Lecture 6: Manipulation of 3D Objects

In building graphical scenes from a number of sub-objects, it is frequently necessary to discover how one object interferes with another. We may wish to do this to remove hidden lines from parts of a wireframe representation, or alternatively we may wish simply to check for colisions between objects, which, in the design of chemical plant piping for example, would indicate errors in the design. This study normally involves two fundamental tests: *containment*, which checks to see if a point is inside an object and *clipping*, which determines where a line (or polygon) intersects an object.

Containment within a convex volume

A simple definition of a convex volume is one where the line joining any two points on the surface is completely contained within the volume. An object produced by the intersection of infinite planes is always convex. The convexity of an object bounded by planar facets can be tested by the following simple algorithm (see Diagram 5.1):

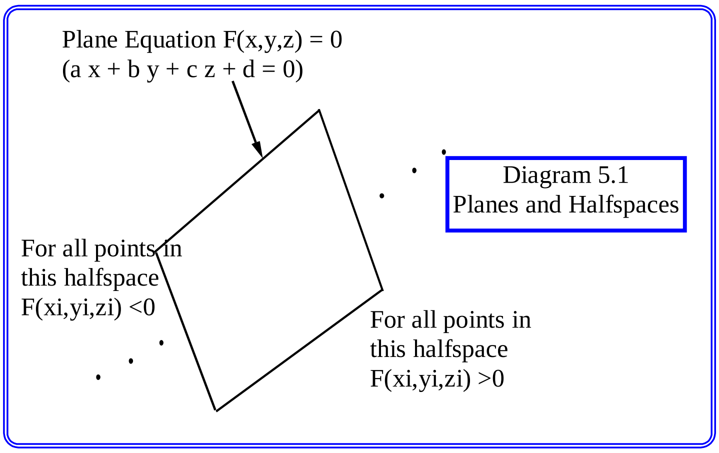

Diagram 5.1
Planes and Halfspaces

convex:=**true**;
for each planar facet **do**
begin
{ find the functional equation of the facet plane f(x,y,z)=0 }
for all other vertices not belonging to this facet **do**
    if sign(f(xi,yi,zi))<>sign(previous vertex)
    then convex:=**false**;
end

This is not a particularly efficient algorithm but it works for all cases. For convex objects there is an easy test to determine whether a point **P** is contained within that volume. Consider a line drawn from the point under test to any point on any surface of the convex volume as shown in Diagram 5.2. If the point is inside the volume then the angle made by the line to the surface inward normal is always acute (or zero).

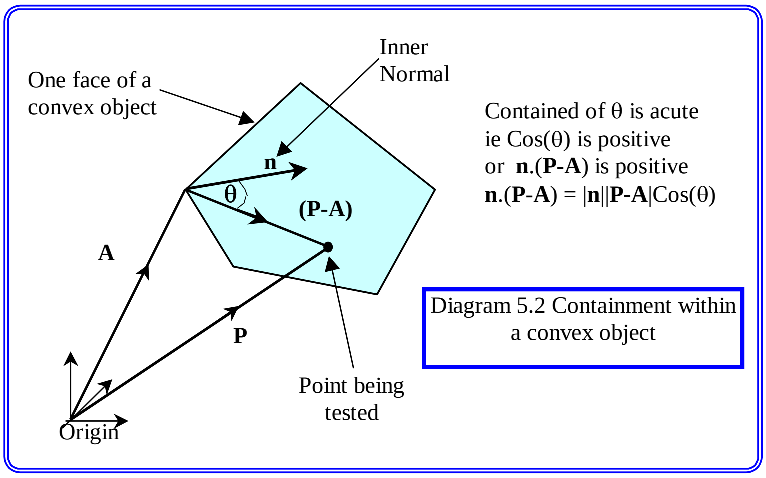

Diagram 5.2 Containment within a convex object

$$\mathbf{n}.(\mathbf{P} - \mathbf{A}) >= 0$$

A is a point on the surface, **n** is the inner normal of the surface. As shown in Diagram 5.3, we do not necessarily know the direction of the surface normal. To find the normal to any surface we take the cross product of two edges, and to determine its sign we test it with a vertex of an adjoining surface as shown in

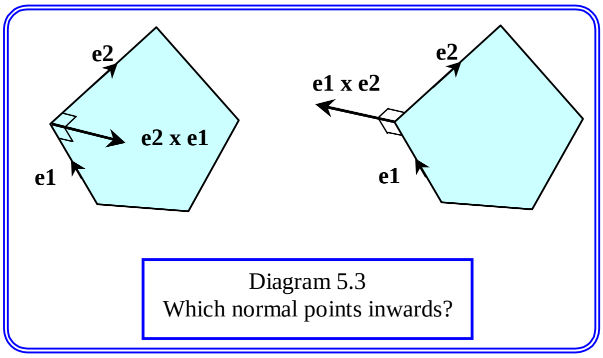

Diagram 5.3
Which normal points inwards?

Diagram 5.4. The inner surface normal is a useful quantity to store in the data structure describing the object.

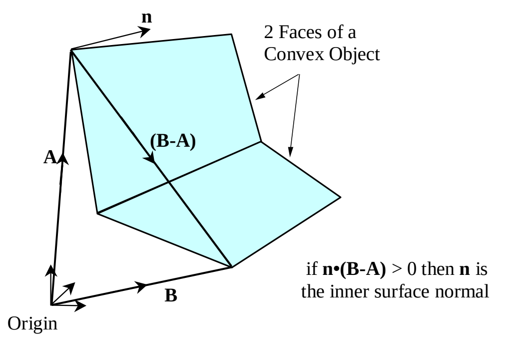

Diagram 5.4: Checking for the inner surface normal

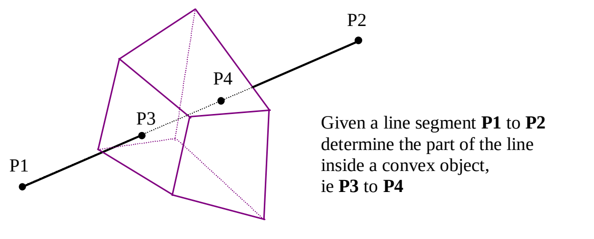

Diagram 5.5: Clipping to a convex volume

Clipping to a convex volume (Cyrus Beck Algorithm)

This algorithm is a simple extension of the containment algorithm given above. Given a line joining **P1** and **P2** and a convex volume, we wish to find the part of that line that lies completely within the volume as shown in Diagram 5.5. For each surface, we take a point **A** on the surface, which could simply be one of the vertices, and examine the signs of:

n.(**P1** - **A**) and **n.**(**P2** - **A**)

where **n** is is the inner surface normal. The following cases are possible:

Both signs -ve or zero: The line is completely outside the body and the algorithm terminates returning null.

Both signs +ve or zero: The line could be inside the body, but it is not clipped by this plane.

Both signs zero: The line lies on the plane. Depending on the desired result, either finish or proceed to the next surface.

One sign negative, one sign positive:

Find the intersection of the line and the plane by solving:

n.(μ**P2** + (1-μ)**P1** - **A**) = 0.

Replace the point that yielded the negative sign by the intersection point. When all surfaces have been considered, the remaining line span is the required result.