlantic Treaty Organization (NATO) under contract CRG.971628. We also acknowledge the support of ALSTOM Schilling Robotics, and SGI. We thank the members of the Visualization and Graphics Research Group at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis.

References

[1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*. MIT Press and McGraw-Hill Book Company, 6th edition, 1992.

[2] Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995. ISSN 0167-8396.

[3] Leonidas J. Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graphics*, 4(2):74–123, April 1985.

[4] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.

[5] P. Moran, C. Henze, and D. Ellsworth. The FEL 2.2 user guide. Technical report, National Aeronautics and Space Administration, 2000. NAS-00-002.

[6] Gregory M. Nielson and Bernd Hamann. The asymptotic decider: Removing the ambiguity in marching cubes. In *Visualization '91*, pages 83–91, 1991.

[7] J. J. Rotman. *An Introduction to Algebraic Topology*. Springer-Verlag, New York, 1988.

[8] Yong Zhou, Baoquan Chen, and Arie Kaufman. Multiresolution tetrahedral framework for visualizing regular volume data. In Roni Yagel and Hans Hagen, editors, *IEEE Visualization 9́7*, pages 135–142. IEEE, November 1997.