POPs with and in DOPs

by Deborah R Fowler*

Overview

*Idea taken from www.sidefx.com refer to sidefx under learning/ tutorials/dynamics Houdini 9 dynamics

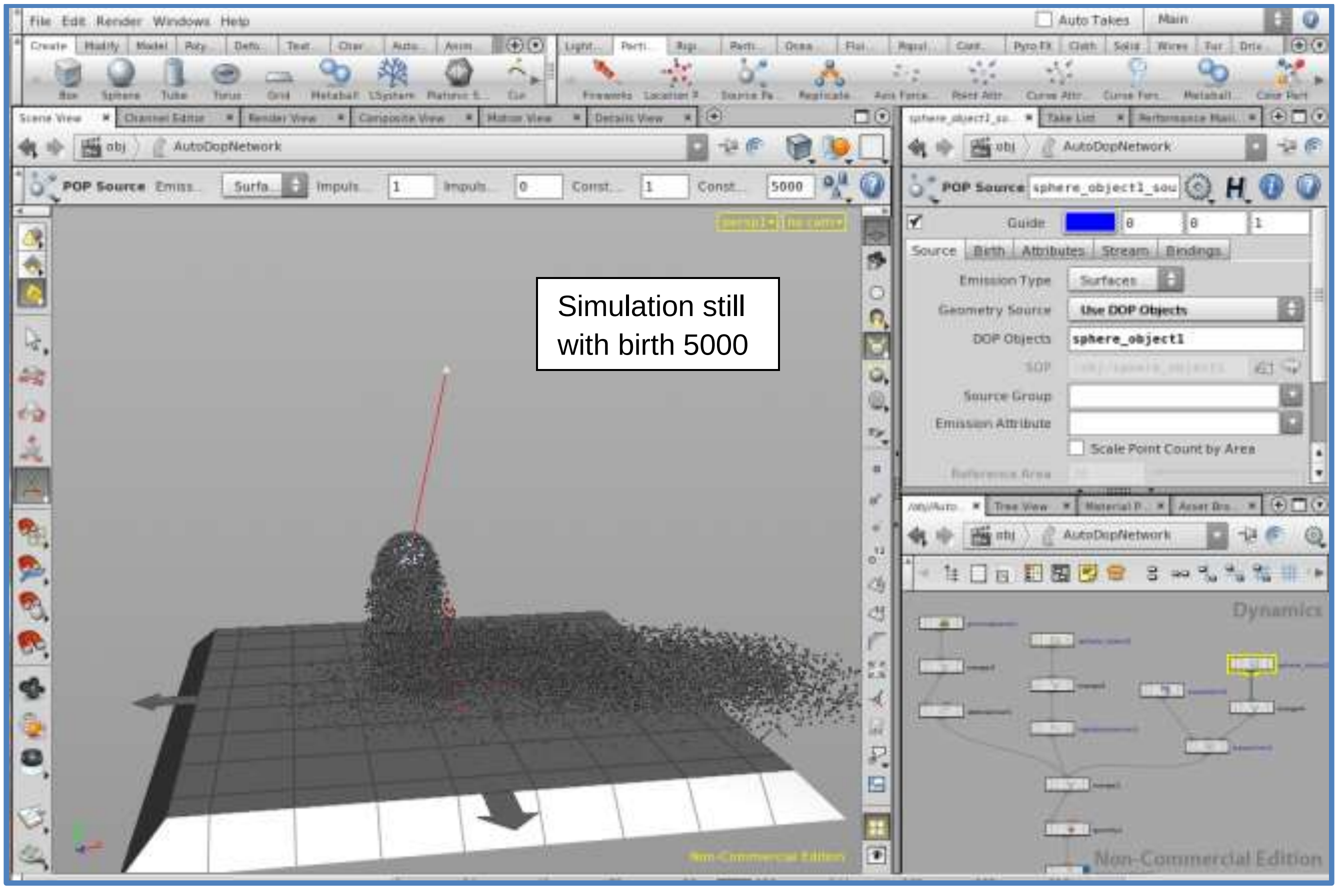
Modified now to be used with H13 Particle Dynamics.

This is a guide to step through the example given in class and is intentionally written in point form style.

Tutorial on Pin Constrained sphere steps: PENDULUM

Create a RBD pendulum, pin constrained sphere

1. Use shelf tools to create a sphere, make it a rigid body object
   - This gives us the AutoDopNetwork – when we look at this we see the RBD object, an RBD solver (required for dynamics) and it has automatically added gravity (the merges are for input later on). Inside the solver we see gravity set to -9.80665 which is earth normal gravity.
   - This network is generated for you automatically.
2. Create a pin constraint*, with geometry selected*, click on RBD pin constraint
   - Gives us a RBD pin constraint and a handle for the constraint
   - Click on the display for the AutoDopNetwork so that the constraint handle can be seen and move it up (or if you are in the dop network just select the pin constraint node)
   - Reduce frames to about 100 (default is longer) icon at the bottom
   - Solution after first time is cached and plays fast – turn on real time playback
   - NOTE: Although particles are “scrubable” you will still need to go back to the first frame and likely will have to hit “Reset Simulation” on your dop network. For example, changing the birth rate – try it and see.
3. Lift up the pin constrained sphere so that we can put a ground plane in and a particle system (particles systems are now dynamic)
   - Go to create particles, select create particles from source (emission type defaults to Surfaces so even though it is a primitive, it will emit from the surface not the center – this is new to H13)
   - now go to the POP Source node in your dopnet and change the ***Geometry Source*** to **Use DOP Objects** and your ***DOP Objects*** to **sphere_object1**
   - With particles you no longer have to set up the collision separately.



Simulation still with birth 5000

POP Source Emiss... Surfa... Impuls... 1 Impuls... 0 Const... 1 Const... 5000

Emission Type: Surfaces

Geometry Source: Use DOP Objects

DOP Objects: sphere_object1

☐ Scale Point Count by Area

Now we have an RBD object constrained by a pin, emitting particles that collide with a ground plane.

4. Add a second object, a sphere with boxes instanced onto it
   - create a sphere, change it to **Polygon** type, and reduce the **Frequency** to 1
   - create a box
   - Under RBD shelf, select sphere and choose **RBD point object**, select box for object to be copied (select RBD point object in this case – not packed)
   - the box is instantiated to points on the polygon sphere
   - now we can smash boxes on sphere and the boxes also collide with ground plane. The DOP network sets this up so that both go through the RBD solver and the ground plane is a static RBD object, with an RBD static solver
   - Note: typing L tidies up the node network

DOPs handles this well

POPS now works fine with all of these RBD objects.

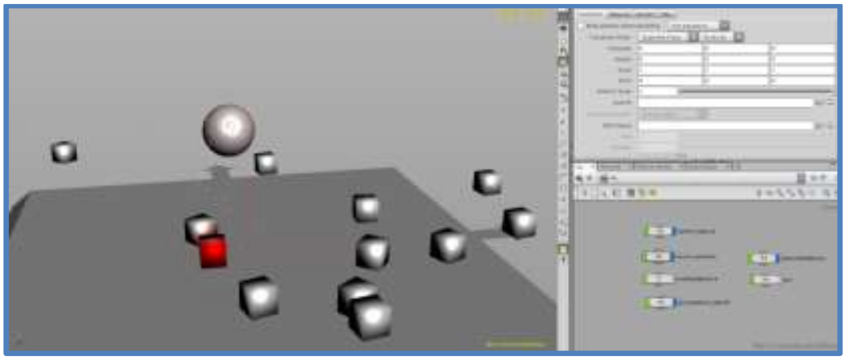
Try changing the behavior of the particles: currently the particles are bouncing. In the POP solver node, go to the Collision Behavior tab. If you check

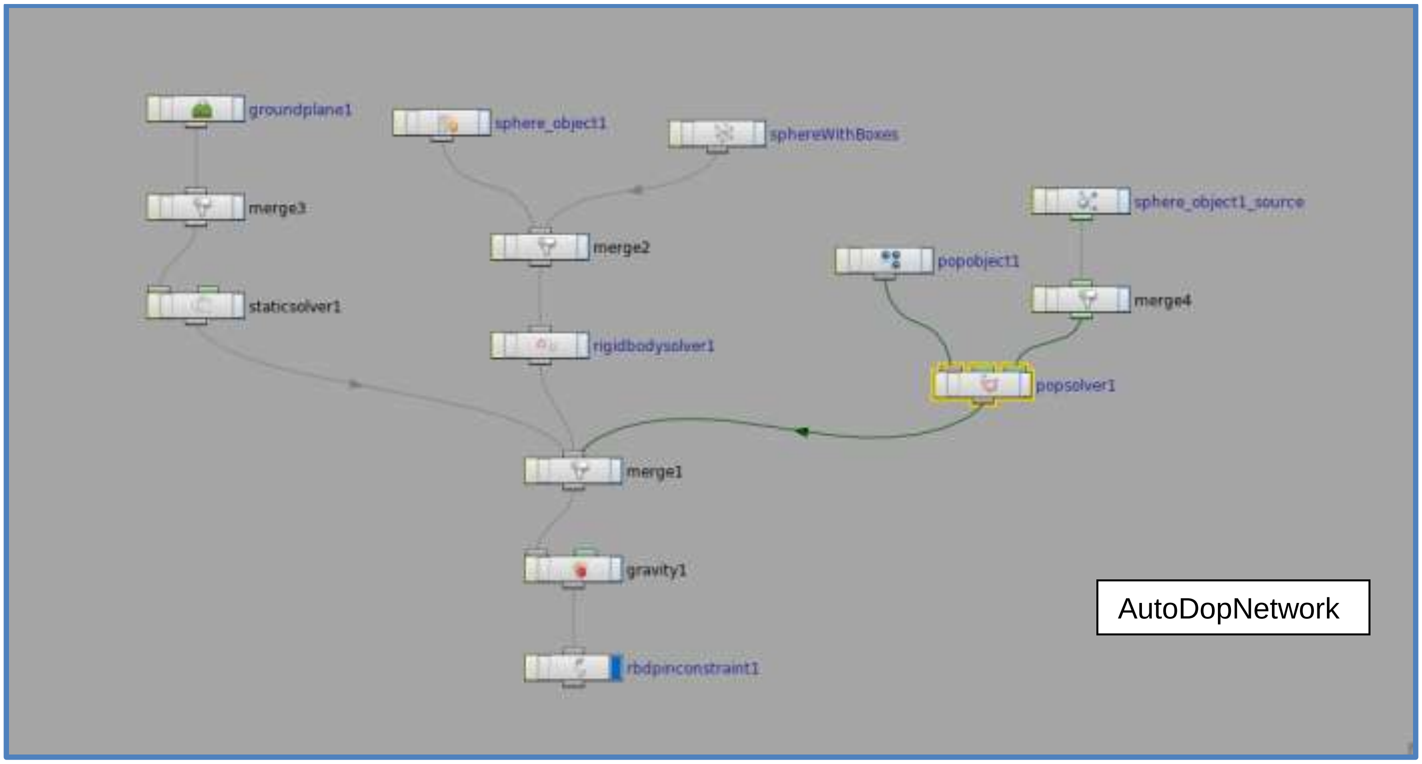
Add Hit Attributes you will see you can select **None, Die, Stick, Stop, Slide**
Also new is the checkbox to color the particles that collide: **Color Hits**.
(No documentation on the Collision Behavior tab as of 11/30/2013)

Have attribute transferred from particles to boxes

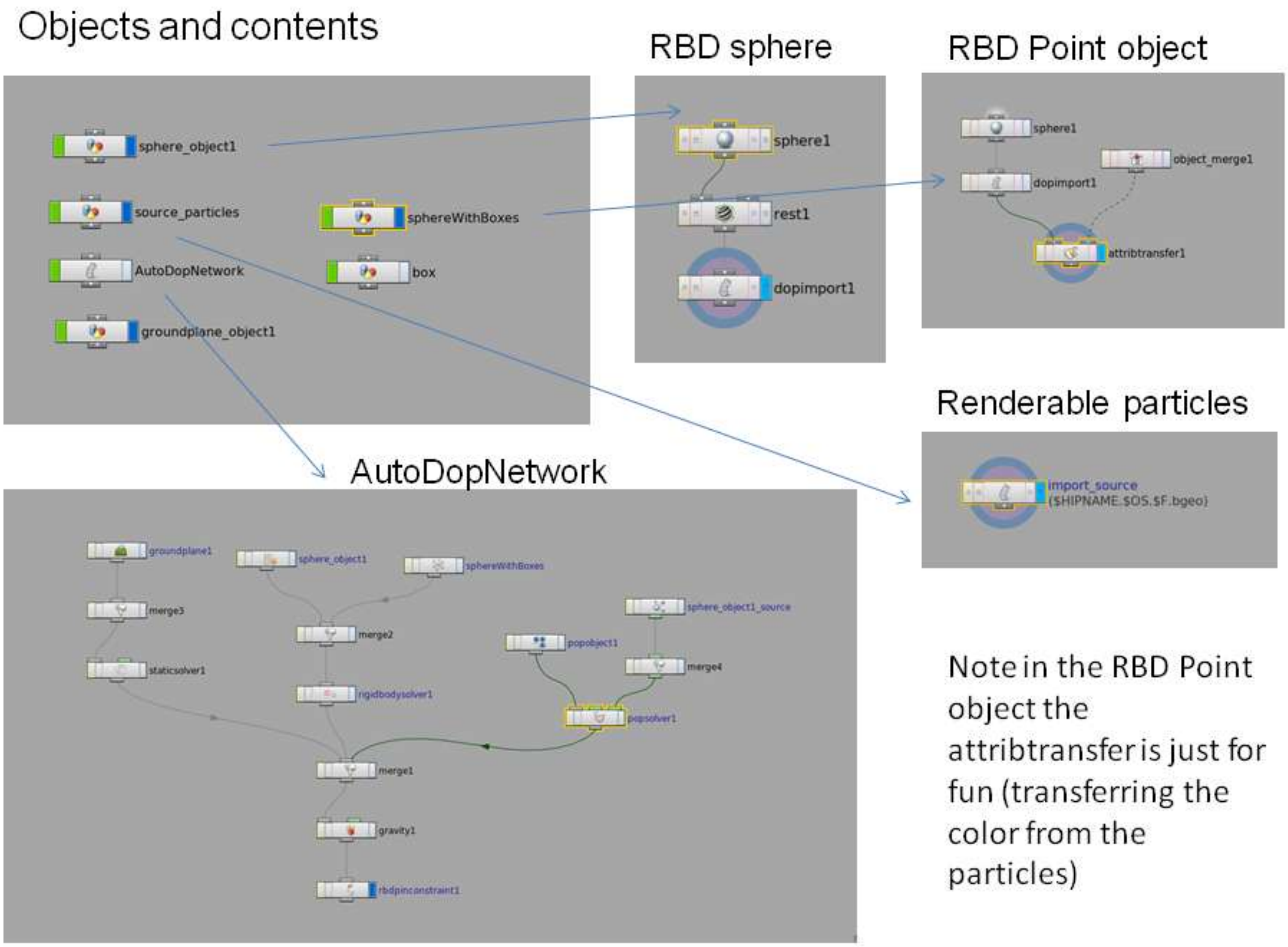
1. Turn on the Color Hits checkbox
2. Prepare to transfer this attribute in the sphereWithBoxes by creating an object merge node and selecting the source_particles/import_source
3. Add an attribute transfer and connect the object merge, just transferring the Cd value in this case. Adjust the distance condition accordingly (say .5)

Here is what we have now:





AutoDopNetwork



Objects and contents

RBD sphere

RBD Point object

Renderable particles

AutoDopNetwork

Note in the RBD Point object the attribtransfer is just for fun (transferring the color from the particles)