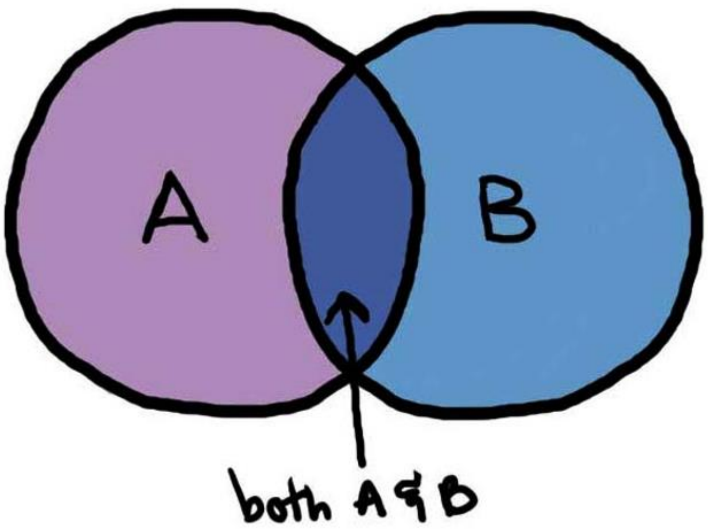
Physics Tutorial 5: Collision Manifolds



New Concepts

- Contact Points
- Collision manifold
- Clipping method
- Sutherland-Hodgman Clipping

Contact Points

Contact Points vs. Point of Greatest Penetration

- If we consider the discussion of contact points up to now, it’s been in terms of point of greatest penetration
- Contact point being treated as the point of one object which has penetrated most deeply into the second object; we’ve then discussed pushing that point along the normal (or pushing both objects half the distance) as a means of resolving this
- While that’s a reasonable enough position to take, we should think a little more philosophically about what the point represents

Contact Points vs. Point of Greatest Penetration

- Rather than thinking in terms of ‘how deep the interface is’, we should instead view our contact points as the points which were in contact with our penetrated object at the *instant* of collision

- In that sense, what we’re trying to do is rewind time to the point during our time step where our objects collided, and imagining what that moment of collision looked like

Contact Points vs. Point of Greatest Penetration

- In viewing the contact point this way, it becomes obvious how it maps to our constraint-based solver
- The contact point is the point on the surface of our object at which our constraint is applied
- The distance from the contact point to our object's centre of mass is the distance at which any resultant torque from the collision is being applied
- And so on.

Contact Points vs. Point of Greatest Penetration

- But will one point always convey enough information to resolve a collision?
- Nope. Which is why we employ the collision manifold.
- Collision manifold represents scenarios where more than one ‘point’ was in contact at the moment of penetration
- Consider the differences between a tyre rolling down the street, and a penny

Consider the differences between a tyre rolling down the street, and a penny…

- Coin has one contact point, normally
- Never deformable
- Can fairly accurately be represented with a single contact point

Consider the differences between a tyre rolling down the street, and a penny…

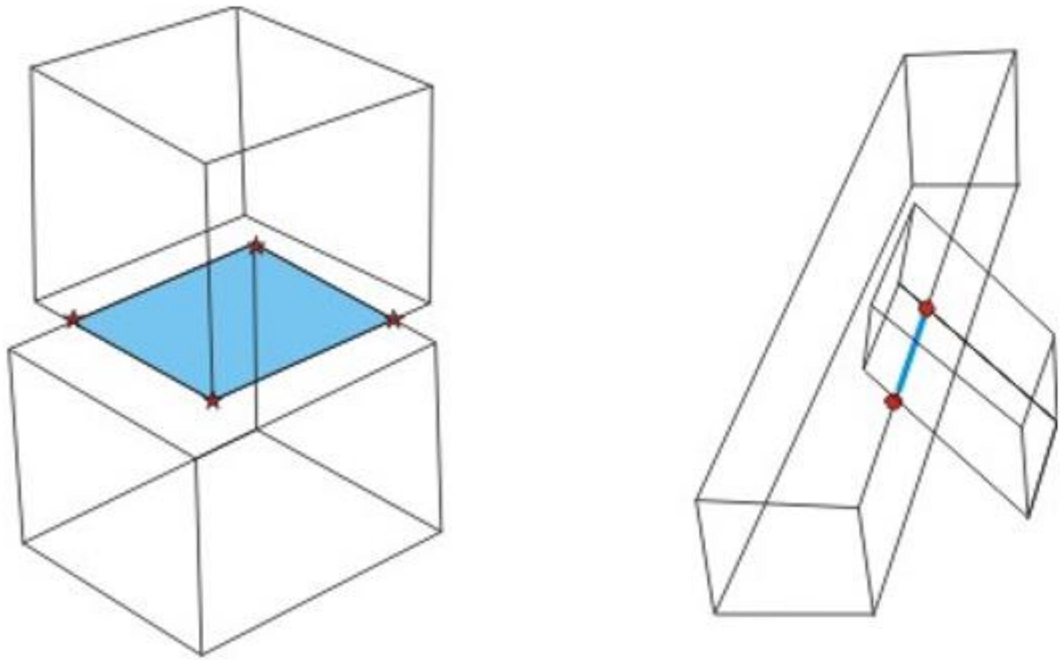
- Tyre has multiple contact points
- Is deformable?
  - Then has variable numbers of contact points
- Not deformable?
  - STILL has variable numbers of contact points

The Collision Manifold

What is a Collision Manifold?

- A collection of points
- These points form all constraints which are required to resolve all interfaces
- Can be considered the ‘surface area’ between two colliding objects
- Could be a point, a line, or a polygon

What is a Collision Manifold?



What is a Collision Manifold?

- This poses a problem for a time stepped simulation
- The actual overlap between objects is in three dimensions (a volume), not in two (an area)
- In real life, because we don’t interpenetrate with objects we collide with, collision manifold would always be an area rather than a volume, even if that area were deformed (different problem which we don’t address in this tutorial series)

What is a Collision Manifold?

- And this brings us back to the point at the beginning of the lecture, regarding collision manifolds/contact points representing a point or area on the surfaces of the objects which collide, not a point inside their volume

- Collision manifolds, therefore, are populated on the basis of what ‘would’ have been touching at the time of collision

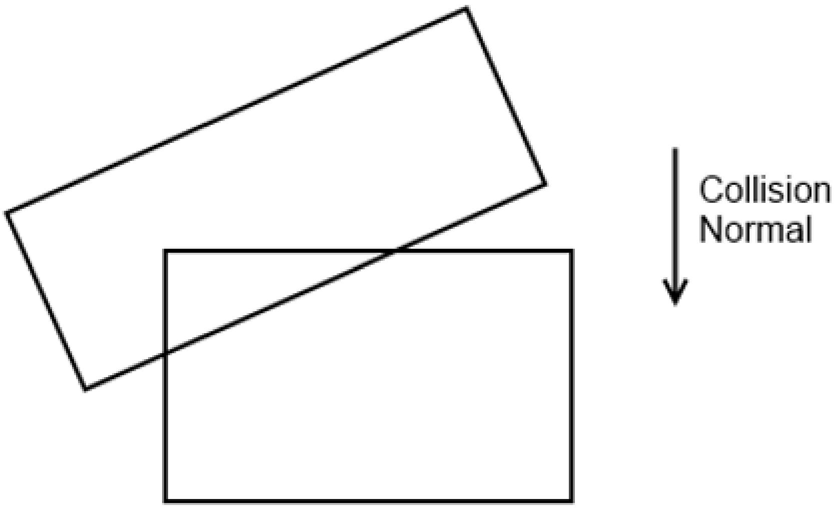
The Clipping Method

The Clipping Method

- The way we determine this manifold at the point of collision involves clipping a face of one object with a surface (perimeter) of a second
- Unimaginatively, this is called the clipping method
- When applied to a 3D environment, this results in a (maximum) 2D manifold which is used to resolve collisions

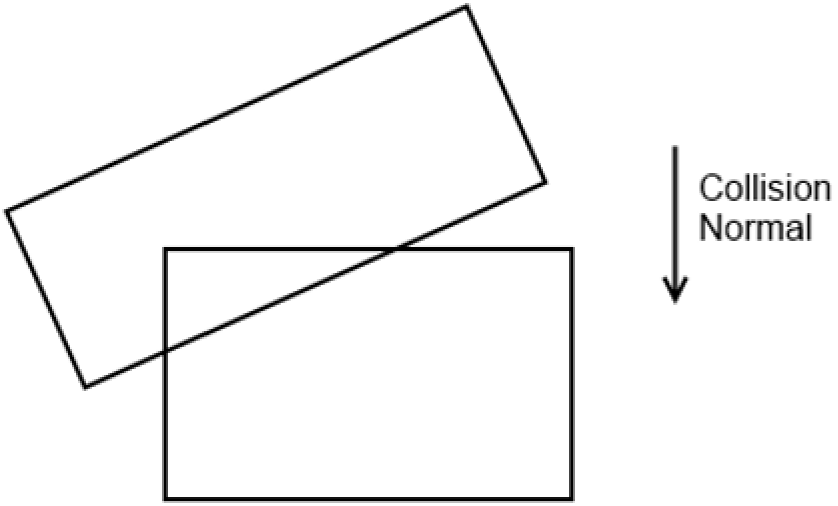
The Clipping Method

- Consider the example below in which two boxes have collided.
- At this point in our execution, we have run our SAT-based collision detection algorithm, and obtained N and p



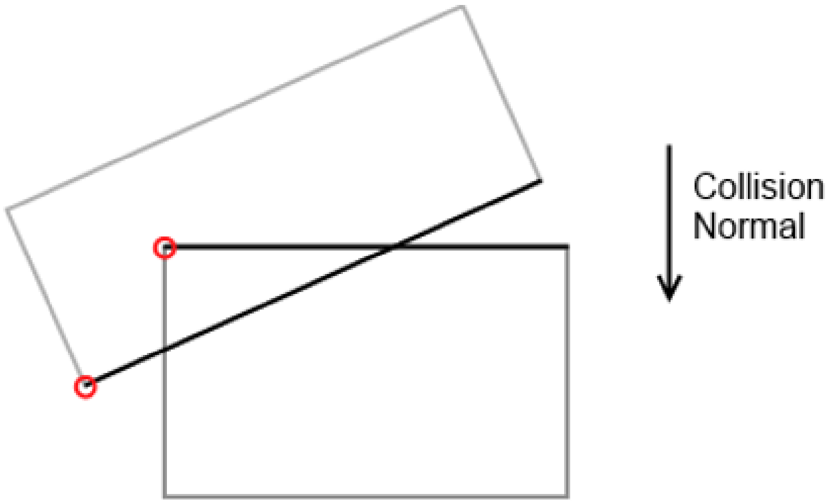
Identify the Significant Faces

- The first step to determining the collision manifold is to resolve the ‘significant faces’ – those involved in the collision



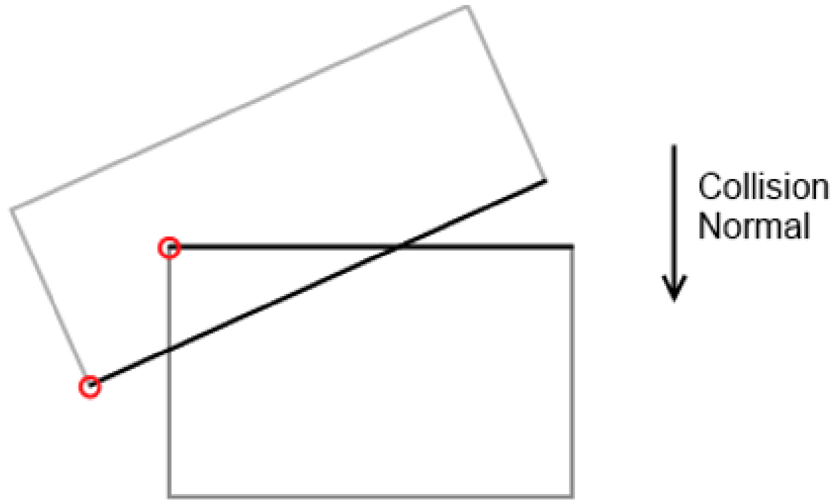
Identify the Significant Faces

- The first step to determining the collision manifold is to resolve the ‘significant faces’ – those involved in the collision



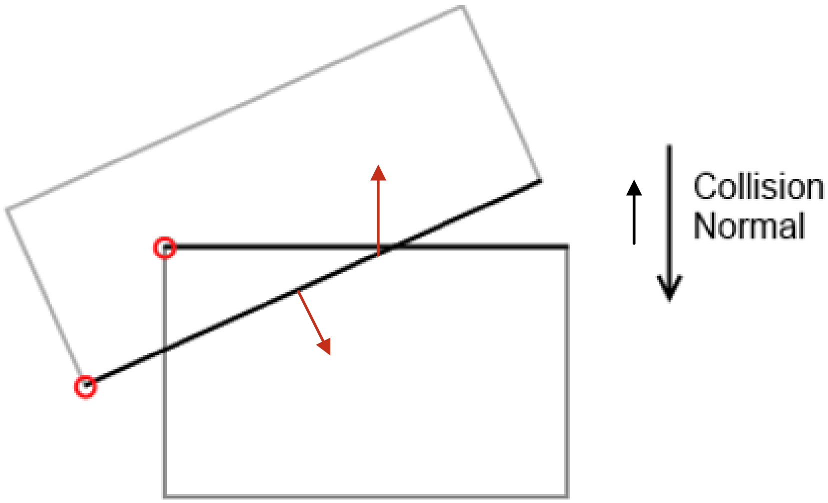
Identify the Significant Faces

- These faces are selected based on the following criteria:
  - The face includes the vertex furthest along the collision normal (see the red circles)
  - Of the applicable faces, its normal is closest to parallel with the collision normal



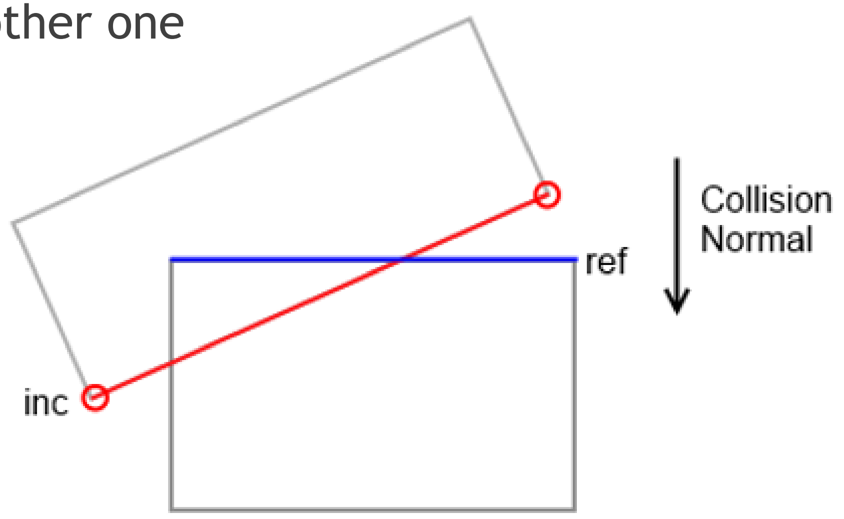
Identify the Significant Faces

- These faces are selected based on the following criteria:
  - The face includes the vertex furthest along the collision normal (see the red circles)
  - Of the applicable faces, its normal is closest to parallel with the collision normal



Calculating the 'Incident' and 'Reference' Faces

- At this point, we need to decide which face will be used as the reference for the clipping algorithm, and which face is actually going to be clipped
- The reference face is the face whose normal is closest to parallel with the collision normal (blue). The incident face is the other one



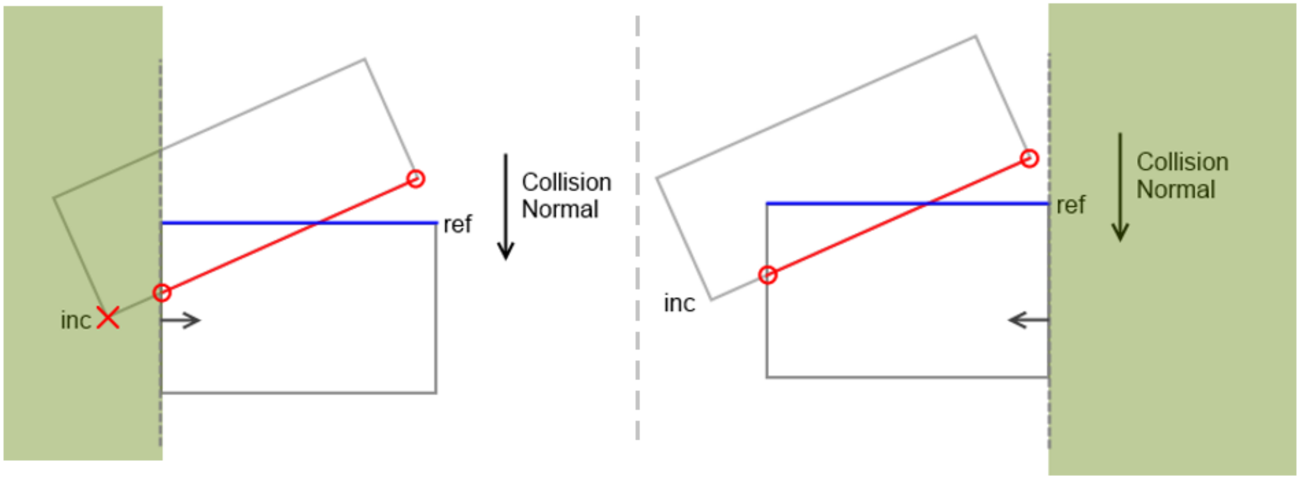
Sutherland-Hodgman Clipping

The Algorithm vs the Principle

- While we explore Sutherland-Hodgman Clipping explicitly, it is as an illustrative tool to help you understand what clipping is actually doing
- You're free to implement alternative clipping algorithms if you wish, and to explore the domain further on your own time
- The important thing is that you understand why clipping is taking place, which the remainder of this talk should help with

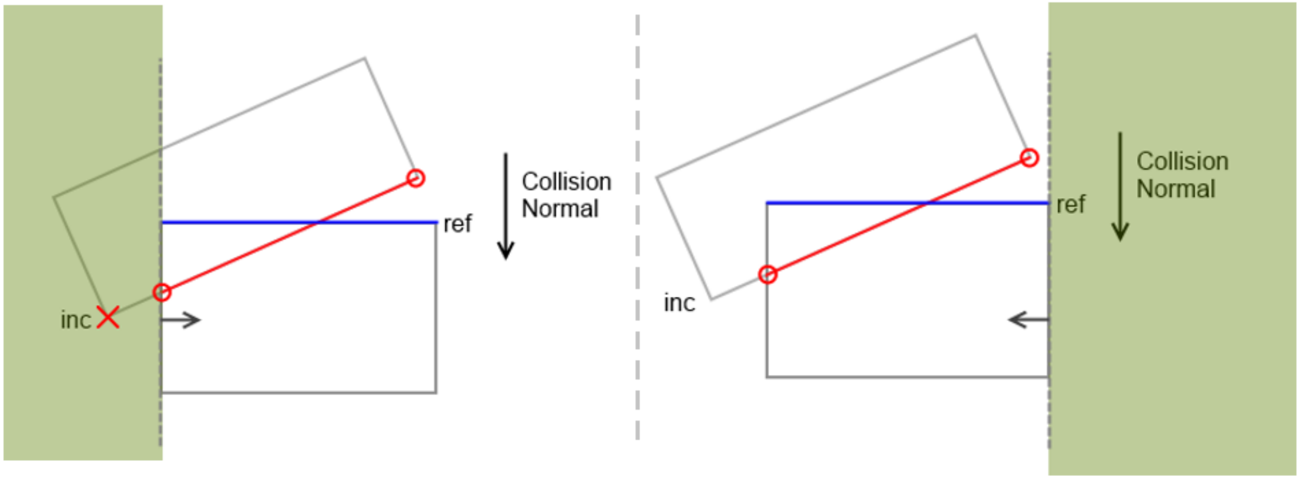
Adjacent Face Clipping

- We’ve already established that the face of the falling box is the incident face, so all clipping will be applied to that face
- The first step is adjacent face clipping, and in our example occurs on both the left and right hand sides
- Easily visualised below



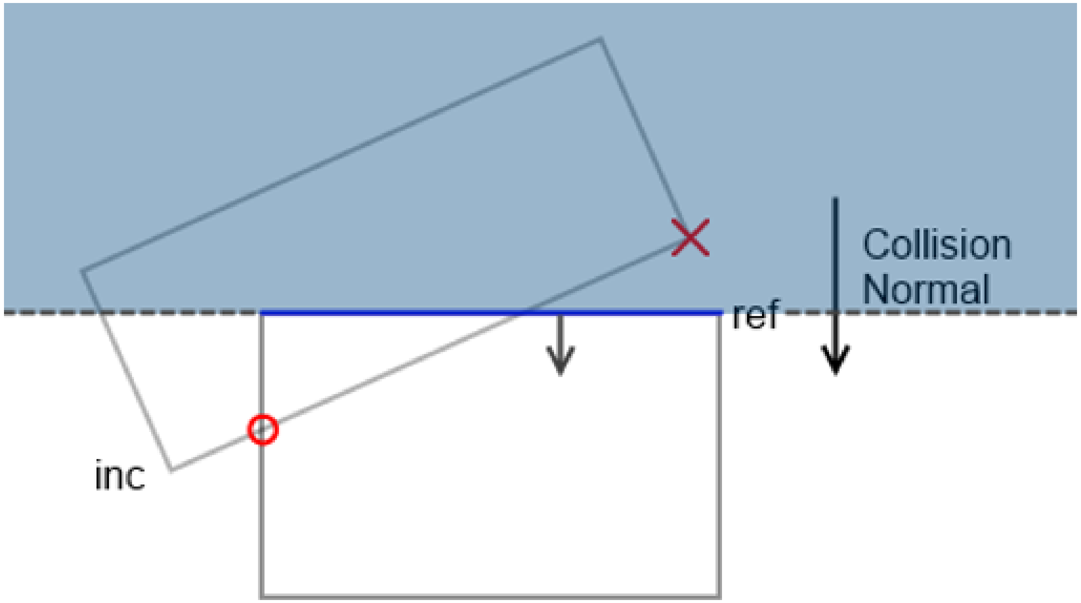
Adjacent Face Clipping

- In the left hand case, we see that one of the vertices of the incident plane lies within the clipping region (we ‘reach’ the vertex before we reach the reference plane)
- That vertex is replaced with a vertex which lies on the edge of the clipping plane
- In the right hand case, nothing changes – we reach our reference plane before we reach a vertex on the incident plane



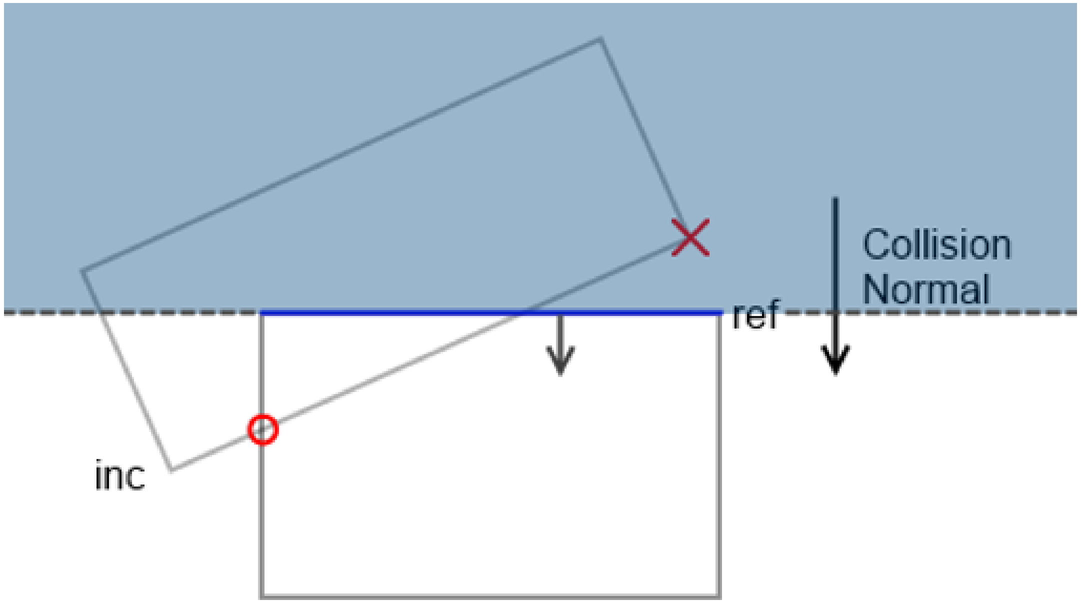
Final Clipping

- At this point, we prune points in the direction of the collision normal, until we reach the reference plane
- This is similar to a ‘perpendicular’ version of adjacent face clipping, except in this phase we simply delete any vertices we encounter, rather than clipping them



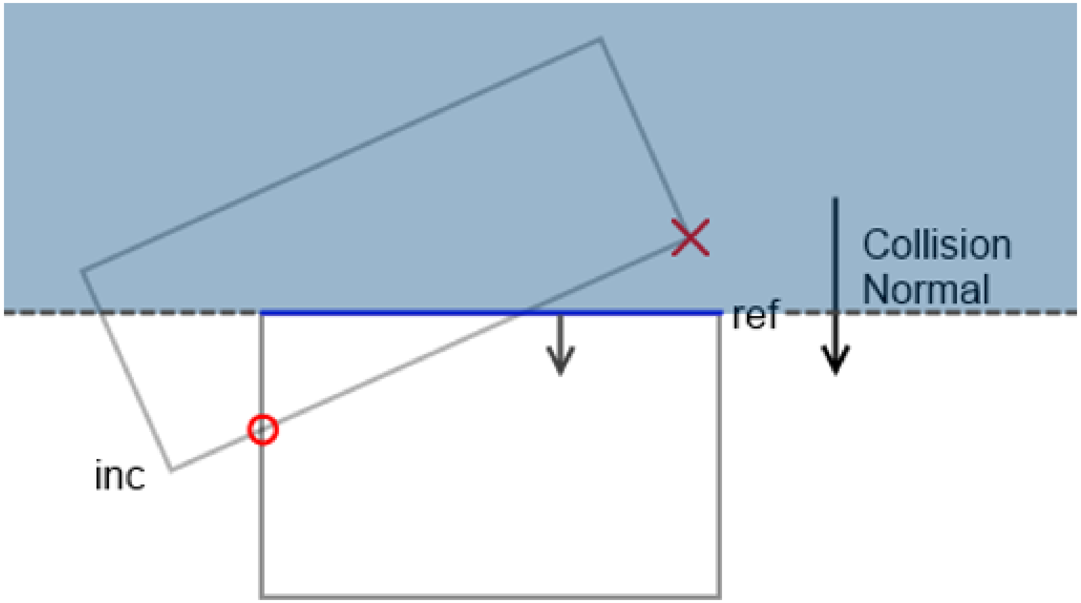
Final Clipping

- This might seem counter-intuitive, but actually it makes sense if we think fourth dimensionally.
- The vertex on the right is elevated significantly above the (clipped) vertex on the left



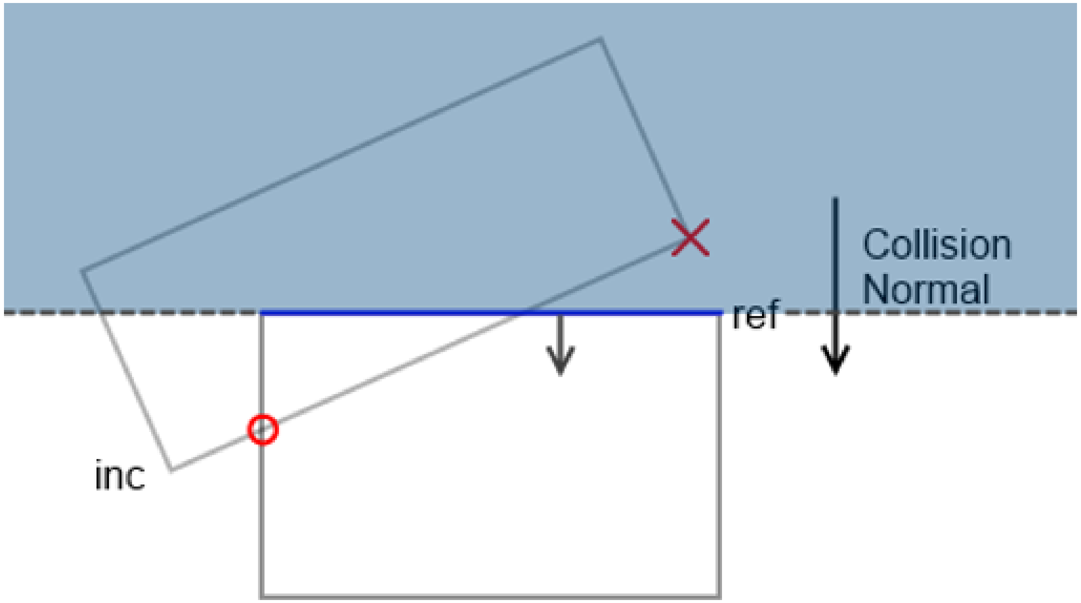
Final Clipping

- Since we want to know which points were in contact at the instant of collision, we need to bear in mind the orientation of the colliding object
- For the right hand vertex to have been involved, the upper box would need to have been perfectly parallel to the lower



Final Clipping

- The result of our clipping is a list of vertices which were in contact at the moment of collision – in this case, only one vertex was in contact, but it's easy to envision a scenario where more points would have been involved
- The computed manifold will be the focus of our solver



A note on Approximations

- The code included in the handout, as always in physics, gives approximate answers
- These answers are always valid, becoming more accurate the closer the reference normal gets to the collision normal.
- It compares distance between the clipped vertex and the vertex of the reference face furthest along the collision normal
  - This isn't entirely correct
  - Can be corrected with the inclusion of plane ray intersection
  - Can look into it on your own time

Summary

- Discussed contact points and contact manifolds
- Introduced the idea of clipping to obtain a collision manifold
- Provided an example of a specific clipping algorithm to generate a collision manifold

Implementation

- Implement the code at the end of the hand-out – you will now have every property required to generate solutions to collisions within your environment. That will be the focus of the final pair of physics tutorials.