Introduction to Dynamics

- General Overview
- Anatomy of a DOPNET

- pops are now part of dops – (old pops exist but you should try to get used to the new method – that's next week's topic)

Dynamics

Dynamic Operators (DOPS) are used to create physically accurate dynamic effects

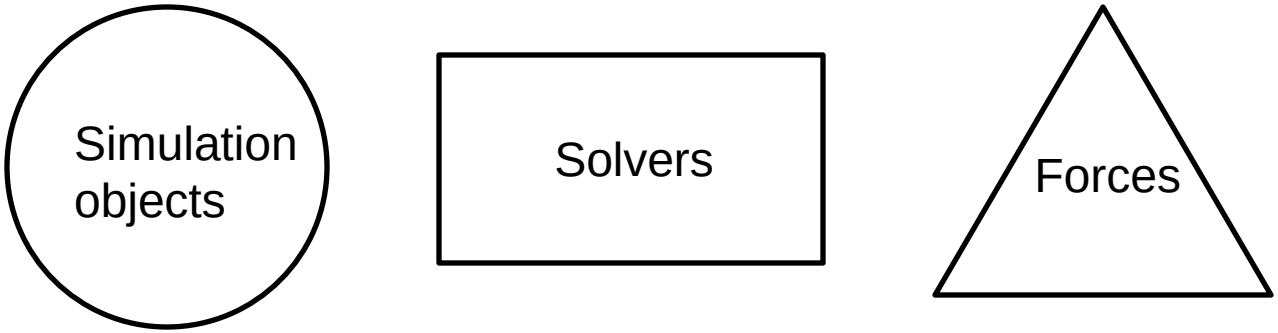
Scale is important!

Houdini dynamics use units of kilograms, meters, and seconds

(1 Houdini unit = 1 meter)

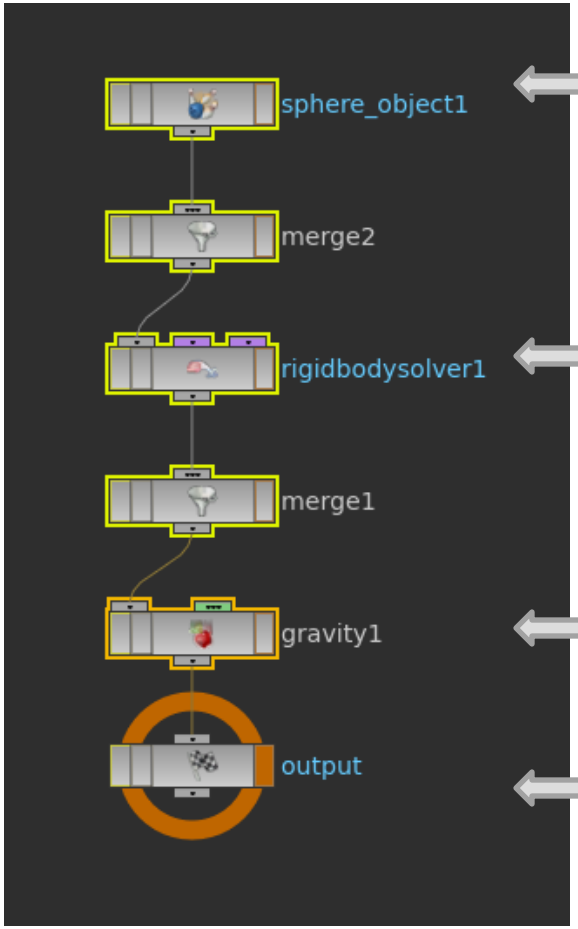
Dynamic networks

- already seen from 350 – RBD
- Anatomy of the autodopnetwork is similar to the setup for fluids, pyro, cloth and wire
- Dynamics networks create simulations



Apply one or more solvers to the simulation object

Basic setup
(AutoDopNetwork is where Houdini puts the nodes defining the simulated object's behavior)



DOP Object node – container for data. Specific object nodes import geometry data for a object type, such as RBD object, Cloth etc.

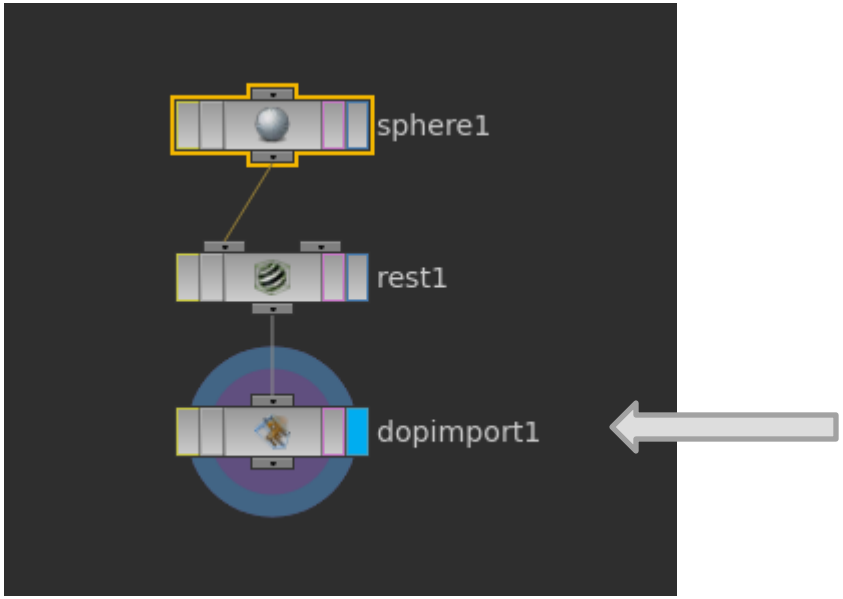
Solver node - attaches data to the object, so that the object's position etc. will be calculated using this solver.

Force node - attaches data to the object, indicating that this force is acting on the object.

Output node - **new in 14** - used to mark the end of a DOP simulation chain. It should normally always have the Output flag set on itself. Can be used like a Dynamics ROP.

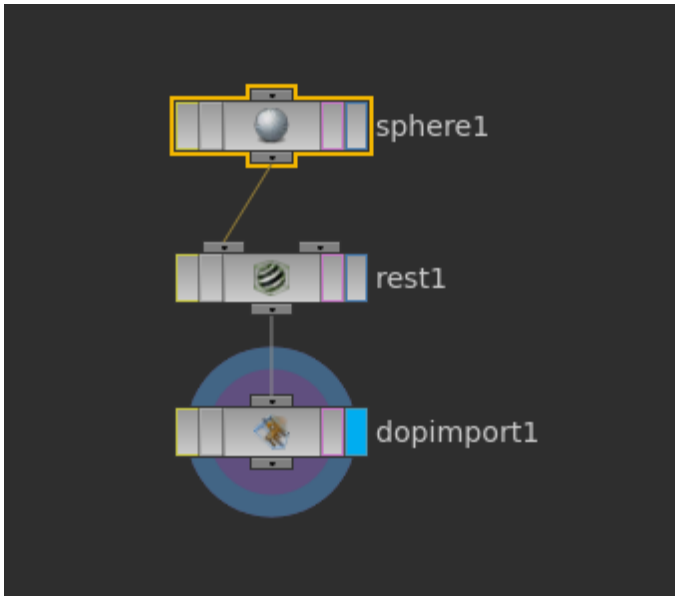
Inside the sphere container is where the calculated information is imported back

think of the DOP network as a calculator



Imports the information back (for rendering for example)

rest and **dopimport** are added



rest – locks the transformation of the object (particularly for texturing)

dopimport – applies the tranformation that the object goes through in DOPS

The automatic setting is set to:
Transform Input Geometry

By hand, often
Fetch Geometry from DopNetwork
is used thus importing the geometry from DOPS to SOPS

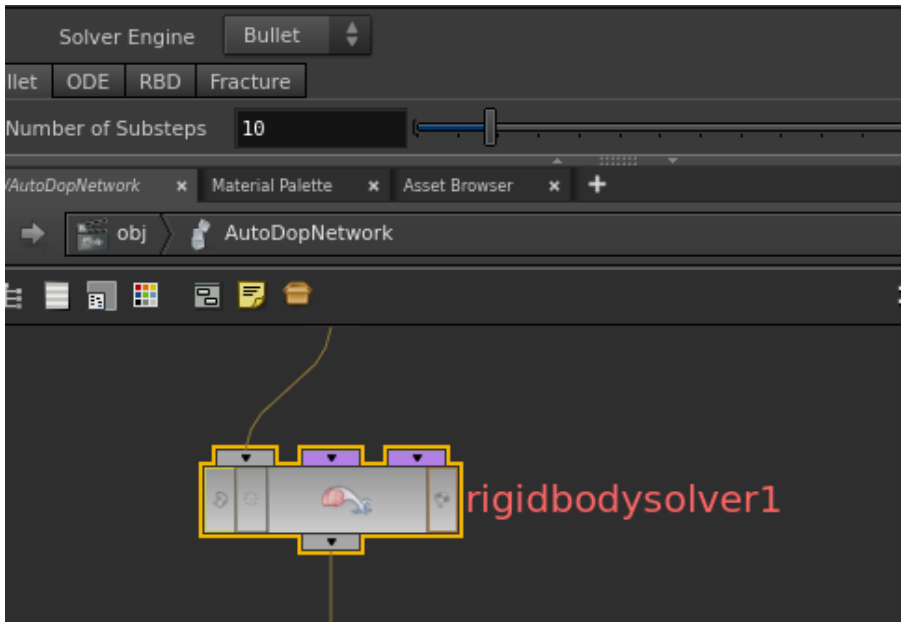
Going back to the AutoDopNetwork …

Merge nodes – unlike other networks that simply combine the output of the nodes fed in, in dops, merge nodes **create relationships**

- most commonly collision relationship, but may also be constraint relationships (pins, springs, wire, cloth)

- they take time to compute

Looking at the diagram again, note that



this node gives you a pulldown menu of which type of solver (defaults to bullet)

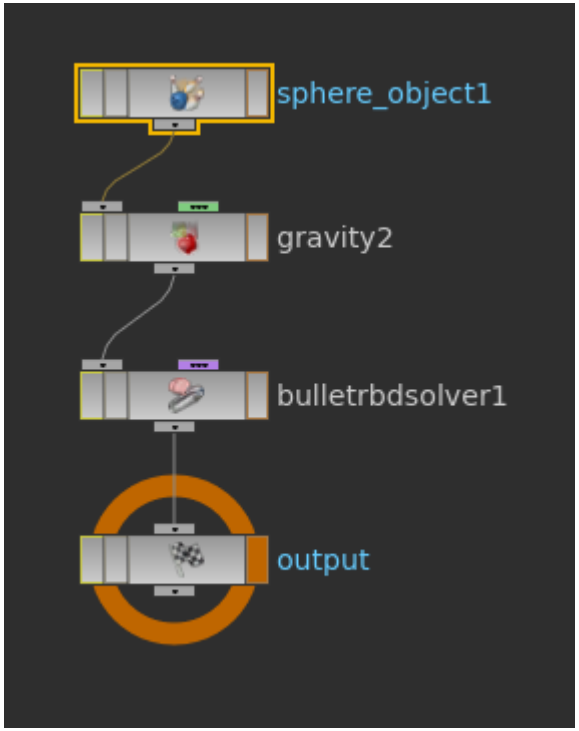
purple or green = data (previous versions was green)
gray = object + data

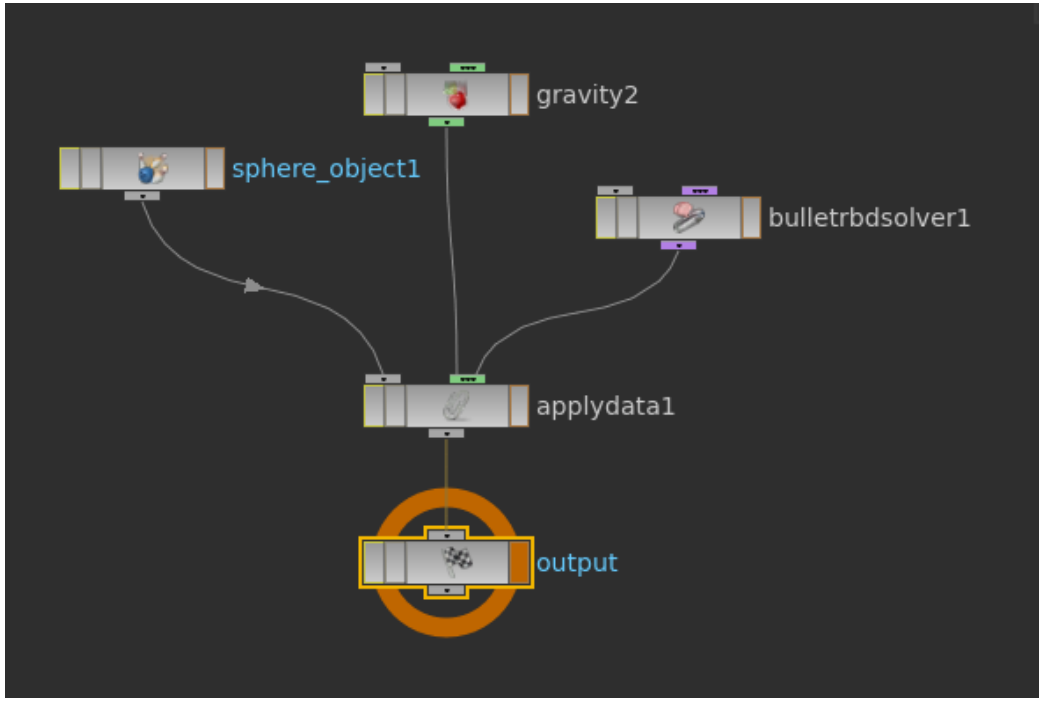
DOP Network evalutes **top down, left to right**

Two step process:

- processes the network to establish object and data attachments
- runs solvers on the object(s)

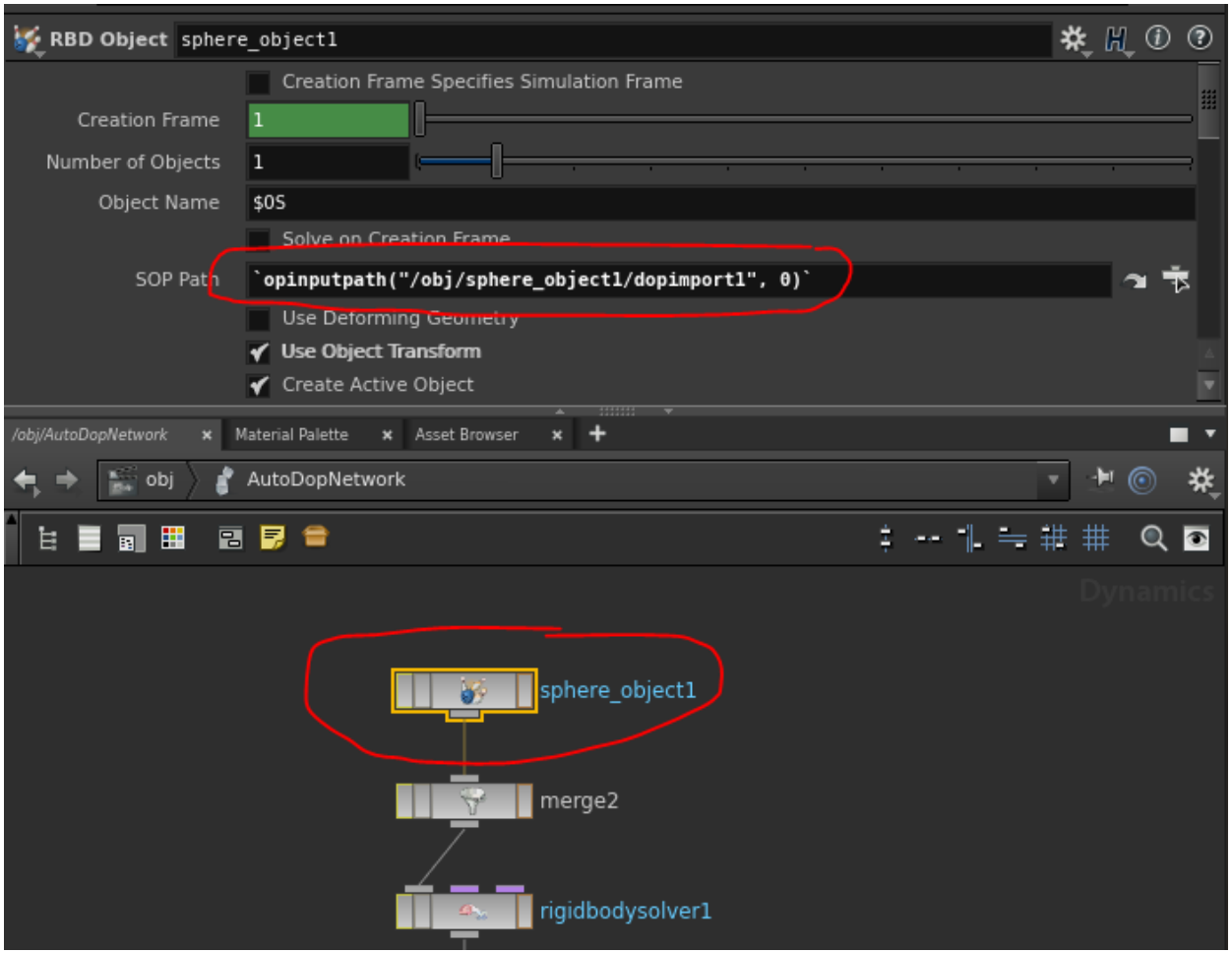
To illustrate the process consider the following networks – they are the same





note in the AutoDopNetwork in the RBD Object is the SOP Path expression:

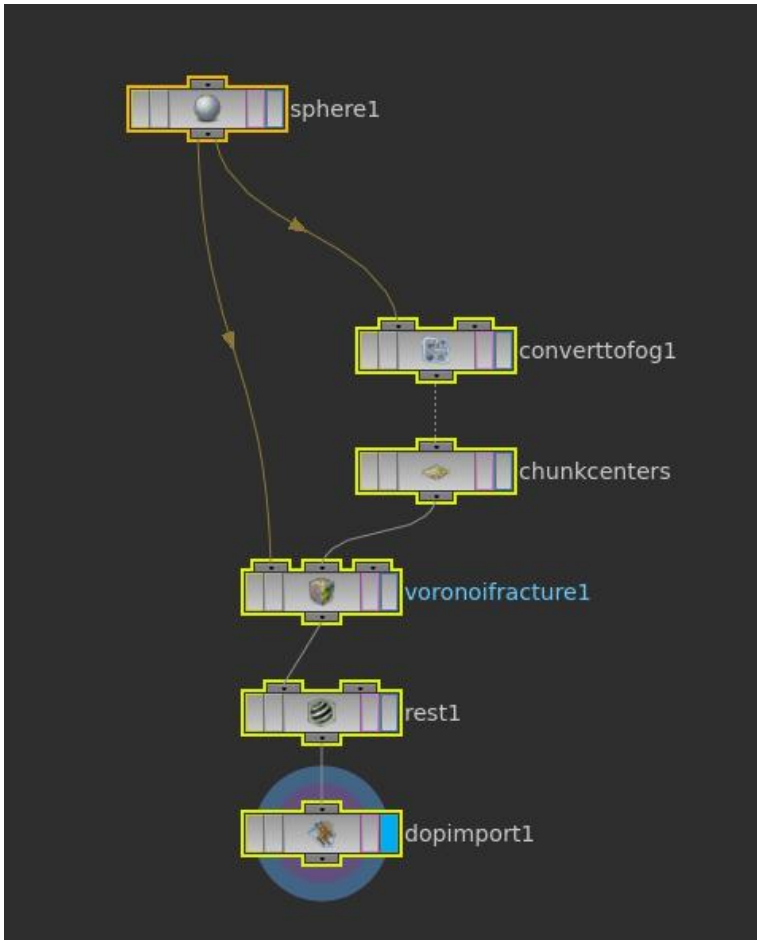
`opinputpath("/obj/sphere_object1/dopimport1",0)`



In Textport (using Technical desktop) type in
exhelp opinputpath
it states “Returns the full path of the node connected to a given input.”

So this is just an expression to use the first input into the dopimport, which is the rest node

Fracturing (Shatter/RBD Fracture tool)



Shatter tool (under Model) creates the top part of the network on the left (before the rest node)

The rest/dopimport are created when this object is made an **RBD Fractured Object**

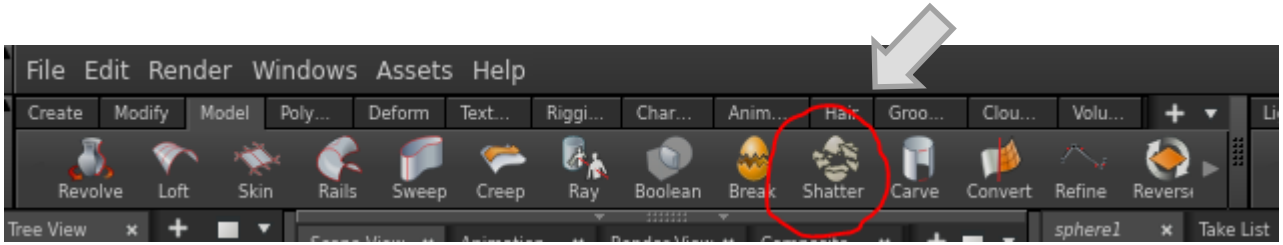
converttofog – creates a fog volume (for complex geometry, may have to adjust the uniform sampling) (isooffset node)

chunkcenters – the parameter **Number of Points** can be increased to create more pieces (scatters points in the volume)

vornoifracture - Voronoi decomposition of space around the input cell points

Try it now …

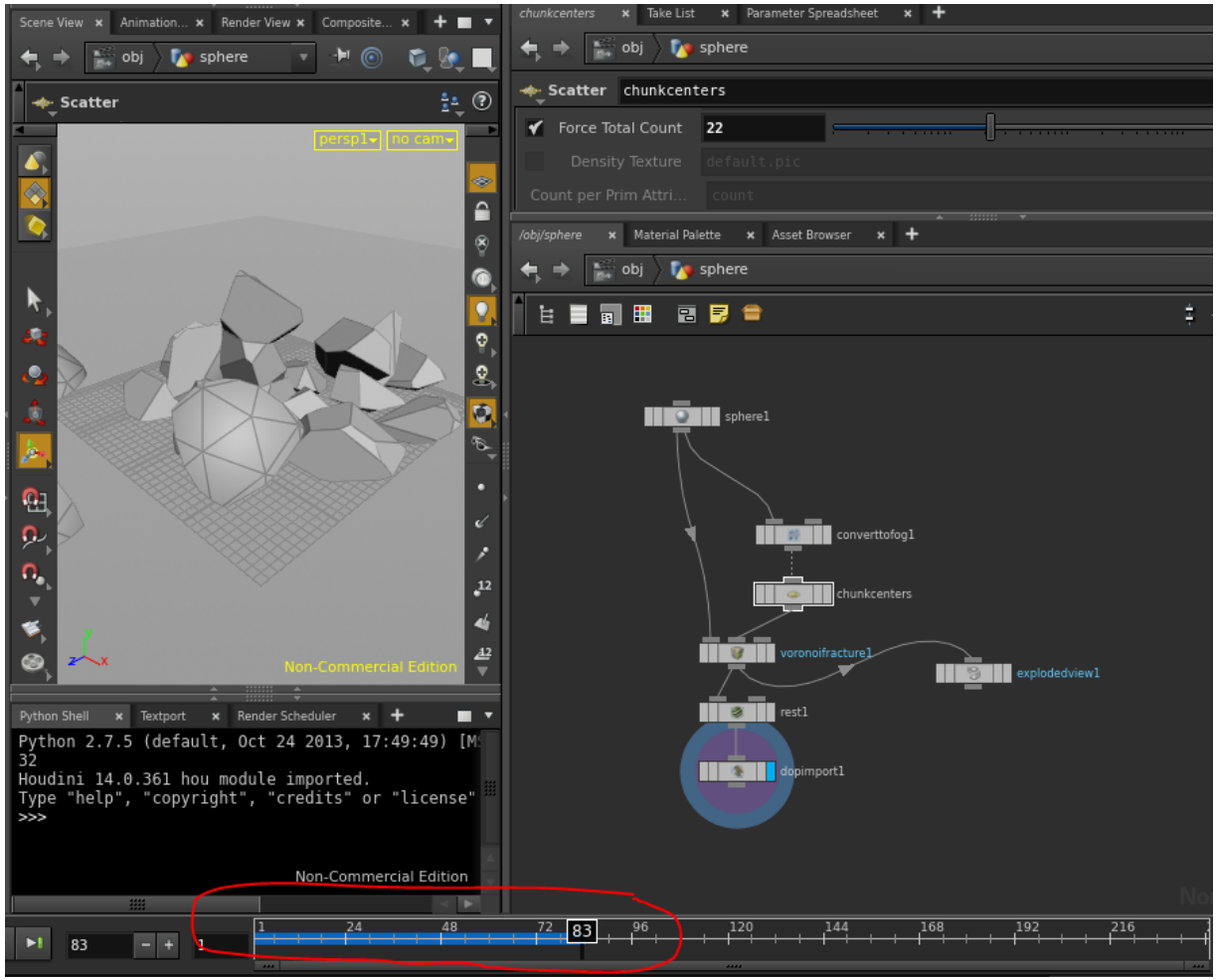
- create a sphere, move it up one unit (make sure it's a polygon)
- from the shelf tool tab model select **shatter**



- from the rigid bodies tab select **RBD Fractured Object**

- now create a Ground Plane and hit play

New in H14 the timeline shows frames that are cached in blue …



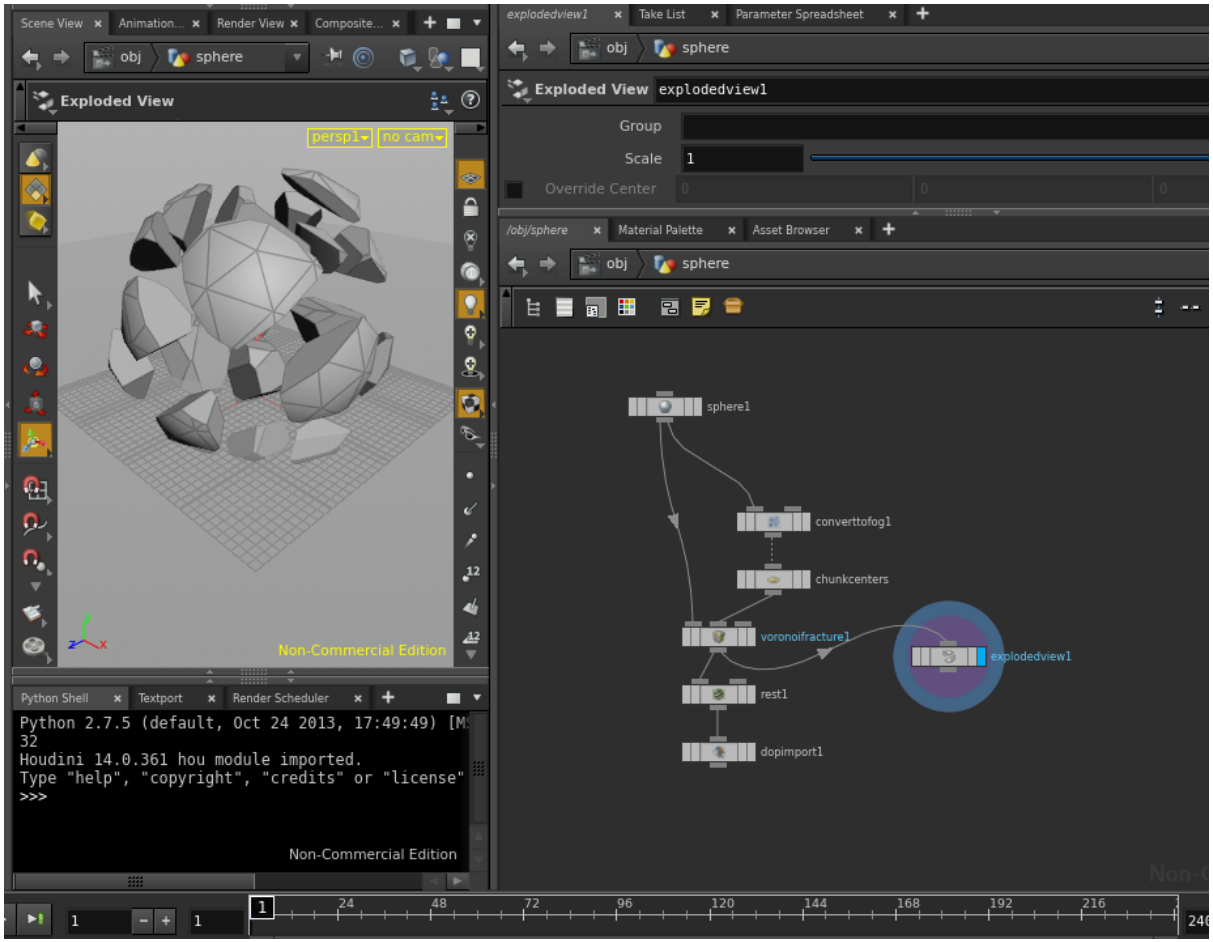
In the figure:
blue
indicateds
cached
playback

Other
indicators are:

purple
indicates .sim
files

orange
indicates
frames out of
date that need
re-simming

You can also use the **explodedview** node to visualize your fracture – we will be covering vex/vops later to customize your fracturing as well



Aside/Reminders

- use $SF and $ST in simulation nodes
- Rigid body dynamics is abbreviated to RBD
- Details view (now called the **geometry spreadsheet**) shows the data at the current time
- Viewport shows the animated geometry created by the sim

Aside/Reminders

- network establishes a tree of objects with **data applied**
- constraints – type of relationship between objects
- forces – different solvers can share the same forces

Scale is important

- **Houdini dynamics use units of kilograms, meters, and seconds**

- Since solvers are simulating real-world physical processes, they need a way to relate numbers in the scene to *real-world units*

Many examples in the documentation (you have to have houdini running to load/launch)

Also the first steps webinar M11 – (select examples in the dropbox) give a good overview of the shelf tools