

Intersection of a Triangle and a Cylinder

David Eberly

Geometric Tools, LLC

<http://www.geometrictools.com/>

Copyright © 1998-2010. All Rights Reserved.

Created: November 2, 2010

Contents

1	Introduction	2
2	The Algorithm	2
3	The Implementation	4

1 Introduction

The triangle has vertices \mathbf{P}_i for $0 \leq i \leq 2$. The cylinder has center \mathbf{C} , unit-length axis direction \mathbf{D} , radius r , and height h . Choose unit-length vectors \mathbf{U} and \mathbf{V} so that $\{\mathbf{U}, \mathbf{V}, \mathbf{D}\}$ is a right-handed orthonormal set; that is, the vectors are unit-length, mutually perpendicular, and $\mathbf{D} = \mathbf{U} \times \mathbf{V}$ (the algorithm still works if the set is left-handed). The (solid) cylinder is parameterized by $\mathbf{C} + x\mathbf{U} + y\mathbf{V} + z\mathbf{D}$, where $x^2 + y^2 \leq r^2$ and $|z| \leq h/2$. Both objects are assumed to be stationary, and the query is a *test-intersection* whose result is a Boolean indicating whether or not the objects intersect.

2 The Algorithm

Compute the representations of \mathbf{P}_i in the coordinate system of the cylinder; that is,

$$\mathbf{P}_i = \mathbf{C} + x_i\mathbf{U} + y_i\mathbf{V} + z_i\mathbf{D}$$

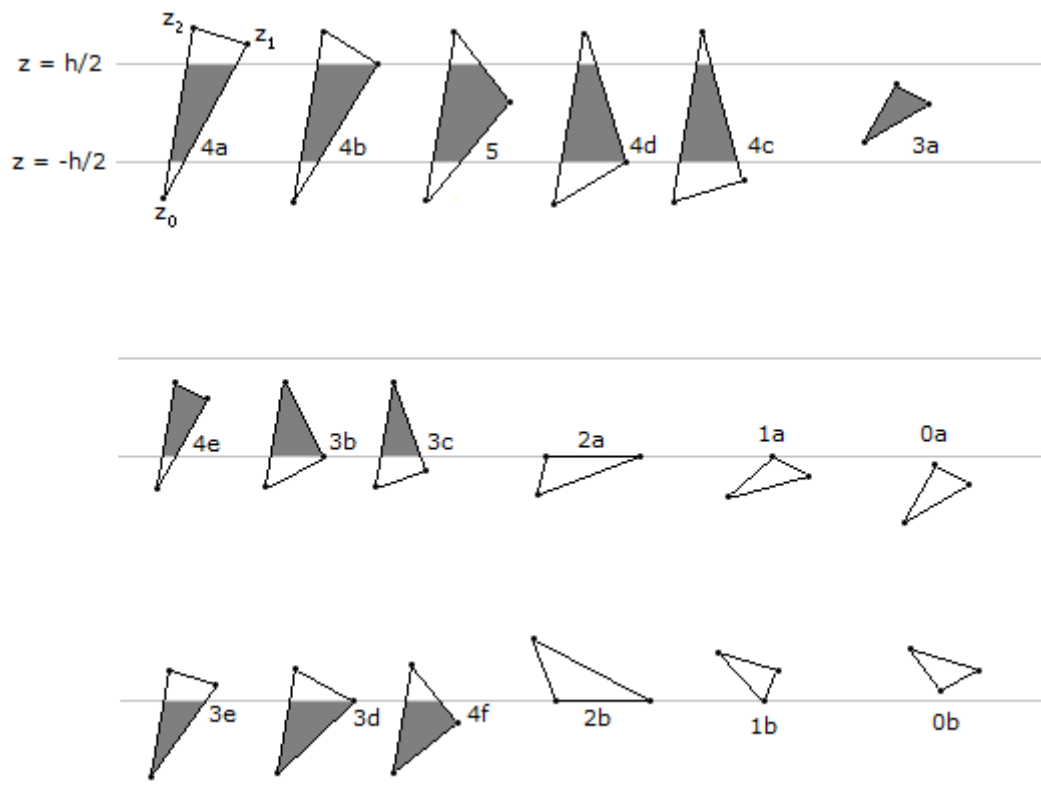
where $x_i = \mathbf{U} \cdot (\mathbf{P}_i - \mathbf{C})$, $y_i = \mathbf{V} \cdot (\mathbf{P}_i - \mathbf{C})$, and $z_i = \mathbf{D} \cdot (\mathbf{P}_i - \mathbf{C})$. Define

$$\mathbf{Q}_i = (x_i, y_i, z_i)$$

for all i . We want to determine whether the triangle whose vertices are the \mathbf{Q}_i intersects the cylinder $x^2 + y^2 \leq r^2$, $|z| \leq h/2$.

Clip the triangle to the slab $|z| \leq h/2$; that is, clip against two planes, $z = h/2$ and $z = -h/2$. The result is one of: a single point, a line segment, a triangle, a convex quadrilateral, or a convex pentagon. Before clipping, sort the \mathbf{Q}_i so that $z_0 \leq z_1 \leq z_2$. Figure 2.1 shows all the possibilities for the relationship between the triangle and the slab.

Figure 2.1 The relationship between the triangle and the z -slab.



The pseudocode for distinguishing among the cases is

```

if (z2 < -h/2) { case 0a }
if (z0 > h/2) { case 0b }
if (-h/2 <= z0 && z2 <= h/2) { case 3a }

if (z0 < -h/2)
{
    if (z2 > h/2)
    {
        if (z1 >= h/2) { cases 4a/4b }
        else if (z1 <= -h/2) { cases 4c/4d }
        else { case 5 }
    }
    else if (z2 > -h/2)
    {
        if (z1 <= -h/2) { cases 3b/3c }
        else { case 4e }
    }
    else
    {
        if (z1 < -h/2) { case 1a }
        else { case 2a }
    }
}
else if (z0 < h/2)
{
    if (z1 >= h/2) { cases 3d/3e }
    else { case 4f }
}
else
{
    if (z1 > h/2) { case 1b }
    else { case 2b }
}

```

The problem now is to determine whether the projection of the clipped triangle onto the xy -plane intersects the disk $x^2 + y^2 \leq r^2$.

If the clipped and projected triangle is a point $\mathbf{A}_0 = (x_0, y_0)$, then intersection occurs when $x_0^2 + y_0^2 \leq r^2$.

If the clipped and projected triangle is a line segment with endpoints \mathbf{A}_0 and \mathbf{A}_1 , compute the squared distance, d^2 from $(0,0)$ to the segment. Intersection occurs when $d^2 \leq r^2$.

Otherwise, the clipped and projected triangle is a convex polygon (triangle, quadrilateral, or pentagon). If $(0,0)$ is contained in the polygon, intersection occurs. If $(0,0)$ is not contained in the polygon, compute the squared distance, d^2 , from $(0,0)$ to each edge of the polygon. If any of the d^2 is less than or equal to r^2 , there is intersection.

If you have an LCP solver, you can set this up as a convex quadratic programming problem and use the

solver. The problem is to minimize $x^2 + y^2$ subject to 1 equality constraint (triangle is in a plane), 3 inequality constraints (point is in triangle), and 2 inequality constraints (point is between bounding planes of top/bottom of cylinder). If the minimum is d^2 and is less than or equal to r^2 , then the triangle and cylinder intersect.

3 The Implementation

The test-intersection query is in

```
WildMagic5/LibMathematics/Intersection/Wm5IntrTriangle3Cylinder3.{h,cpp}
```

A sample application to test the code is in the folder

```
WildMagic5/SampleMathematics/IntersectTriangleCylinder
```

A red cylinder and a blue triangle are displayed. You can select the object that is rotated by the virtual trackball by pressing keys. Press '0' to rotate the scene (both objects). Press '1' to rotate the triangle. Press '2' to rotate the cylinder. Whenever the triangle and cylinder intersect, the triangle changes color to green.