Houdini 16 Objects and Collisions

Introduction

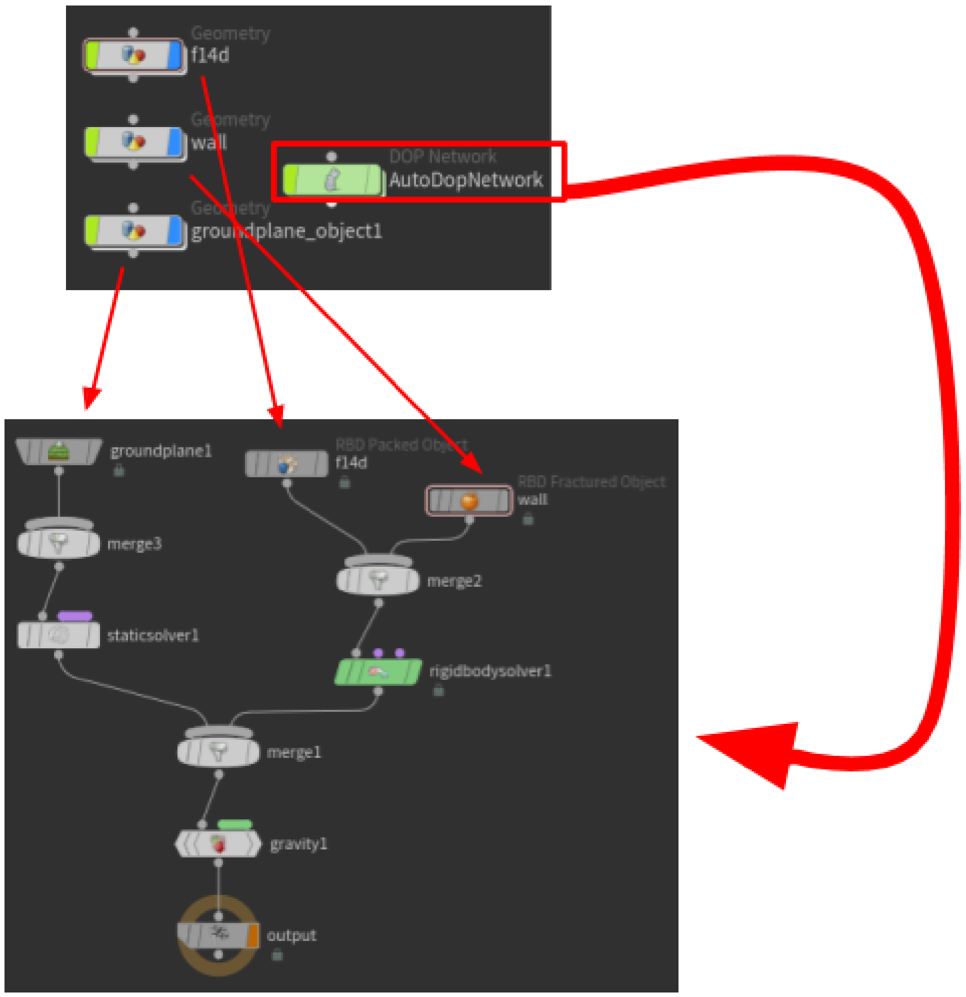
This physics portion of Houdini is called dynamics. There's a ton of different things you can do with dynamics, but this text is specifically about handling collisions between objects. You can use this instead of normal keyframe animation to get realistic looking animations going. All the basic parameters are there: gravity, velocity, angular velocity, density, bounce, friction, etc.. etc..

The shorthand for dynamics is DOPs (Dynamic OPerators).

Dynamics Network

The following is an example of a DOPS network. This specific network (AutoDopNetwork) is automatically created by Houdini when you use the rigid-body shelf, but networks that you create manually will look very similar.

Essentially what happens is that, you have your various geometry nodes at the /obj level. In your DOP network, you create DOP rigid-body/static/collision nodes that reference those geometry nodes. Those rigid-body/static/collision nodes are then passed to “solvers” and output.



NOTE: If you’re using a AutoDopNetwork and your nodes aren’t organized, hit L to layout the nodes in a nice tree structure.

There are lots of different types of solvers, but the 3 most important ones are…

- Static → only collides with stuff, never moves
- Rigid body → collides with stuff and moves, but is rigid
- Finite element → collides with stuff and moves, but is squishy

  NOTE: What are solvers? Solvers is a fancy name for “rule book”. Essentially all it does is contain the logic to do physics simulations. You’ll notice that there are 2 types of solvers in the example: rigidbody and static.

  Static is for static collision geometry (stuff that can be collided with but never moves). Rigidbody is for objects that have physics actually applied (e.g. rag-doll on a model).

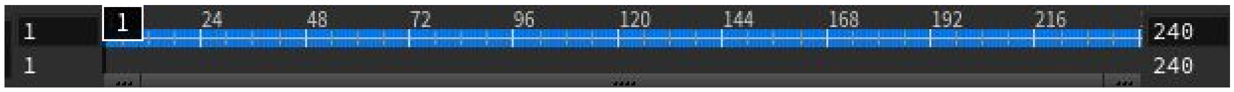
  NOTE: What about gravity? That’s handle by the gravity node just before the output node.

  NOTE: All of this is just an interface to bullet. I imagine bullet’s actual interface is nothing like this and more like what Box2D is like. For example, gravity would likely be a system-wide property that can only be set once. WTF happens if I have 2 gravity nodes?

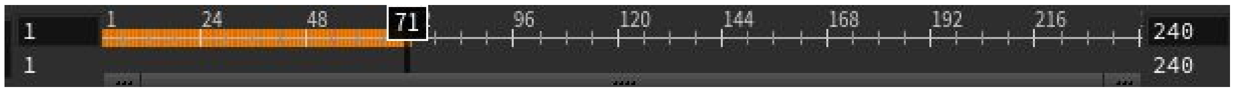
Dynamics Playback

Once you create your dynamic bodies and stuff, the calculations for them be done AS YOU SCRUB YOUR ANIMATION SLIDER / AS YOU PLAY YOUR ANIMATION.

Calculations are done once and then cached. You’ll know which frames are calculated because the timeline will show blue for those frames.

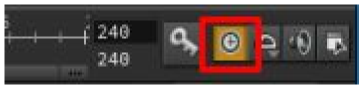


If you change something, the cache will get nuked and the computation will happen again the next time you scrub/play from frame 1. YOU HAVE TO GO BACK TO FRAME 1 FOR THE ANIMATIONS TO RECOMPUTE, otherwise you’ll play stale frames. You’ll know your frames are stale cause they’ll change color from blue to orange.



  NOTE: Keyframe animations and animations from dynamics don’t seem to be mutually exclusive? You can have some geo keyframed and others animated dynamically.

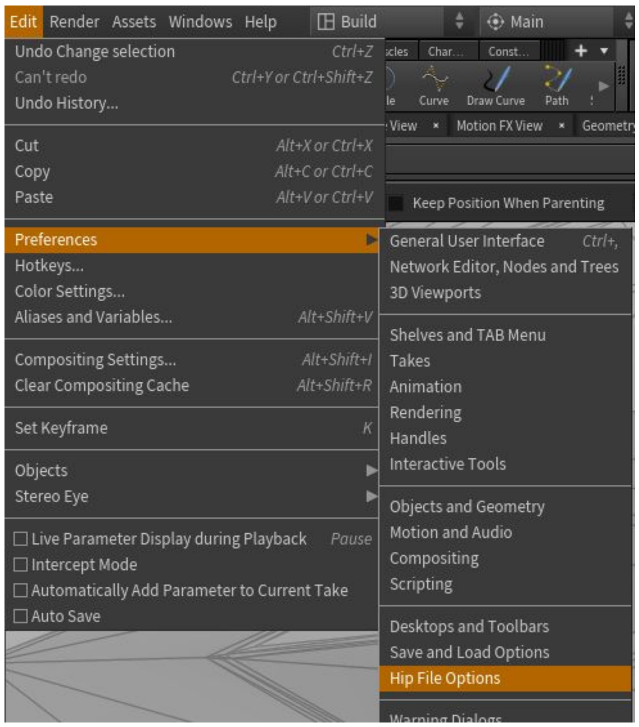
NOTE: Remember to enable the realtime playback button. Otherwise Houdini will playback as fast as possible…

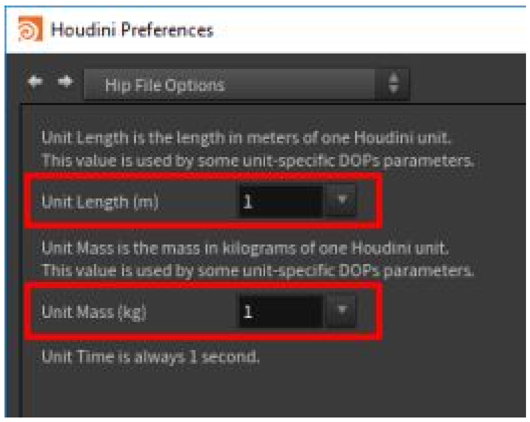


Scene Scale

Scene scale is a super important concept when it comes to Houdini. The size of your objects has a direct effect on the way in which your simulations run. This is exactly like when you were using Box2D on Android.

By default, Houdini is set to treat 1 unit as 1 meter in length / 1 kg in weight. You generally don't want to change these values, but you can do so by going to Edit -> Preferences -> Hip File Options…





Unit Length is the length in meters of one Houdini unit.
This value is used by some unit-specific DOPs parameters.

Unit Length (m) 1

Unit Mass is the mass in kilograms of one Houdini unit.
This value is used by some unit-specific DOPs parameters.

Unit Mass (kg) 1

Unit Time is always 1 second.

NOTE: If you're having problems visualizing this, think of a building collapsing. How long did it take the world trade center building to fall to the ground? Now imagine if the buildings were shrunken down to the height of a smartphone.

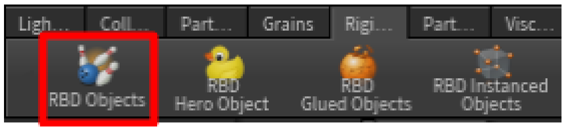
In the new shrunken scale, the building would fall down much faster. The pieces falling to the ground would be much lighter. They wouldn't be in the air as long. They'd have less impact when they hit the ground. They'd have less weight. etc..

Rigid Body Objects

Rigid bodies are physics objects that can move around and collide with stuff. The shape and the volume of the geometry won't change, it'll only be transformed each frame as per the physics calculations.

To create a rigid body object…

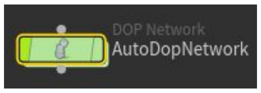
1. Select your object
2. Go to the Rigid Body shelf and select RBD Object

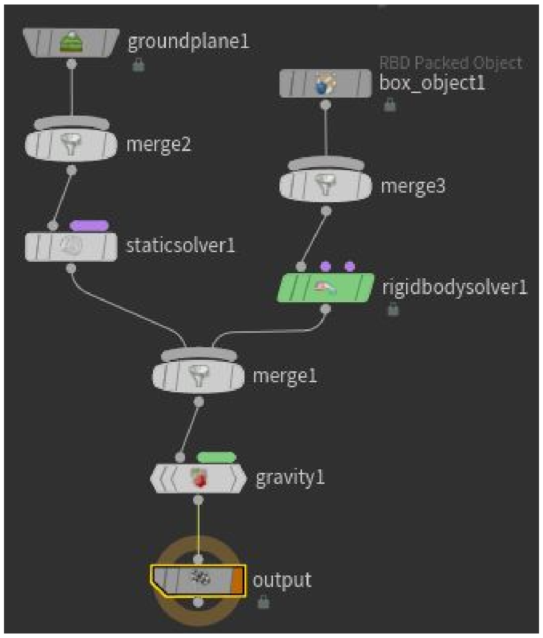


NOTE: Using RBD Objects will create what's called a packed object. Packed objects can't be simulated with finite element objects (objects that go squishy). If you're going to have finite element objects in your scene, use RBD Hero Object. They're less efficient but they work with everything.

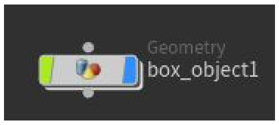
Once you do this, 2 things will happen…

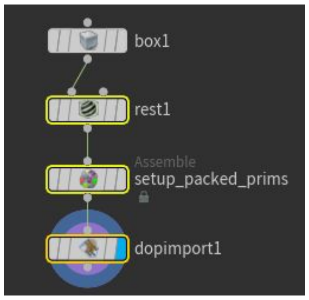
- You'll get a new node called AutoDopNetwork (if it didn't already exist) which contains a node with the same name as the object you've applied RBD to.





- Your object you've applied RBD to will get a few nodes added to it





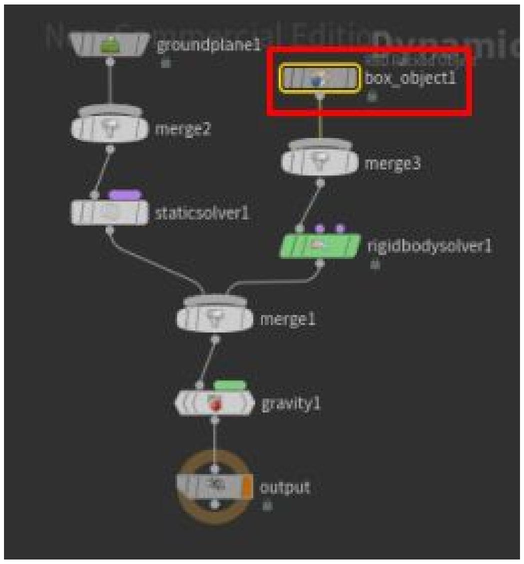
NOTE: The extra nodes that are added into your object, you generally don't want to touch those. They set things up for use in dynamics and create links between your geometry and dynamics.

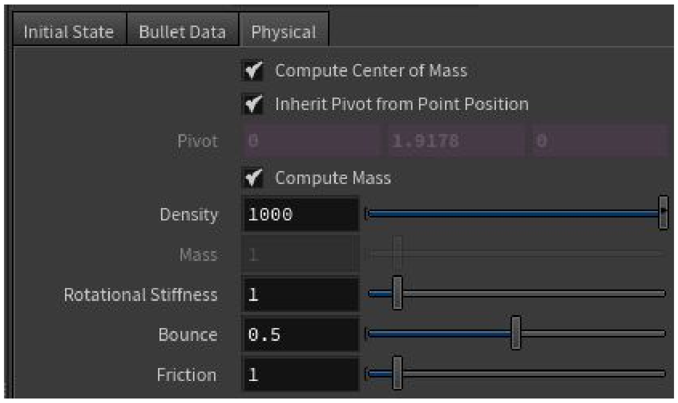
The AutoDopNetwork node is what you traverse into if you want to change the physics properties of your node (e.g. bounciness/friction/etc..)....

Physical Properties

You can change the physical properties for the simulation of your object by…

1. Traversing into the AutoDopNetwork.
2. Selecting the RBD object for that object.
3. In the properties pane, going to the Physical tab.





✓ Compute Center of Mass
✓ Inherit Pivot from Point Position
Pivot 0 1.9178 0
✓ Compute Mass
Density 1000
Mass 1
Rotational Stiffness 1
Bounce 0.5
Friction 1

Bounce	Bounce is how bouncy the object is. Note that the value applied on collision is a mix of this value and the bounciness value from the object being collided with.
Friction	Friction is how much friction the object has to it (e.g. is it like ice or sandpaper?) . Note that the value applied on collision is a mix of this value and the friction value from the object being collided with.

NOTE: Density = volume * mass. If you disable the Compute Mass checkbox, you can specify mass directly.

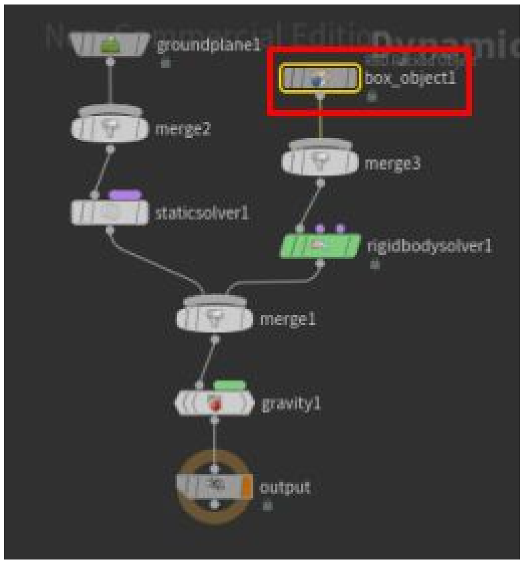
NOTE: No idea what Rotational Stiffness is used for. Best not to change it.

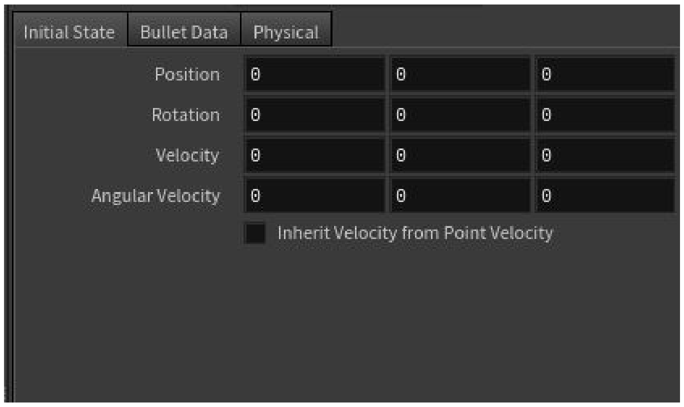
Initial State

Initial state is the physics state of the object at the start of the scene. For example, when the scene starts the object may be throttling down the Z axis.

You can change the physical properties for the simulation of your object by…

1. Traversing into the AutoDopNetwork.
2. Selecting the RBD object for that object.
3. In the properties pane, going to the Physical tab.





Position	0	0	0
Rotation	0	0	0
Velocity	0	0	0
Angular Velocity	0	0	0

☐ Inherit Velocity from Point Velocity

Velocity	The direction this object is moving + the force it's moving at.
Angular Velocity	How fast this object is spinning + the direction it's spinning at.

NOTE: You almost always never want to touch Position and Rotation. These are available for artists to make small tweaks.

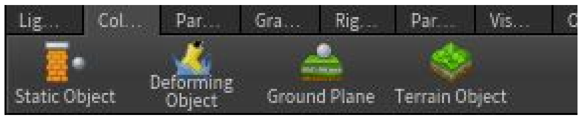
Collisions

Go to the Solver Engines and Collision section and read it THOROUGHLY.

Once you're done that, read the Optimized Collisions section to find out strategies for faster simulations.

Collision Objects

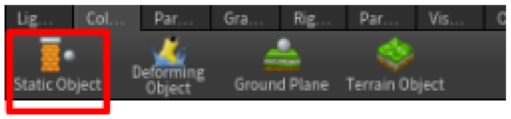
Collision objects are physics objects that can't move around but are used for collisions with other objects. Nothing will change about the geometry, not even the transform.



There are many different types of collision objects you can create…

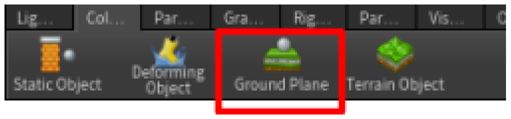
- Static Object → turns your geometry into a collision object

  To create...
  1. Select your geometry object
  2. Got to the Collisions shelf and select Static Object



- Ground Plane → an infinite ground plane not linked to any geometry object.

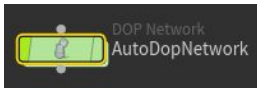
  To create...
  1. Got to the Collisions shelf and select Ground Plane

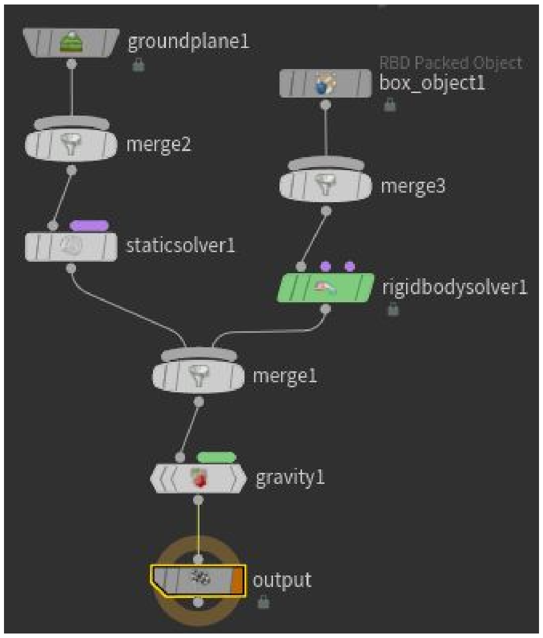


NOTE: What about the other 2?
- No idea what Deforming Object is used for.
- Terrain Object is like a static object but meant specifically for cases where the object is uber thin (e.g. a deformed grid that's meant to represent a terrain) -- apparently this type of static object needs volume applied to it or something so it works better with collisions/solvers/whatever.

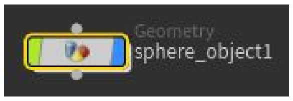
Once you do this, 2 things will happen…

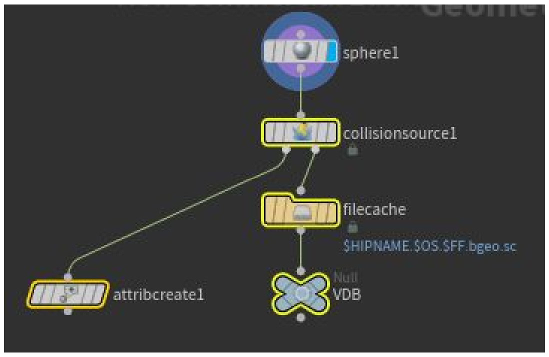
- You'll get a new node called AutoDopNetwork (if it didn't already exist) which contains a node with the same name as the object you've applied collision to.





- Your object you've applied collision to will get a few nodes added to it OR your collision object will generate some new geometry.





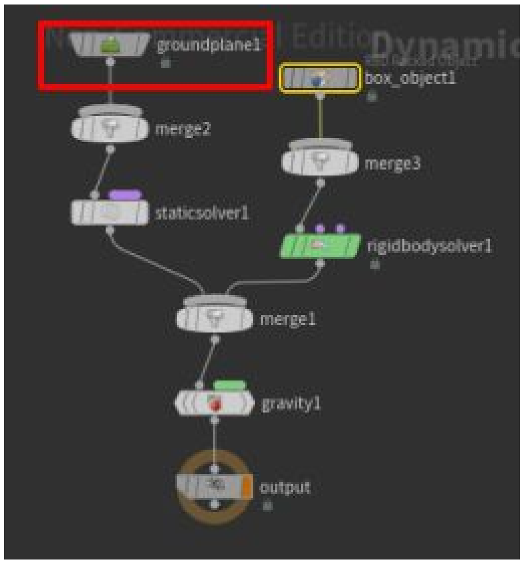
NOTE: The extra nodes that are added into your object, you generally don't want to touch those. They set things up for use in dynamics and create links between your geometry and dynamics.

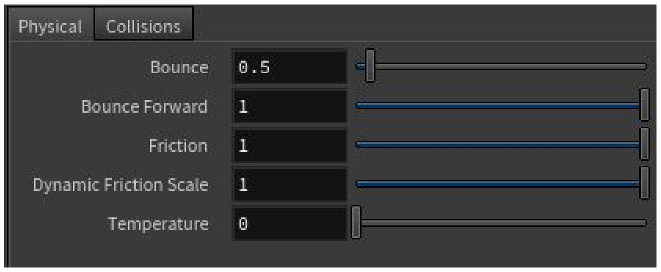
The AutoDopNetwork node is what you traverse into if you want to change the physics properties of your node (e.g. bounciness/friction/etc..)....

Physical Properties

You can change the physical properties for the simulation of your object by…

1. Traversing into the AutoDopNetwork.
2. Selecting the collision object for that object.
3. In the properties pane, going to the Physical tab.





Bounce 0.5
Bounce Forward 1
Friction 1
Dynamic Friction Scale 1
Temperature 0

Bounce	Bounce is how bouncy the object is. Note that the value applied on collision is a mix of this value and the bounciness value from the object being collided with.
Friction	Friction is how much friction the object has to it (e.g. is it like ice or sandpaper?) . Note that the value applied on collision is a mix of this value and the friction value from the object being collided with.

NOTE: No idea what the rest of the properties are used for. They have descriptions but I don’t really understand what they mean.

Collisions

Go to the Solver Engines and Collision section and read it THOROUGHLY.

Once you’re done that, read the Optimized Collisions section to find out strategies for faster simulations.

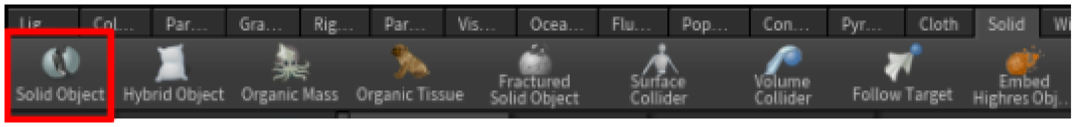
Finite Element Objects

Finite element objects are objects that can move around and collide with stuff (just like rigid body objects). But, the shape and the volume of the geometry can also change. So, in addition to transforming based on physics calculations, the physics calculations will also deform the geometry.

> **NOTE**: Cloth sims use to happen via finite element solver in Houdini 15, but in Houdini 16 it's been updated to use its own solver (called cloth solver).

To create a finite element object…

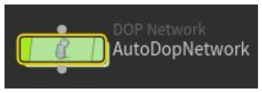
1. Select your object
2. Go to the Solid shelf and select Solid Object

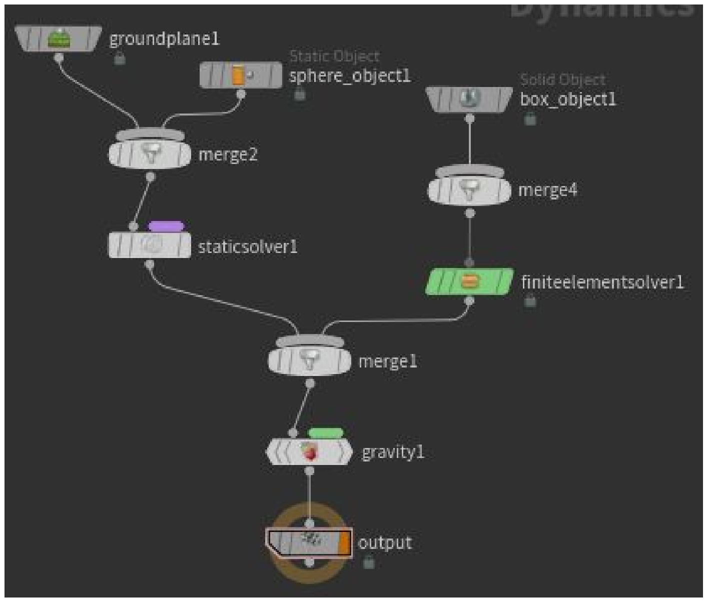


> **NOTE**: It seems that Organic Mass/Tissue will generate the same thing, just with different physics properties (e.g. more stiff / less stiff)?

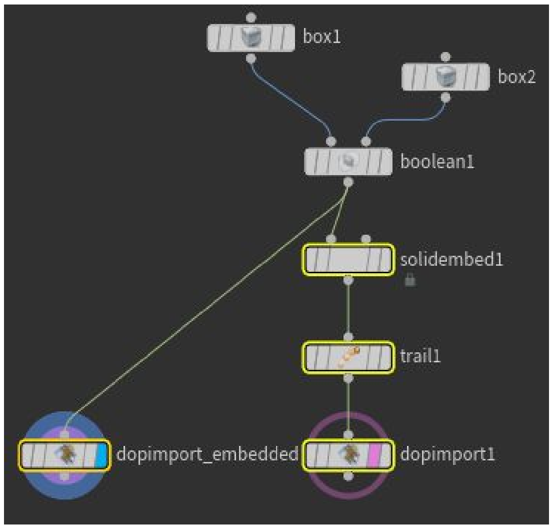
Once you do this, 2 things will happen…

- You'll get a new node called AutoDopNetwork (if it didn't already exist) which contains a node with the same name as the object you've applied RBD to.





- Your geometry object will get a few nodes added to it



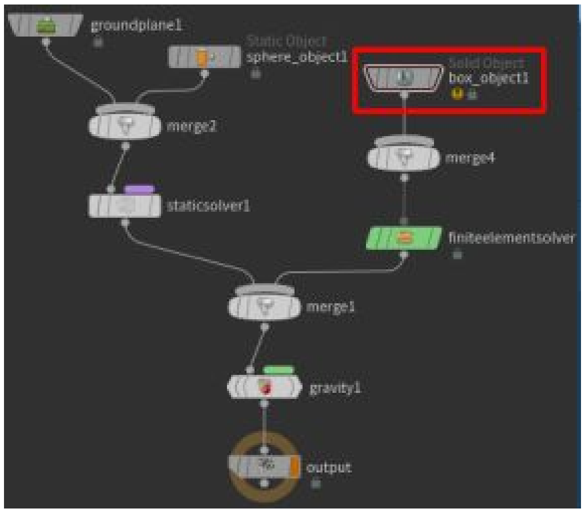
NOTE: The extra nodes that are added into your object, you generally don't want to touch those. They set things up for use in dynamics and create links between your geometry and dynamics.

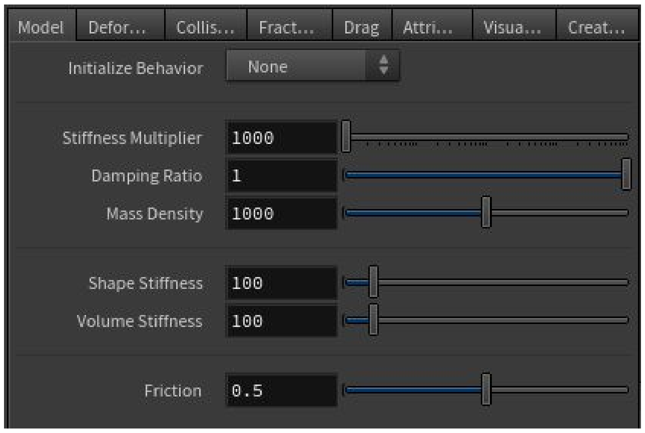
The AutoDopNetwork node is what you traverse into if you want to change the physics properties of your node....

Physical Properties

You can change the physical properties for the simulation of your object by…

1. Traversing into the AutoDopNetwork.
2. Selecting the solid object for that object.
3. In the properties pane, going to the Model tab.





Initialize Behavior: None
Stiffness Multiplier: 1000
Damping Ratio: 1
Mass Density: 1000
Shape Stiffness: 100
Volume Stiffness: 100
Friction: 0.5

NOTE: It looks like these values were really finicky with older versions of Houdini. In certain cases they would cause the deforming geometry to explode. I couldn't get a similar effect to show up in Houdini 16. Houdini 16 seems to have solved that issue?

I'm unsure what all of these settings do. The 'Initialize Behaviour' dropdown up top will change the parameters to represent common materials (e.g. rubber).

Damping Ratio	Officially this is described as "how quickly the object stops deforming". You can also think of this as the springiness of your object. For example, imagine that you're dealing with a cloth-like piece of geometry that falls/drapes over a sphere. As the cloth drapes, the ends of the cloth may spring back-and-forth heavily. If you jack up the damping ratio, this will happen less? **NOTE**: The lower this value is, the more jitter you have to deal with when your object is at rest.
Mass Density	Officially described as "the amount of mass per volume". I assume this means the amount of mass per 1 unit squared of volume. **NOTE**: Remember that distance defaults to 1 = 1 meters, so 1 unit of volume is 1 meter squared.
Shape Stiffness	Officially described as how much your object resists changes to its shape. I describe this as how 'melty' your object is. The lower it is, the more your object melts.

	Remember that this value is dependent on the mass density value.
Volume Stiffness	Officially described as how much your object resists changes to its volume. Realistically, modifying this value doesn’t seem to do much other than maybe make the object less drape-y as it melts (this is with super high values). It’s like as the object deforms and goes flat (this happens with a very low SHAPE stiffness or very high mass density), the volume stiffness seems to prevent it from draping over the object. It’s almost like throwing a heavy rubber mat over an object.
Friction	Friction is how much friction the object has to it (e.g. is it like ice or sandpaper?) . Note that the value applied on collision is a mix of this value and the friction value from the object being collided with.

Accuracy

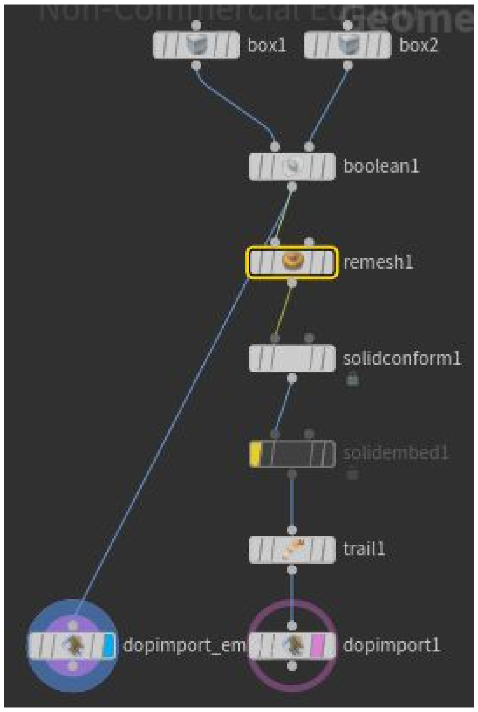
You may notice a small amount of bleedthru with your finite element object when it collides, especially if one of your objects goes really loose and drapes over the other (low surface stiffness).

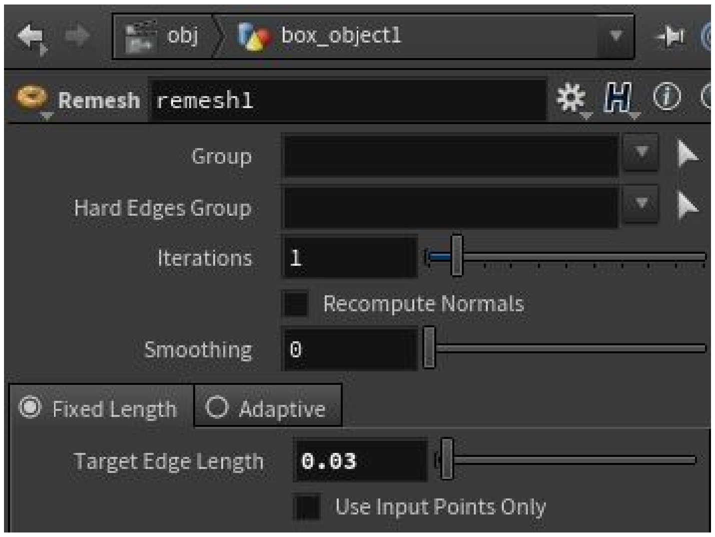
Increasing the poly count of the objects doesn’t seem to do anything. What’s going on is that the nodes added to the geometry object are automatically creating tetrahedrons based on what the internal algorithms thinks are efficient for the object being passed into the AutoDopNetwork.

Increasing the polys won’t increase the number of tets. But, we can MANUALLY increase the number of tets being created for the object by adding a few nodes into the chain. The more tets we have, the more we can stop bleeding (but the sim will also be a lot slower).

To increase the number of tets…

1. Go to your object node
2. Delete the Solid Embed node (or bypass it)
3. Add a Remesh node and a Solid Conform node in its place
4. Go to the Remesh node and change the Target Edge Length property





Remesh remesh1

Group

Hard Edges Group

Iterations 1

Recompute Normals

Smoothing 0

Fixed Length [✓] Adaptive

Target Edge Length 0.03

Use Input Points Only

The lower the Target Edge Length, the more tets will be generated.

> **NOTE**: Feel like doing this to a lot of objects? Shove the nodes inside a subnetwork and make a houdini digital asset out of it. Check out the section on subnetworks in the main document.

Collisions

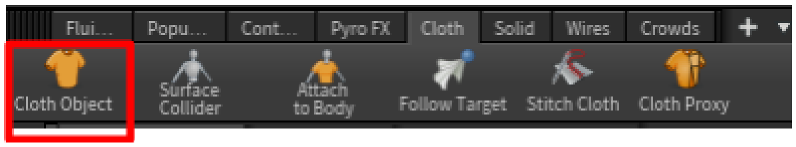
Go to the Solver Engines and Collision section and read it THOROUGHLY. Finite Element objects use RDB, so the objects you're colliding with should have good collision geometry setup for RDB.

Also, the number of test in your model can be boosted (see the Accuracy subsection above) to help if certain parts (tight areas and hard edges?) of your finite element object is being clipped through.

Cloth Objects

As of Houdini 16, cloth no longer uses the finite element solver for simulations. Cloth objects have their own cloth solver that they use. I'm unsure how collisions are done here (Bullet solver? RBD solver? other?).

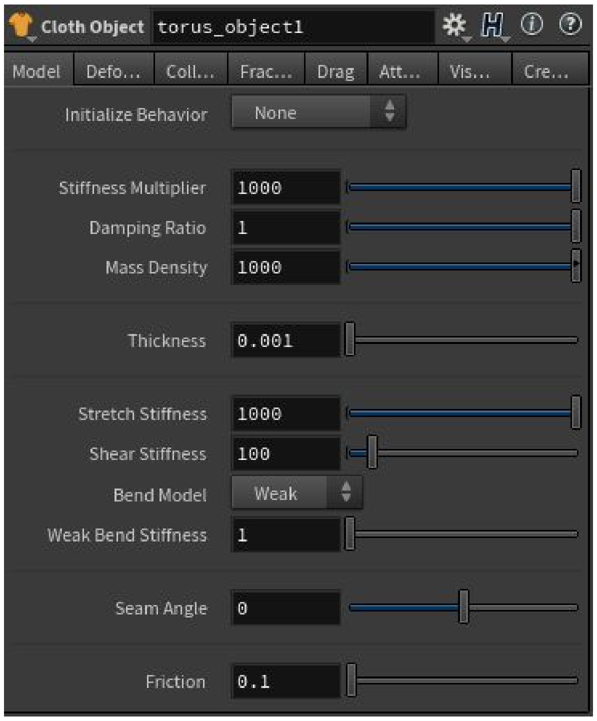
You can turn an object into "Cloth" by going to the Cloth shelf and selecting Cloth Object…



NOTE: Unlike finite element objects, this uses polygons directly. IT DOES NOT CONVERT TO TETRAHEDRONS. If you want more accurate simulations, just up your polycount.

Physical Properties

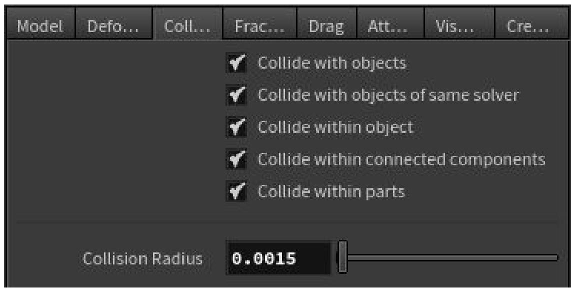
If you go to the AutoDopNetwork and look at the properties of whatever it was you cloth'd up, you'll see properties similar to those for finite element objects…



Cloth Object torus_object1

Model | Defo… | Coll… | Frac… | Drag | Att… | Vis… | Cre…

Initialize Behavior	None
Stiffness Multiplier	1000
Damping Ratio	1
Mass Density	1000
Thickness	0.001
Stretch Stiffness	1000
Shear Stiffness	100
Bend Model	Weak
Weak Bend Stiffness	1
Seam Angle	0
Friction	0.1



Model | Defo… | Coll… | Frac… | Drag | Att… | Vis… | Cre…

✓ Collide with objects
✓ Collide with objects of same solver
✓ Collide within object
✓ Collide within connected components
✓ Collide within parts

Collision Radius 0.0015

The Initialize Behavior dropdown lets you set these parameters to specific types of cloth (e.g. silk). Also, if you notice any bleed-through happening, you can try fiddling with the Thickness option or go under the Collision tab and fiddle with the Collision Radius option.

Collisions

Go to the Solver Engines and Collision section and read it THOROUGHLY. Finite Element objects use RDB (I think), so the objects you're colliding with should have good collision geometry setup for RDB.

Solver Engines and Collision

There are 3 different solver engines that dynamics can use to calculate how things move and collide:

- Bullet ← open-source physics engine
- RBD ← Houdini's internally built physics engine
- ODE ← open-source physics engine that only deals with shape primitives (e.g. sphere)

<u>Which one gets used is dependent on the solver settings / what's being interacted with</u>. Ultimately, if you're going to be doing anything serious with Houdini, you'll need appropriate collision geometry being generated for both RBD and Bullet.

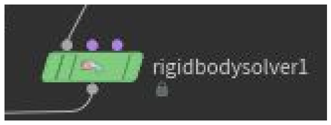
Just remember these two points…

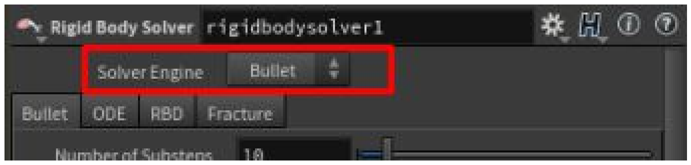
1. Bullet is used for rigid body interactions with other rigid bodies and collision objects
2. RBD is used for everything else (fluids, particles, volumes, cloth, finite element, etc..)

Collision Geometry for Bullet

By default, the rigidbody solver uses the Bullet open-source physics engine to do physics calculations. Bullet seems to generate collision geometry that's more efficient for its calculations vs your real geometry.

> **NOTE**: You can change which engine the solver uses by going to the rigid body solver node and choosing something other than Bullet for the solver engine. If change it to RDB, none of this collision geometry stuff in here will apply to objects being fed into it -- instead see the RDB subsection below.

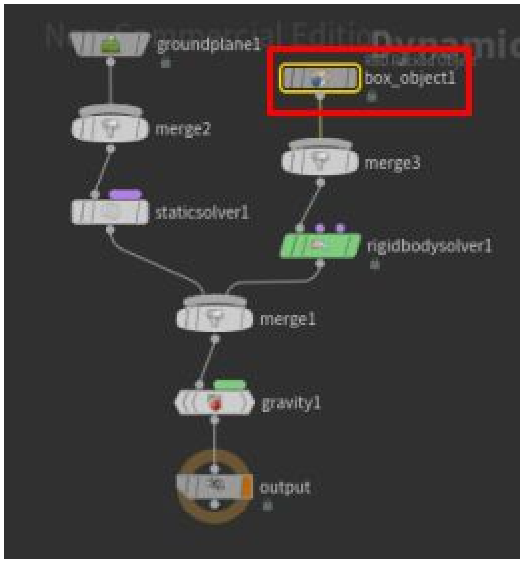


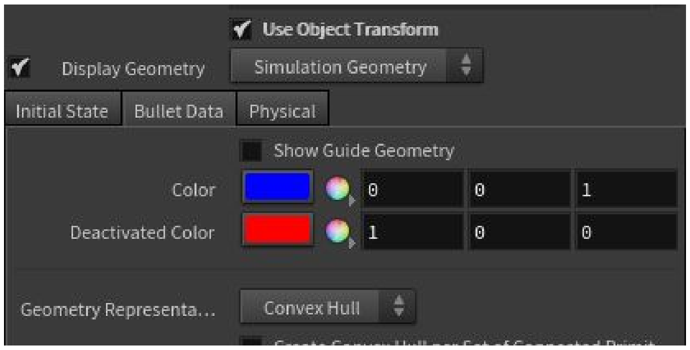

Rigid Body Solver rigidbodysolver1

Solver Engine: Bullet

You can get to the collision geometry used for the simulation of your object by…

1. Traversing into the AutoDopNetwork.
2. Selecting the RBD object (or Static object) for that object.
3. In the properties pane, going to the Bullet Data tab.





✓ Use Object Transform

✓ Display Geometry: Simulation Geometry

Initial State | Bullet Data | Physical

Show Guide Geometry

Color: 0 0 1

Deactivated Color: 1 0 0

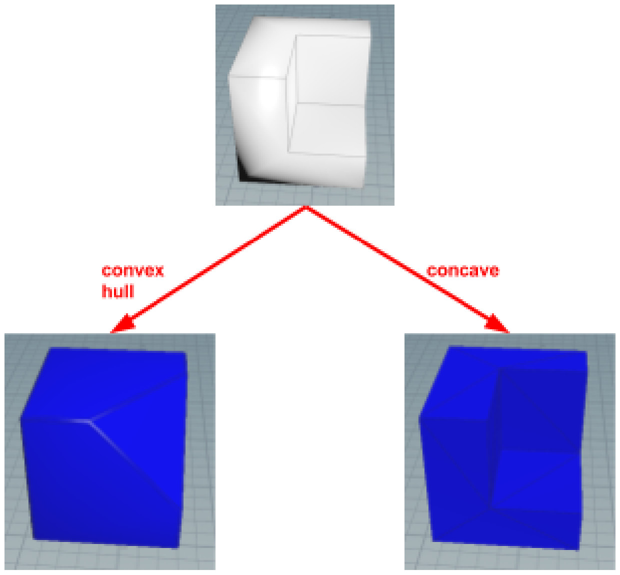
Geometry Representa...: Convex Hull

To show the collision geometry being generated for your object…

1. Uncheck Display Geometry (so only your collision geo will show)
2. Check Show Guide Geometry

   NOTE: If you play your dynamics animation while guide geo is showing, the Color and Deactivated Color will be used to show if Bullet is using your object for computations. Objects that are at rest will be deactivated until they're interacted with again.

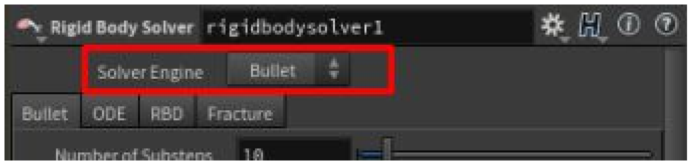
To change your collision geometry…
Use the Geometry Representation dropdown. By default Convex Hull is used, which is a highly reduced version of your geo that doesn't cave in at all. If this isn't good enough, you can choose Concave.



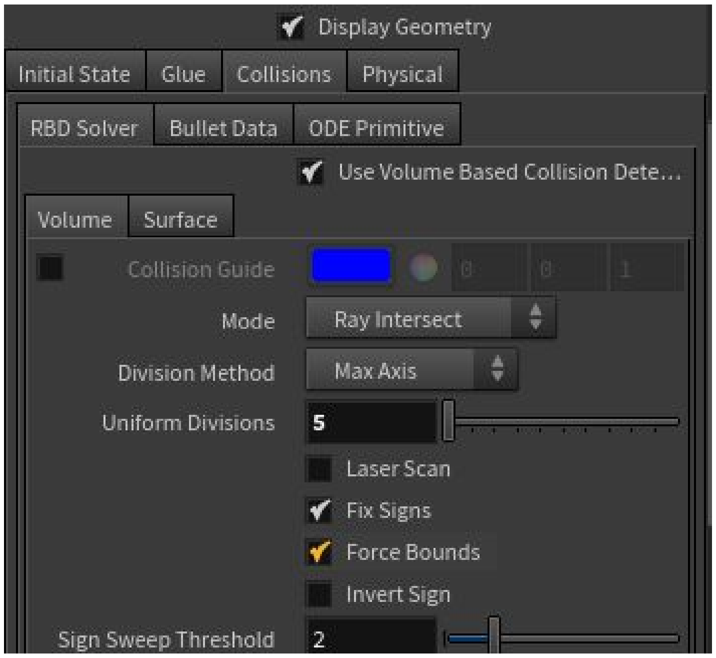
Collision Geometry for RBD

For anything that isn't a rigidbody-to-rigidbody or rigidbody-to-static interaction, the RBD engine is used. So for example, if your rigid body collides with a finite element object/particles/flip fluids/volumes/etc.., RBD is used.

> **NOTE**: If your rigidbody solver is set to use RBD via its Solver Engine property, then your rigidbody-to-rigidbody and rigidbody-to-static interactions will also use RBD (it defaults to Bullet).


Rigid Body Solver: rigidbodysolver1
Solver Engine: Bullet

The RBD Solver tab of your static/rigidbody object will have 2 tabs: Volume and Surface...


✓ Display Geometry

✓ Use Volume Based Collision Dete...

Collision Guide: 0 0 1
Mode: Ray Intersect
Division Method: Max Axis
Uniform Divisions: 5
Laser Scan
✓ Fix Signs
✓ Force Bounds
Invert Sign
Sign Sweep Threshold: 2

> **NOTE**: Remember to uncheck Display Geometry if you're going to show the Collision Guide

By default, finite element objects use "Volume Based Collision". I think this tries to approximate a simple volume based on the geometry.

According to the lesson, this is fast but it may not be accurate. You can turn up the uniform division count to make it more accurate, but if your object has a lot of large holes it still won't be all that accurate. Furthermore it seems that in certain cases colliding geometry will just slowly bleed through the object instead of coming to rest on collision (e.g. fluids or cloth).

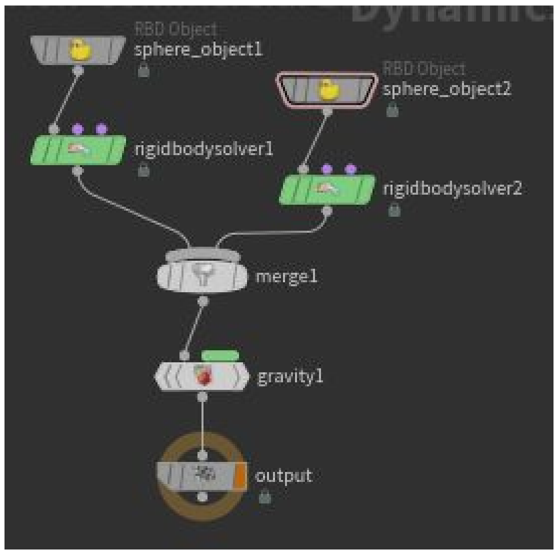
If you want more accurate collision geometry…

1. Uncheck Use “Volume Based Collision Detection”
2. Switch to the Surface tab
   NOTE: Is there anything worth adjusting in the Surface tabs? You can switch from Points to Edges? The video says depending on the number of points you have vs the number of edges, one may be more efficient than the other? But wouldn’t this be an easy thing to automatically calculate? Why are they making you manually pick it? I think in some cases I saw that you could set a collision radius for surface collisions that’d give the collision surface some extra padding.

Or, you can try generating simpler collision geometry and use that for the volume based collision instead. Simpler geometry means better volumes are generated. See the section on Optimized Collisions for more information.

Solver Gotchas

As far as I know, rigidbodysolver is the only solver node that lets you pick the physics engine to use for your simulations. So what happens when we have 2 rigidbodysolver nodes being merged, one set to use Bullet and the other set to use RBD?



The behaviour slightly varies between the the various options…

- Bullet vs RBD
- RBD vs Bullet
- Bullet vs Bullet
- RBD vs RBD

Ultimately I'm not 100% sure what's going on here. From experimenting it looks like it uses the engine from the LAST input to the merge (remember that you can reorganize inputs into the a merge by going to the properties).

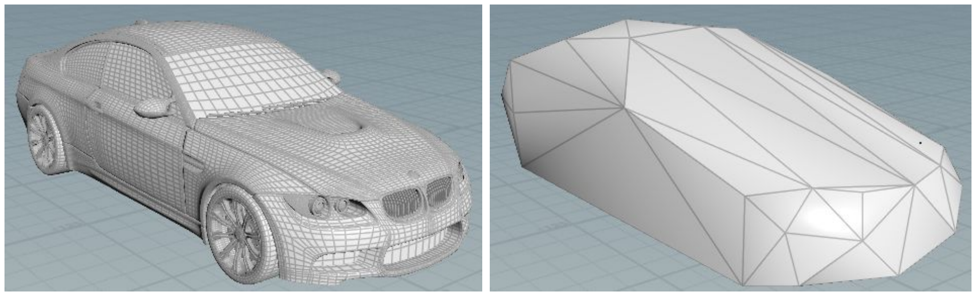
> **NOTE**: I was able to come to this conclusion by having 2 same sized spheres but jacking up the Bullet collision padding property on the second sphere. If it used the padded collision geometry for the collision (instead of sinking into it until it hit the actual sphere), it was using Bullet. No gravity was involved in the test.
>
> **NOTE**: Each one of the Bullet/RBD combos above seemed to act a little differently? No idea what's going on here.

Optimized Collision Geometry

If you're using highpoly models in your scene, you may not necessarily want to use those same models for your simulations. Your simulations may be able to get away with using much lower resolution models.

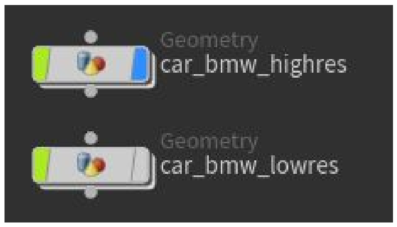
Why would you want to do this? It'll boost the speed of your simulations by a significant amount if you use lower resolution geometry.

For example, imagine we had 2 models given to us by our modeller: a high-res model for rendering and a low-res model for our rigidbody simulations…



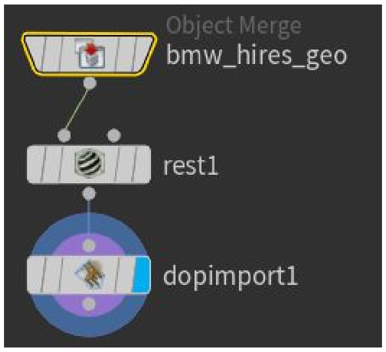
How do we do this?

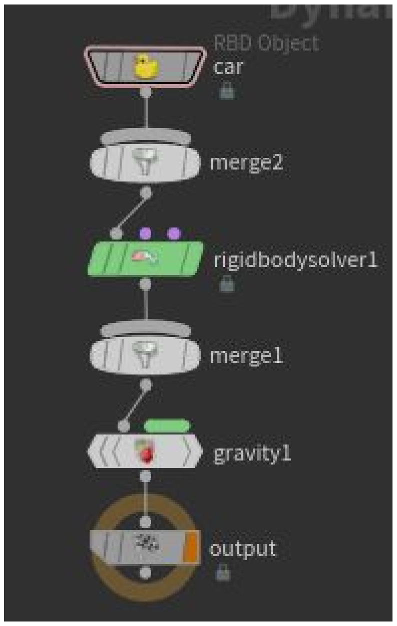
1. Import both models into your scene…



2. Convert the high-res version intro a rigid-body/static object...

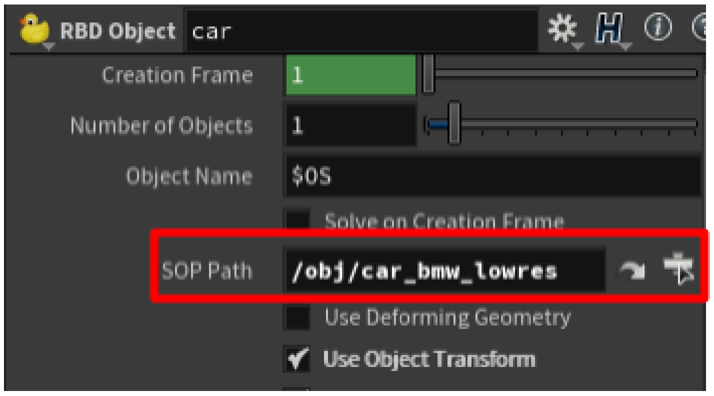
   You'll end up getting the typical nodes in your Geometry node and your AutoDopNetwork





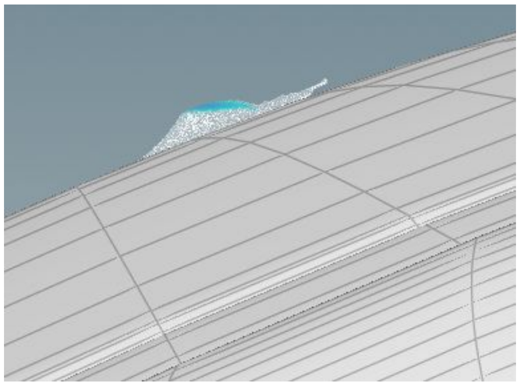
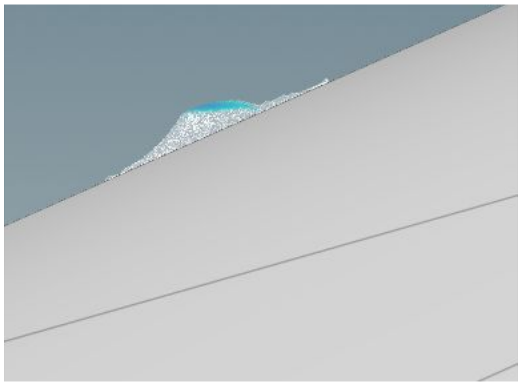
3. Point the RBD Object created in the AutoDopNetwork to the lowres version..

   You can do this by changing the SOP Path property.


RBD Object car
Creation Frame 1
Number of Objects 1
Object Name $OS
Solve on Creation Frame
SOP Path /obj/car_bmw_lowres
Use Deforming Geometry
✓ Use Object Transform

   NOTE: You can use the little button on the rightmost next to the text field to get a outliner window to quickly pick the geo...

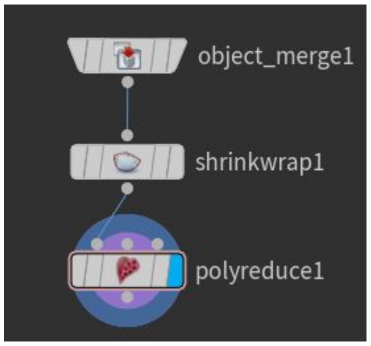
Once you do this, you'll see the low-res version if you go into your AutoDopNetwork, but the high-res version when you come out…

 ..vs.. 

Notice how the water isn't really colliding exactly with the highres geometry, but it is with the lowres geometry. A lot of times this is perfectly fine for a render (depends on the angle and various other things), and if you wanted a more accurate collision you can ask for some more polygons in your lowres version.

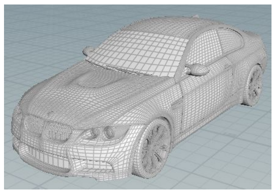
Shrinkwrap Method

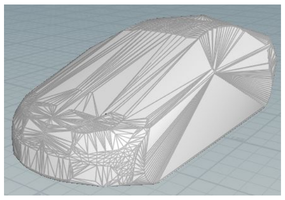
If we don't have a simpler geometry given to us by our modeler, we can use Houdini directly to generate some…

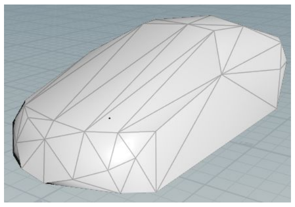


So what's going on here? Here's what each node does…

1. Object Merge → pulls in the high-res geometry



2. Shrinkwrap → creates a convex hull polygon mesh out of the geometry


3. Polyreduce → reduces the poly count on the convex hull


   You'll need to go into the properties for this node and manually turn down the Keep % number to something you think is appropriate

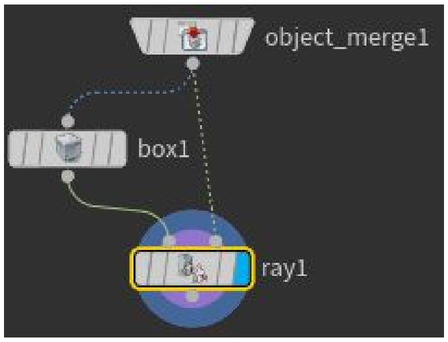
Just as you would have done if lowres geometry was provided to you, point your RBD Object in your DOP network to this new low-res geometry node (or, if you did this in the same geometry node, point directly to the polyreduce node).

This produces geometry very similar to Bullet's default Convex Hull collision geometry. Why would you want to use this instead? If you don't care about the holes in your object for your collision (for example, my car's rims have holes but I'm not doing any specific collision with those in my scene), this will generate a much better RBD collision volume and finish the simulation much faster.

> **NOTE**: Feel like doing this to a lot of objects? Shove the nodes inside a subnetwork and make a houdini digital asset out of it. Check out the section on subnetworks in the main document.

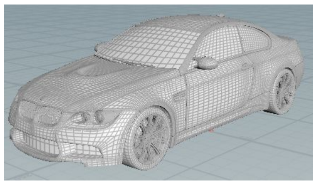
Ray Method

If we don't have a simpler geometry given to us by our modeler, we can use Houdini to directly generate some…

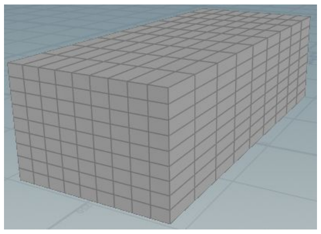


So what's going on here? Here's what each node does…

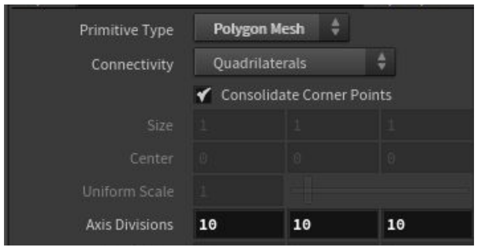
1. Object Merge → pulls in the high-res geometry

2. Box → generates a new polygon box

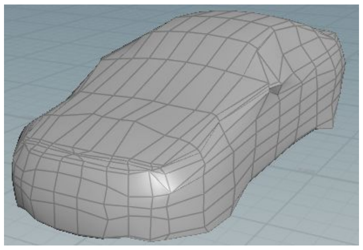
   

   By feeding in our original geometry, the box's size will match the bounding box of the geometry. You'll need to go into the properties for this node and make sure the Primitive Type is set to Polygon Mesh and the Axis Divisions will give you the number of faces you want for your low-res geo…

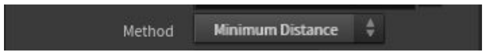
   

   Primitive Type: Polygon Mesh
   Connectivity: Quadrilaterals
   ✓ Consolidate Corner Points
   Size: 1 1 1
   Center: 0 0 0
   Uniform Scale: 1
   Axis Divisions: 10 10 10

3. Ray → projects rays from first input to the second input (in the direction of its normal) and moves the geometry to where the rays hit



You'll need to go into the properties for this node and manually set the Method to Minimum Distance...



Method Minimum Distance

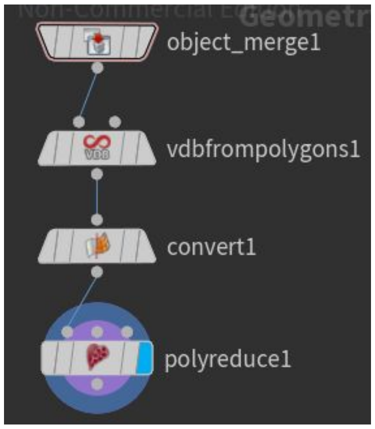
Just as you would have done if lowres geometry was provided to you, point your RBD Object in your DOP network to this new low-res geometry node (or, if you did this in the same geometry node, point directly to the polyreduce node).

This produces geometry that isn't a Convex Hull, so you can use it for those cases where you need a bit more detail for your collisions (e.g. it's important for the colliding object to hit the bottom of the car exactly). Much like the shrinkwrap method, this will generate a much better RBD collision volume and finish the simulation much faster.

> **NOTE**: You may want to add in a polyreduce node at the end of this chain, just because it seems edges seem to accumulate more of the faces than are needed.

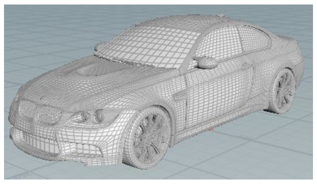
VDB Convert Method

If we don't have a simpler geometry given to us by our modeler, we can use Houdini directly to generate some…

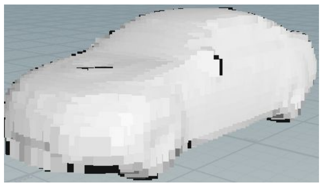


So what's going on here? Here's what each node does…

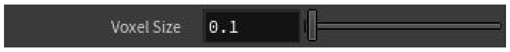
1. Object Merge → pulls in the high-res geometry



2. VDB from Poly → converts your polygon model into voxels

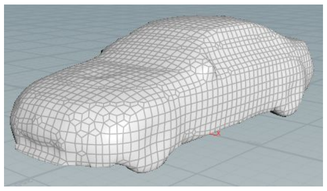


If you have a very large model, this may take a long time to cook during your first run. You'll want to jack up your voxel size in the node's properties so that you can reduce detail…

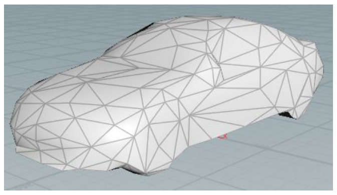


Voxel Size 0.1

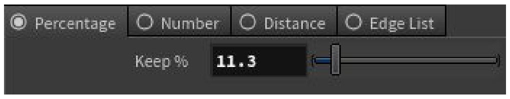
3. Convert → converts your voxels back to polygons



4. Polyreduce → reduces the polycount



You'll need to go into the properties for this node and manually turn down the Keep % number to something you think is appropriate…

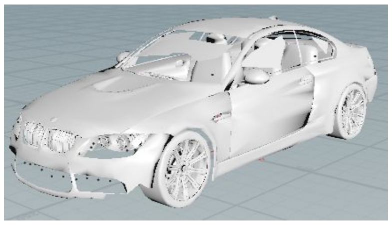


[✓ Percentage] Number Distance Edge List

Keep % 11.3

Just as you would have done if lowres geometry was provided to you, point your RBD Object in your DOP network to this new low-res geometry node (or, if you did this in the same geometry node, point directly to the polyreduce node).

This produces geometry that isn't a Convex Hull, so you can use it for those cases where you need a bit more detail for your collisions (e.g. it's important for the colliding object to hit the bottom of the car exactly). Much like the shrinkwrap method, this will generate a much better RBD collision volume and finish the simulation much faster.

> **NOTE**: What the fuck is the point of this? Why don't I just use a polyreduce node directly? Apparently it causes gaps??? Here's what happened when I fed the high-res geometry directly into a polyreduce node and jacked the polycount down…
>
> 
>
> It looks like it just randomly kills faces?

> **NOTE**: Feel like doing this to a lot of objects? Shove the nodes inside a subnetwork and make a houdini digital asset out of it. Check out the section on subnetworks in the main document.