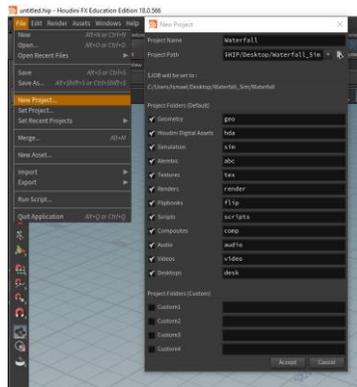


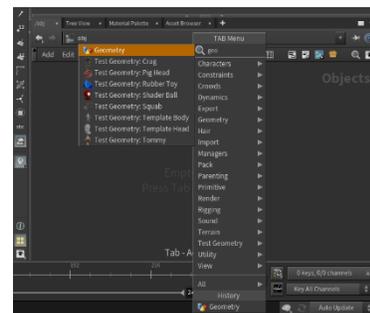
Waterfall Simulation in Houdini

Houdini is a very powerful software package and has tools for multiple parts of the pipeline. This paper will focus on Water Simulations specifically to create a waterfall. While there are several built in tools that Houdini has to accomplish the different parts of a waterfall, it is hard to understand what is going on under the surface of all the nodes it creates automatically and can be very confusing looking for the different attributes that will allow you to fine tune the waterfall for the style you are going for.

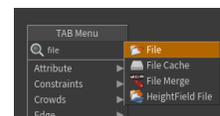
1. We will start with a new file and a new project. Set Project to the New Project that was just created. Make sure any terrain or other objects that you want to bring in from outside of Houdini are located under the geo folder in this project. In this case we have an fbx terrain object that is going to be brought in.

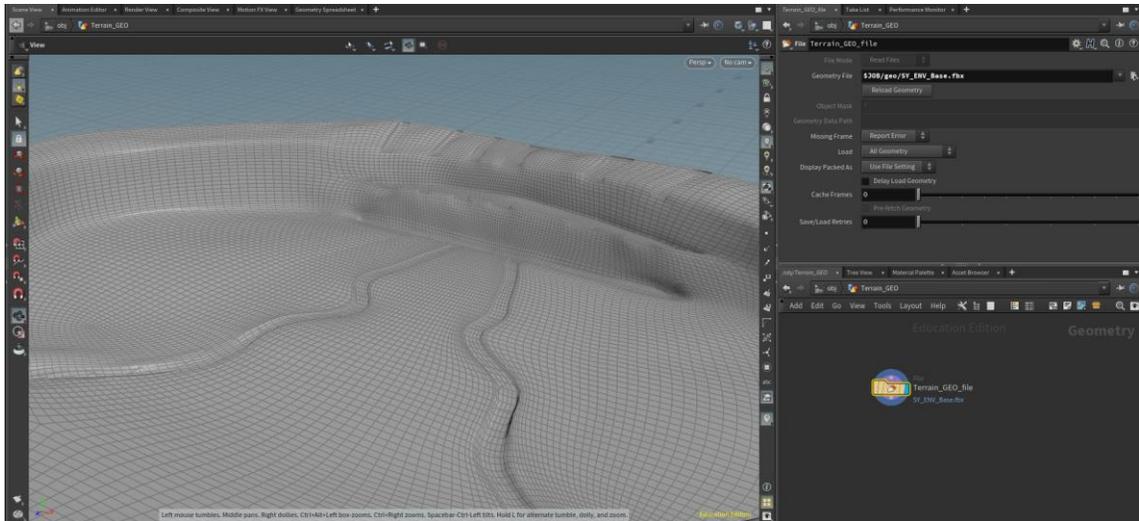


2. In the Network View at the bottom right, drop down an empty Geometry node. You can do this by hitting the Tab key which will then give you a dropdown where you can search for the nodes you are looking for. Type in Geometry and it will search for any nodes with that as part of the name. Once you find the node, you can click on it or hit enter when it is highlighted and then click in the Network View to place it.

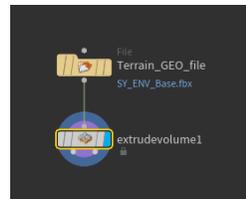


3. Rename the node to Terrain_GEO by double clicking on the name and then dive inside it by double clicking on the node.
4. Once inside, we will place down a File node and rename to Terrain_GEO_file
5. Select the File node and look at the Parameters that are above the Network View. For the Geometry File, click on the icon to the right to select the terrain fbx file to bring in.

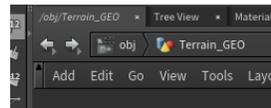




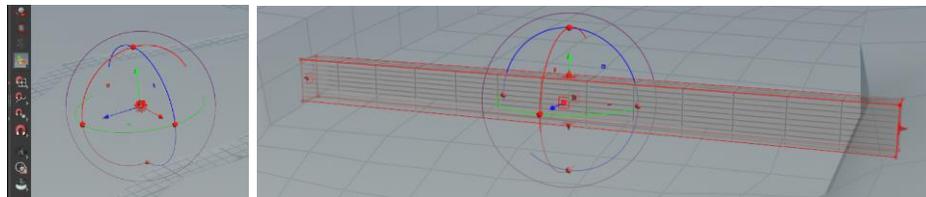
- Drop down an Extrude Volume Node and connect the output, bottom connection, from the File node to the input, top connection, for the Extrude Volume Node. This will be important for when we want to make the Terrain a collision object for the water simulation.



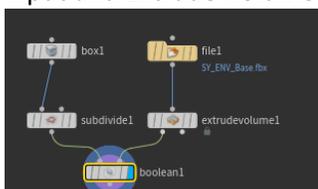
- Let's go back up to the obj level by clicking on either the back arrow or clicking on obj in the network view



- We'll drop another Geometry node and rename to Waterfall_Emitter
- Dive into the Waterfall_Emitter and drop down a Box node. This will create a cube at the origin. Using the Show Handle tool, move the box to the appropriate location where you want the Waterfall to begin. Inset the cube into the Terrain Geo. Change the primitive type to Polygon Mesh and increase the Axis Divisions



- Drop a File node and bring in the terrain object here as well. Connect the File node to an Extrude Volume. Connect the Box Node to a Subdivide. Connect them to a Boolean, Subdivide in the first input and Extrude Volume into the second. Then connect the Boolean to a VDB from Polygon



Note that each node is split into multiple flags. The right most flag is blue and when selected will show in the viewport everything that is happening up to that Node. This will become important when trying to understand what exactly is happening up to that point and for troubleshooting.

11. Go back to the obj level. From here we will create the fluid emitter. Select the Waterfall_emitter and then in the shelf to the top right, under the Particle Fluids tab, click Emit Particle Fluid.



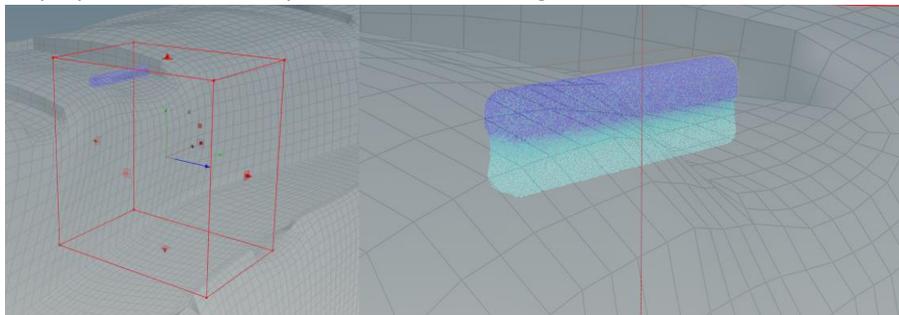
12. Houdini will prompt you to “Select a Fluid object to Emit into, if any. Press Enter to complete.” Since the Waterfall_emitter is already selected, hit Enter. Houdini will then add a series of nodes that has default values for the Water Simulation. There is a problem though. If we hit play on the timeline, nothing seems to happen besides the newly created particles seem to shuffle around a bit. This is because as part of one of these nodes, there is a limit to where the water simulation can occur. Outside of this area and it kills the particles. What we are seeing is the new particles that are created each frame that are promptly being killed in the next.

- a. When playing simulations, make sure that you have the Real Time Toggle on which you can find at the bottom left with the watch icon.

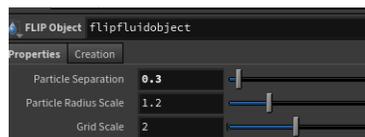


13. To fix this, in the AutoDopNetwork that was created, select the flipsolver1.

At the origin, we will see the outline of a box. This is the actual bounding box for where the simulation can occur and any particles outside this area will be killed. Move this box so that it encompasses the emitter and the area that you want the simulation to occur. You can also change the size of this bounding box by using the red handles on the sides. Now when we hit play, we can see the particles will fall straight down.



14. Before we set up the collision settings, let’s tweak a couple settings. Select the flipfluidobject. Change the Particle Separation to 0.3 to reduce how many particles are created. This can be increased later when the simulation gets finalized for better detail. For now, we don’t need to full detail so this will help speed up simulations.



- In the Waterfall_emitter, select the create_surface node. Scroll down in the Parameters until you see Velocity and check the box to its left. This will give the particles a velocity as they are created. For my terrain, I'll set this to 10 on the Z. The input boxes go in order of X, Y, Z so the third input box is the one we want in this case.

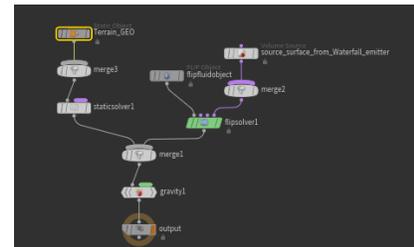


- Let's set up the collision object now. In the obj level, select the Terrain_GEO.

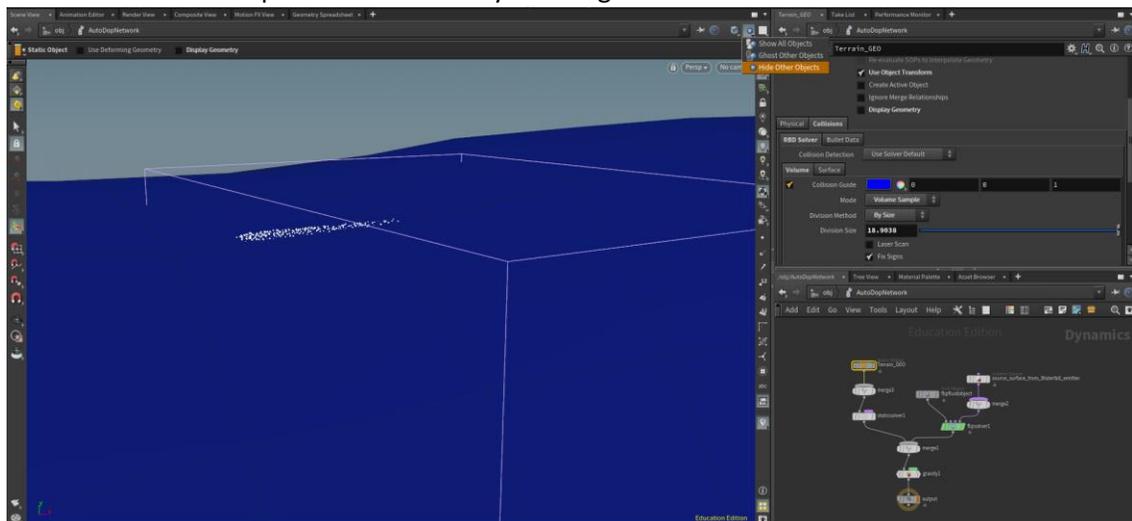
- Go to the shelf in the top right, under Collisions, and click on Static Object. The particles will now collide with the Terrain_GEO however if you hit play things won't look right. That is because the actual collision geometry doesn't match the geometry that we see.



- Dive into the AutoDopNetwork and hit "L" to organize the nodes. A couple of new nodes were created and connected to the simulation when we created the Static Object. Select the Terrain_GEO static object as we want to change its parameters

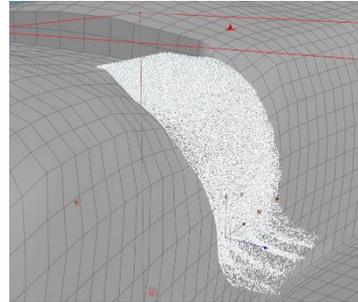


- Scroll down in the Parameters and uncheck Display Geometry for the moment and also in the Viewport select Hide other Objects. Click the Collisions tab and turn on the Collision Guide. This will show us what the particles are actually colliding with.

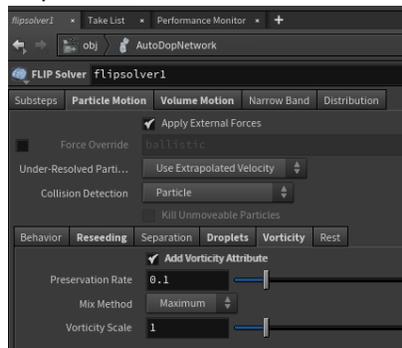


- Change the Division Size to 1 for now. This will cause the collision mesh to match fairly closely to the geo but not 100% accurate. The smaller this number the more accurate the Collisions will be but the less efficient the simulation.

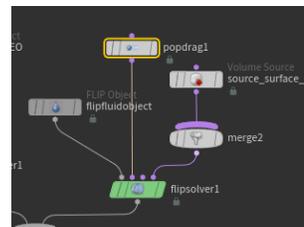
20. We can turn the Collision Guide back off, turn Display Geometry back on and Show All Objects in the Viewport. Now if we play the simulation, we can see that we get a fairly accurate collision of the water simulation to the terrain.



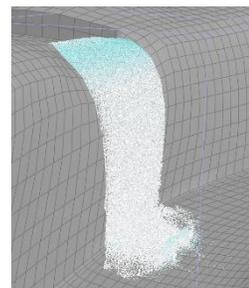
21. Select the flipsolver, and under Particle Motion turn on Droplets and Vorticity.



22. Drop down a Pop Drag and connect it to the flipsolver's second input. This will introduce air resistance to the simulation.



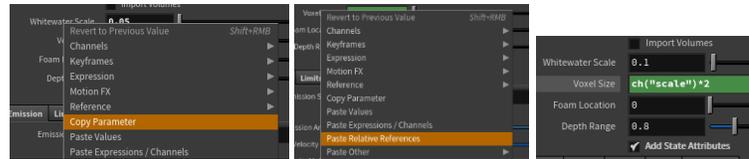
23. Select the popdrag and change the air resistance value to 0.05. This will slow down the simulation a bit but will also help keep the particles from flying apart as much as they were.



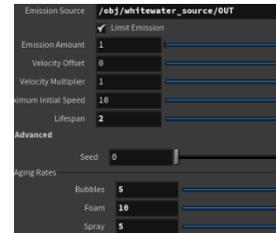
24. Select the flipfluidobject and click on Whitewater in the Particle Fluid tab in the shelf. The Whitewater simulation is used to represent the foam, spray, and bubbles that you don't see from the default water simulation. This simulation will interact with the water simulation and will collide with it while staying on top but its default values cause a lot of this whitewater to be initially created so let's tone it down.



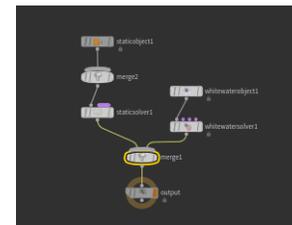
25. In the new whitewatersolver, change the Whitewater Scale to .05. Right click Whitewater Scale and Copy Parameter. Right click on Voxel Size > Paste Relative References and multiply the pasted relative reference by 2 so that is doubles in the Whitewater Scale and check Add State Attributes



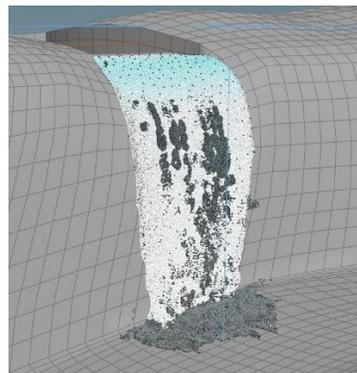
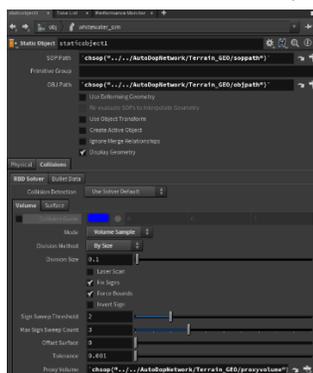
26. Under Emission, let's reduce the amount of whitewater that is created. Reduce the Lifespan to 2. Increase the Aging Rates for Bubbles to 5, Foam to 10, and Spray to 5. This will help keep the whitewater from existing for very long times.



27. While the whitewater collides with the water simulation, it doesn't collide with the terrain so it will need to be added in. Drop a static object node and connect it to a Merge Node. Connect the Merge Node to a Static Solver Node. Connect the Static Solver Node to the Merge Node that the whitewatersolver is already connected to and make sure the staticsolver connects to the left side indicating it is plugged in first. You can change the order by clicking on the Merge node and dragging the staticsolver to the top.



28. We are going to want the same values that the static object has in the AutoDopNetwork for the water simulation. The parameters that need to be copied over are the SOP Path, the OBJ Path, Mode, Division Method and the Proxy Volume



At this point we have a basic Waterfall Simulation going on. We could always tweak the values that have been placed to change up the look and style. We can increase the resolution of the simulation if we wanted at this point before exporting out the file as an alembic cache for importing to Maya. There are also plenty more forces that we could add to the simulation to increase the randomness and realism to the simulation.