Immersion of Self-Intersecting Solids and Surfaces: Supplementary Material 2: Tetrahedralization

Yijing Li and Jernej Barbič, University of Southern California

May 15, 2018

1 Inside-outside test without forming the region boundary surface

In this section, we prove the statement from Section 7 of the main paper that we do not need to form the region boundary surface (which would require using CSG) to perform inside-outside region queries. The closest site theorem of Baerentzen [Bærentzen and Aanæs, 2005] states that a point is located outside of a closed manifold non-self-intersecting mesh if and only if the vector from the closest site on the mesh to the query point has a positive dot product with the outward surface pseudo-normal. Pseudo-normals are defined in [Bærentzen and Aanæs, 2005]. We now generalize this test to non-closed non-self-intersecting manifold meshes M (i.e., meshes with boundary). We say that a point p is pseudo-outside of M, by definition, if the vector from the closest site on M to p has a positive dot product with the outward pseudo-normal at the closest site.

Theorem: Let p be a point inside a tet, and let R be one of the regions (both + and − regions are ok) of this tet, as defined in Section 7 of the main paper. Then, p is inside the region if and only if p is pseudo-inside the closest piece of R. This theorem also gives an algorithm to rapidly perform the inside-outside test: by this theorem, it is sufficient to perform pseudo-tests against all the pieces of a region. This means that the boundary mesh of R does not need to be explicitly formed on the faces of the tet. We perform the point-vs-triangle distance calculations for the pseudo-tests using exact arithmetic.

Proof: Suppose a query point p is inside a tetrahedron, but outside of a region. Then, we claim that the closest site to p on the region has to be on a triangle on one of its pieces. If this was not the case, the closest site is on the tetrahedron surface. Consider the line segment joining this closest site and p. Because the tetrahedron is convex, the line segment is completely inside the tetrahedron. If the segment does not intersect any pieces, this implies that the line segment never leaves the region, which is a contradiction with p being outside of the region. Therefore, it intersects one piece, but then this intersection is even closer to p than the original closest site, which is a contradiction. Because the closest point is therefore on a boundary triangle, we can now apply Baerentzen's theorem, which proves that our algorithm will in this case correctly identify that p is outside of the region.

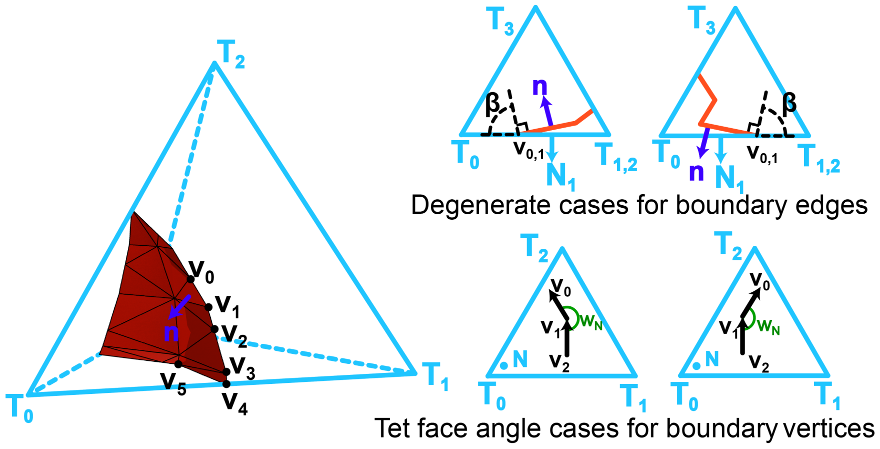
Suppose the query point p is inside the region. Let P be the piece of this region which has the closest point to p compared to the other pieces. Piece P cuts the tetrahedron into two parts, e_0, e_1. With out loss of generality, we assume e_0 does not contain p, and therefore, e_1 contains the region. Since p is outside e_0, according to the proof in the last paragraph, the closest site on e_0 to p must be on P and the pseudo-normal test on P to p should report outside for e_0. Since e_0 and the region share the piece P, P must have different orientations for e_0 and the region. Therefore, the pseudo-normal test on P for the region should give the opposite result to the test on P for e_0, which proves that our algorithm correctly reports that p is inside the region. ■

2 Pseudo-normal on the piece boundary

Although we do not need to form the complete region mesh, we still need the tet faces to compute the pseudo-normal if the closest site to p on a piece is on the boundary of the piece, i.e., it is on a tet face. The pseudo-normal that we compute must be identical to the one that would be computed using a complete region mesh. We achieve this as follows. We have to consider two separate cases.

Boundary edges: We first consider the case where the pseudo-normal is located in the interior of a boundary edge of a piece. Figure 1 explains the procedure to find the correct pseudo-normal. For the boundary edge v_1v_0 in the figure (left), we first determine on which tet face this edge lies. Then, we compute the pseudo-normal $\tilde{n}$ as the average of the triangle normal n on the piece neighboring v_1v_0, and tet face normal $N : \tilde{n} = \text{unit}(n + N)$. As described by Baerentzen's work, the final inside-outside result is the sign of $\text{dot}(p - c, \tilde{n})$, where p is the query position and c is the closest site on v_1v_0. Degeneracy may occur if n is close to $-N$, or $p - c$ is almost perpendicular to $\tilde{n}$. In either case, we reach the situation shown in Figure 1, top-right. Looking along the direction of the boundary edge v_0v_1, the tet face $T_0T_1T_2$ is seen as a line segment $T_0T_{1,2}$, and the edge v_1v_0 is seen as a point $v_{0,1}$. When n is close to $-N$ (left sub-figure), we observe that the angle at $v_{0,1}$ of the triangle $T_0v_{0,1}p$ is less or equal to the angle β, because otherwise, the closest site will not be on v_1v_0. Since β is entirely outside the region, we can safely classify p as outside. When $p - c$ is almost perpendicular to $\tilde{n}$, n must be close to N (right sub-figure). In this case, the angle at $v_{0,1}$ of the triangle $T_{1,2}v_{0,1}p$ is less or equal to β, so we can safely classify p as inside.

Boundary vertices: We now consider the case where we need the pseudo-normal on a boundary vertex of a piece. For one boundary vertex v, we first find the location of this vertex and its two neighboring boundary edges to find whether we are in the case of v_1 (interior of tet face) or v_4 (on tet edge), shown in Figure 1, left. (1) In the case of v_1, by Baerentzen's theorem, the pseudo-normal is $\tilde{n} = \sum_i w_i n_i + w_N N$, where w_i is the weight for normal n_i on triangle i surrounding v_1, and w_N is the weight for the tet face normal N. Normal weights are computed as the triangle angle at v_1. To compute w_N, we first compute the angle α between edge v_1v_0 and v_1v_2. Depending on the orientation of the piece, w_N is either α or $2\pi - \alpha$. Our intuition for choosing between these two options is that the faces contributing to the pseudo-normal should have consistent orientations. Our specific procedure (illustrated in Figure 1, bottom-right) is as follows. Without loss of generality, assume the directions of edges v_2v_1 and v_1v_0 are consistent with the orientations of the piece triangles that they belong to. Then, if $\text{cross}(v_1 - v_2, v_0 - v_1)$ has an opposite sign to N, we deduce that w_N must be $2\pi - \alpha$. Otherwise, w_N has to be α. (2) In the case of v_4, we have $\tilde{n} = \sum_i w_i n_i + w_{N_1} N_1 + w_{N_2} N_2$, where w_{N_i}, N_i are the weights, and tet face normals neighboring the tet edge containing v_4. Let N_1 be the tet face normal of $T_0T_1T_2$ and N_2 be the tet face normal of $T_0T_3T_1$. Then, w_{N_1} is the angle between edge v_4v_3 and edge v_4T_i, and w_{N_2} is the angle between edge v_4v_5 and edge v_4T_i, where i is either 0 or 1. We again use the intuition that faces around v_4 should have consistent orientations.



Degenerate cases for boundary edges

Tet face angle cases for boundary vertices

Figure 1: **Handling closest sites at piece boundaries.** Left: a piece inside a tetrahedron $T_0T_1T_2T_3$, where T_3 is occluded by the piece. Vector N_1 is the normal of the tet face $T_0T_1T_2$. Vertices v_i are on the boundary of a piece. Vector n is the normal of the triangle with the boundary edge v_1v_0. Top-right: degeneracy cases for boundary edges. Bottom-right: tet face angle cases for boundary vertices.

Without loss of generality, assume that the directions of edges v_5v_4 and v_4v_3 are consistent with the orientation of the piece triangles that they belong to. We compute $\text{dot}(\text{cross}(v_4-v_5, v_3-v_4), T_0-T_1)$. If the resulting sign is positive, we select $i = 1$, otherwise $i = 0$. We perform these computations in exact arithmetic.

3 Meshing self-touching cells

Special care must be taken when the tetrahedralization algorithm of Section 7 in the main paper is used to mesh self-touching cells produced by our immersion algorithm. If a tetrahedron includes two regions that touch each other at a self-touching edge of the cell, then during the pseudo-normal test, we also check the distance between the closest site on the piece and the query point. If the distance is zero under exact arithmetic, we return the "outside" result for the pseudo-normal test.

References

[Bærentzen and Aanæs, 2005] Bærentzen, J. and Aanæs, H. (2005). Signed distance computation using the angle weighted pseudo-normal. *IEEE Trans. on Visualization and Computer Graphics*, 11(3):243–253.