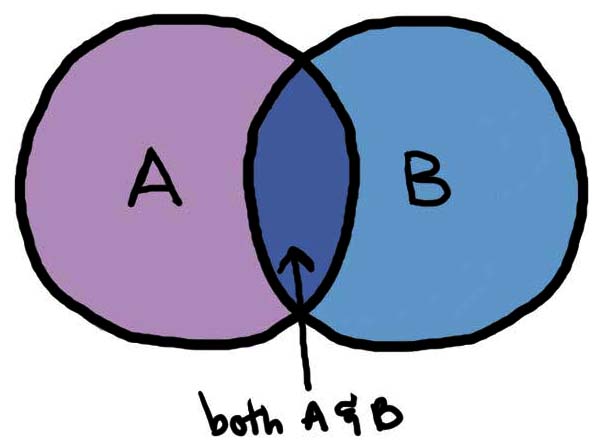
Physics Tutorial 8 – Collision Manifolds



**Summary**

In this tutorial we be expanding our collision detection procedure in order to accurately generate a collision manifold. We will be covering what collision a manifold entails, along with a discussion of the clipping method – which will become the main method of computing the collision manifold in this tutorial series.

**New Concepts**

Contact points, collision manifold, clipping method, Sutherland-Hodgman clipping

**Introduction**

At this point we have identified when two objects have collided and retrieved the collision normal as well as the penetration distance. However, one more piece of information is required before being able to solve the collision; the contact points. In this tutorial we will be covering what a collision manifold is and how it can be computed using clipping.

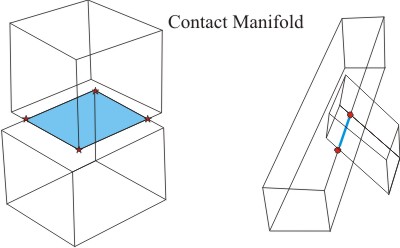
**What is a contact point?**

At the moment we have the direction of the collision (normal) and the penetration distance. However if this was all that was used to resolve a collision between two objects then no rotation would ever occur in our physics engine.

A contact point describes a point at which two objects touch. Which can be used to resolve collisions in the form of a distance constraint, to constrain the two objects from overlapping in the following time step.

**What is a Manifold?**

A manifold is a collection of contact points that form all of the necessary constraints that allow the object to properly resolve all penetrations. It can be seen as the summation of the surface area between two colliding objects. As shown below, this could form either a single point, a line or a 2D polygon.

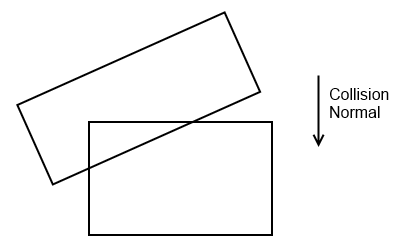


In a discrete physics system, this can pose a problem as collisions are only detected after the two objects are already overlapping. This results not in a 2D surface area where the two objects are touching but a 3D volume.

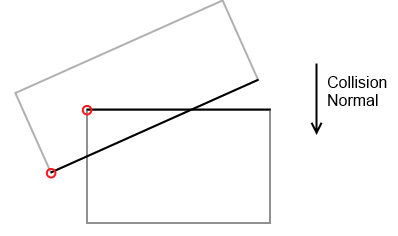
To overcome this, we infer the contact manifold as if the two objects were touching.

**Implementation**

To compute the manifold we will be using the clipping method, in which we will be progressively clipping a face of one object with the perimeter of the second object. This results in a 2D collision manifold which can then be used in our resolution calculations.

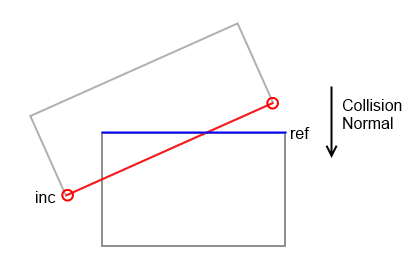
The best way to show how this algorithm works is through an example. 

So if we look at the following case where two boxes have collided. At this point we have just finished SAT and know the collision normal and penetration depth.

**Identifying the significant faces**

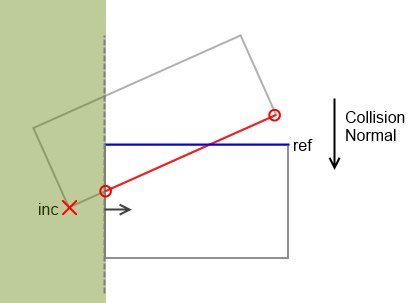
The next step is to identify the significant faces that are intersecting. This is accomplished by selecting the vertex furthest along the collision normal, in this case the red vertices. Then we pick a face on each object whose normal is closest to parallel with the collision normal that includes the selected vertex. This gives us the two most significant faces for contact generation.

N.B. The normal is inverted when selecting the second objects vertex.

**Calculating the incident and reference faces**

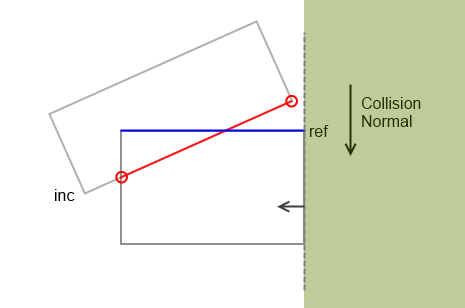
The reference face will become the point of reference when clipping occurs in the following steps. The incident face will in turn become a set of vertices that will be clipped.

To do this we compute which of the two significant faces have a normal that is closest to parallel with that of the collision normal. In this case the blue line is closest to parallel, and such becomes the reference face. The other face then becomes the incident face which we will clip to generate the contact points, currently it is comprised of two vertices.

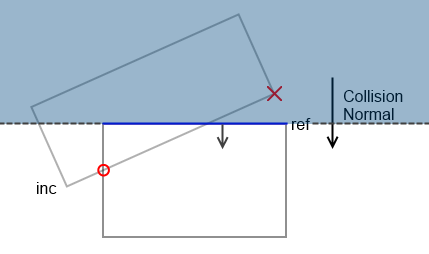
**Adjacent Face Clipping**

We now clip the incident with all the adjacent faces of the reference. This is done by taking the adjacent face’s normal and any vertex that it contains to produce a plane equation.

The algorithm we use to compute the clipping is known as Sutherland-Hodgman Clipping and can be easily adapted to the 3D scenario.

The first clipping plane we will do in this example is the left hand face. As one of the vertices of the incident face lies within the clipping region, it will be replaced with a vertex that lies on the edge of the clipping plane.

The second clipping plane is that of the right hand face. None of the points of the incident face lie in the clipping region, so no changes will be made.

**Final Clipping**

The final clip plane is that of the reference face itself. However instead of clipping the incident face like in the previous stage, we now just remove all points that lie inside the clipping region. In this example this leaves us with just a single contact point and not a line/edge.

Although at first glance it seems like we are ignoring critical contact points, this is in fact correct. As what we are trying to infer is the points of contact of when the two objects first touched, not now when they are overlapped. So in this example only the corner of the reference face would be in contact with the other object. Which in our inferred result, is directly along the collision normal.