Particle Systems

Term refers to a computer graphics technique

simulate fuzzy phenomena

hard to reproduce with conventional rendering techniques

Common uses of Particle Systems

fire, explosions, smoke, moving water, sparks, falling leaves, clouds, fog, dust, fur, grass, glowing trails, meteor trails, tornadoes

Particle Systems History

Bill Reeves used a particle system approach in The Adventures of Andre and Wally B (1983/4)

Using a particle system:

each tree created
position of the trees controlled

Particle Systems Pioneers

Bill Reeves used a particle system approach in The Adventures of Andre and Wally B

Craig Reynolds used particle systems to model flocking birds (Boids)

Also, Karl Sims (see class website for details)

Typical Implementation

System’s position and motion in 3D
controlled by EMITTER

EMITTER – acts as the source of the particles
- location in 3D space

System’s position and motion in 3D
controlled by   EMITTER

spawning rate
initial velocity vector
lifetime
color

and so on … usually randomness applied

Typical Implementation

Two stages

- Parameter update/simulation
- Rendering

Next

Look at particle systems in HOUDINI

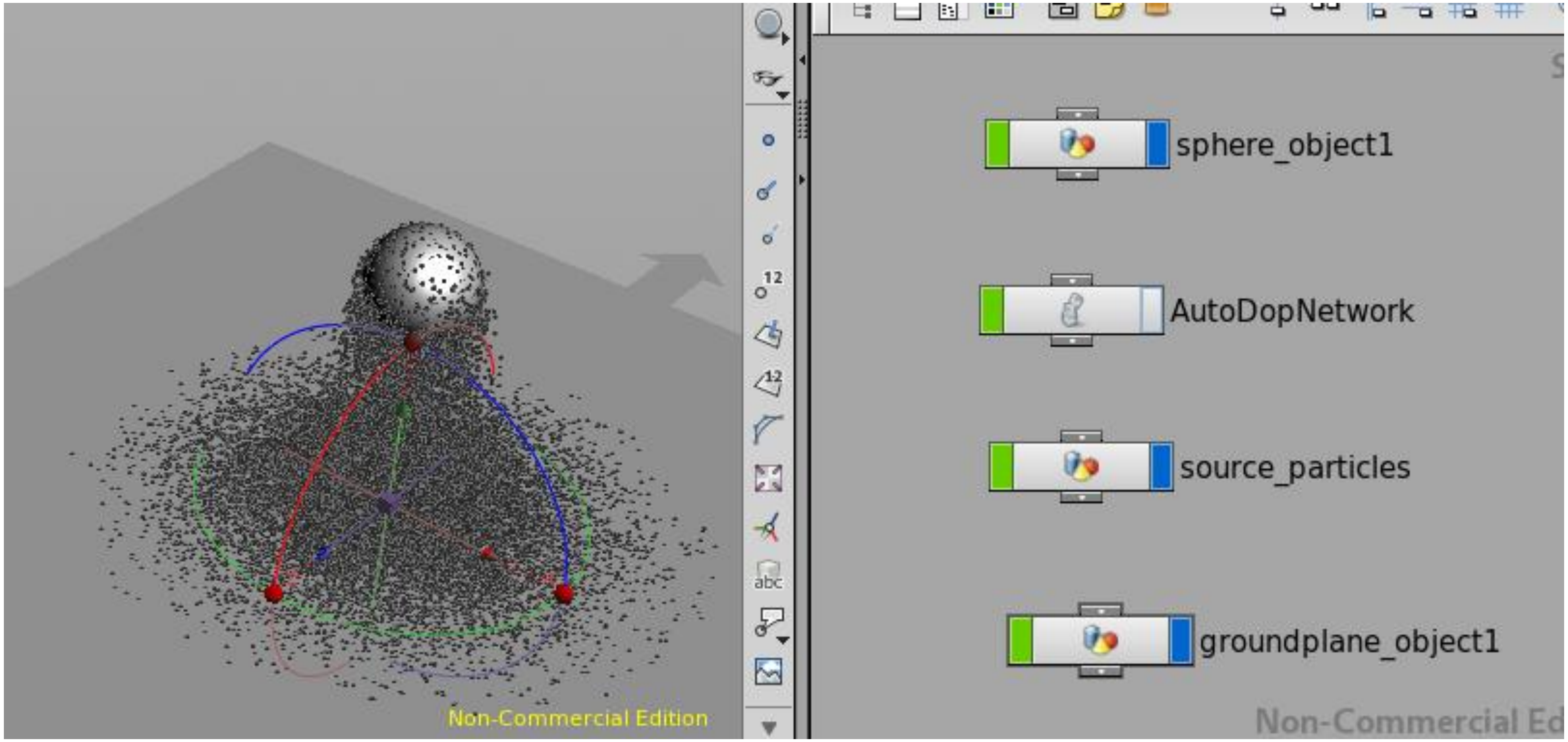
Parameter update/simulation

Rendering

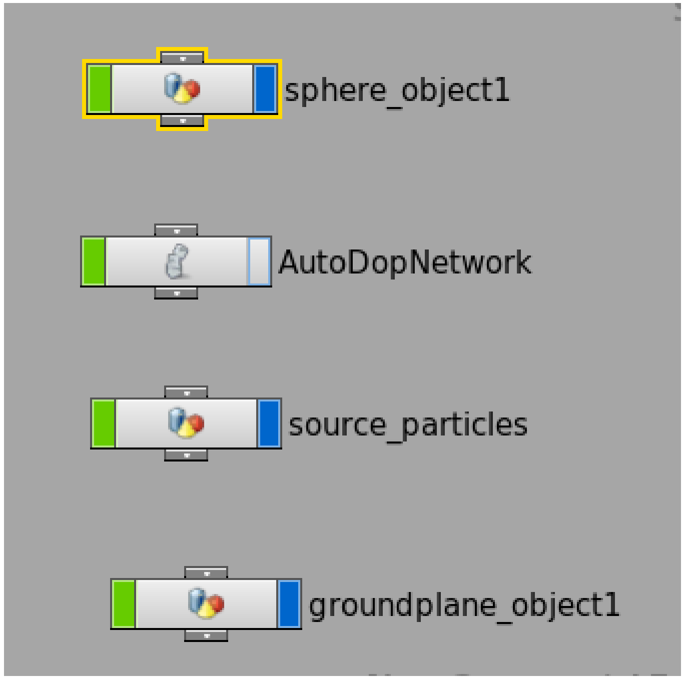
Anatomy of a Particle Network

as of H13 – particles are now fully integrated into dops

For example if you emit from a sphere:



- collisions are “built in” – in the sense that particles interact with RBD objects so the ground plane seen in the previous slide is a standard RBD static object

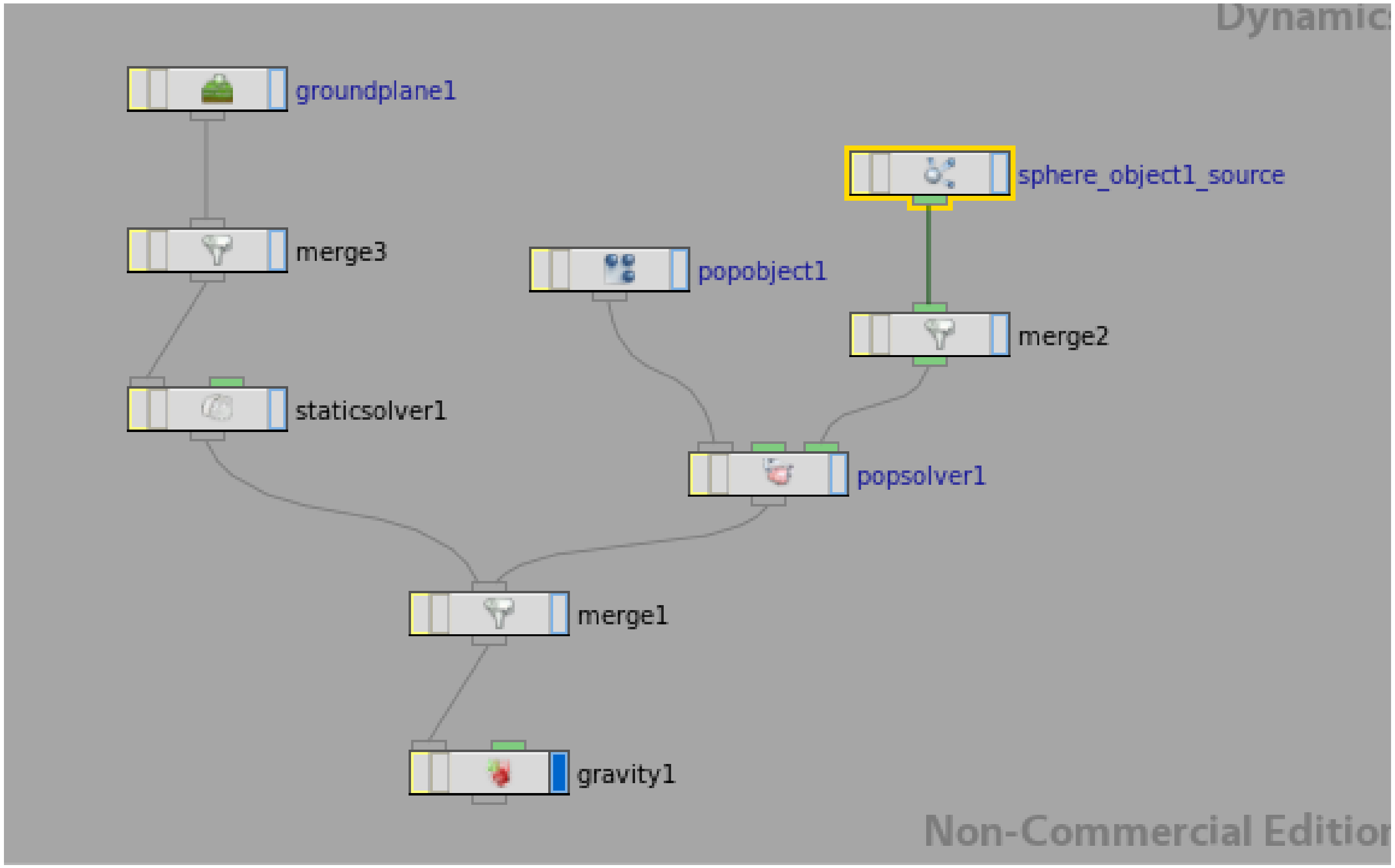


just your sphere
(and primitives work now btw – explain)

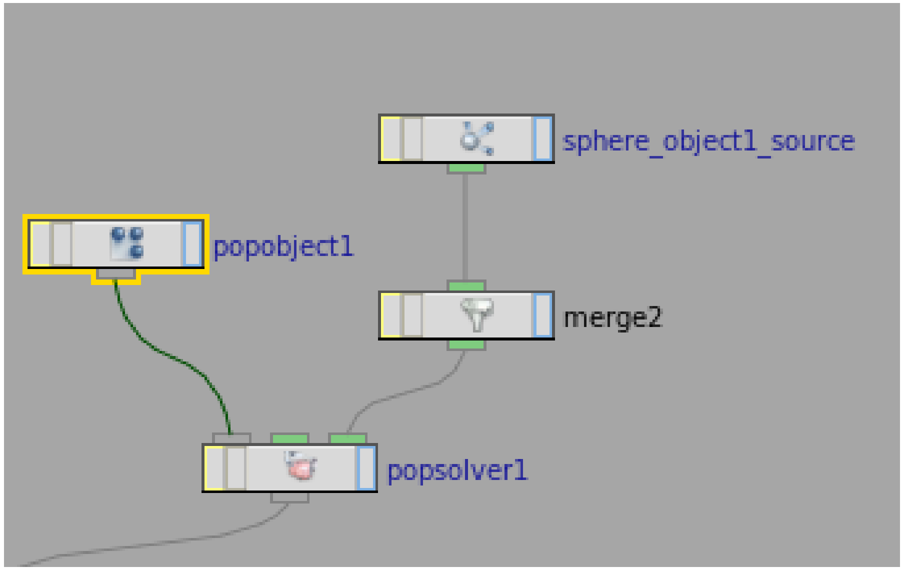
NEXT SLIDE

import from dops

standard rigid body groundplane



looks familiar on the left, let's look on the right



popobject1 - converts a regular particle system into a dynamic object for the POP Solver
- of particular interest is the physical tab controling particle behavior

popsolver1 - updates particles according to velocity and force –
- each timestep, computes where they will be next
- green inputs are POP microsolvers
- three inputs: object/pre-solve/post-solve

sphere_object1_source - **pop source** - generates particles from geometry
- this is where the birth attributes are (note – defaults to 5000, old was 100)
- usually this is a referenced SOP network

- hit the ? on the pop solver and you will find examples

  be careful as some of these examples do not use the new H13 popsolver (they still work but … I have updated the ones I have included on my page).

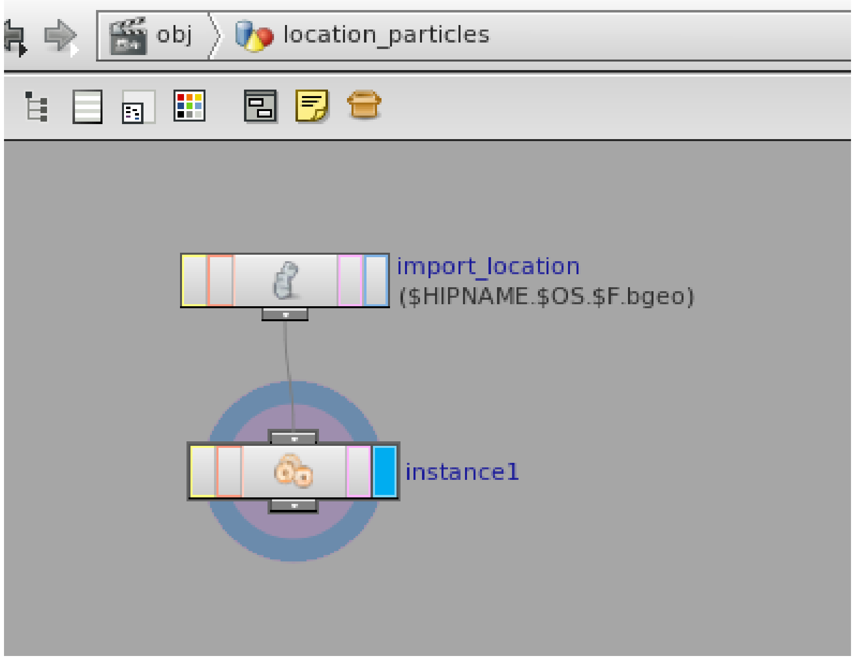
popDenting.hipnc

popMatchShape.hipnc

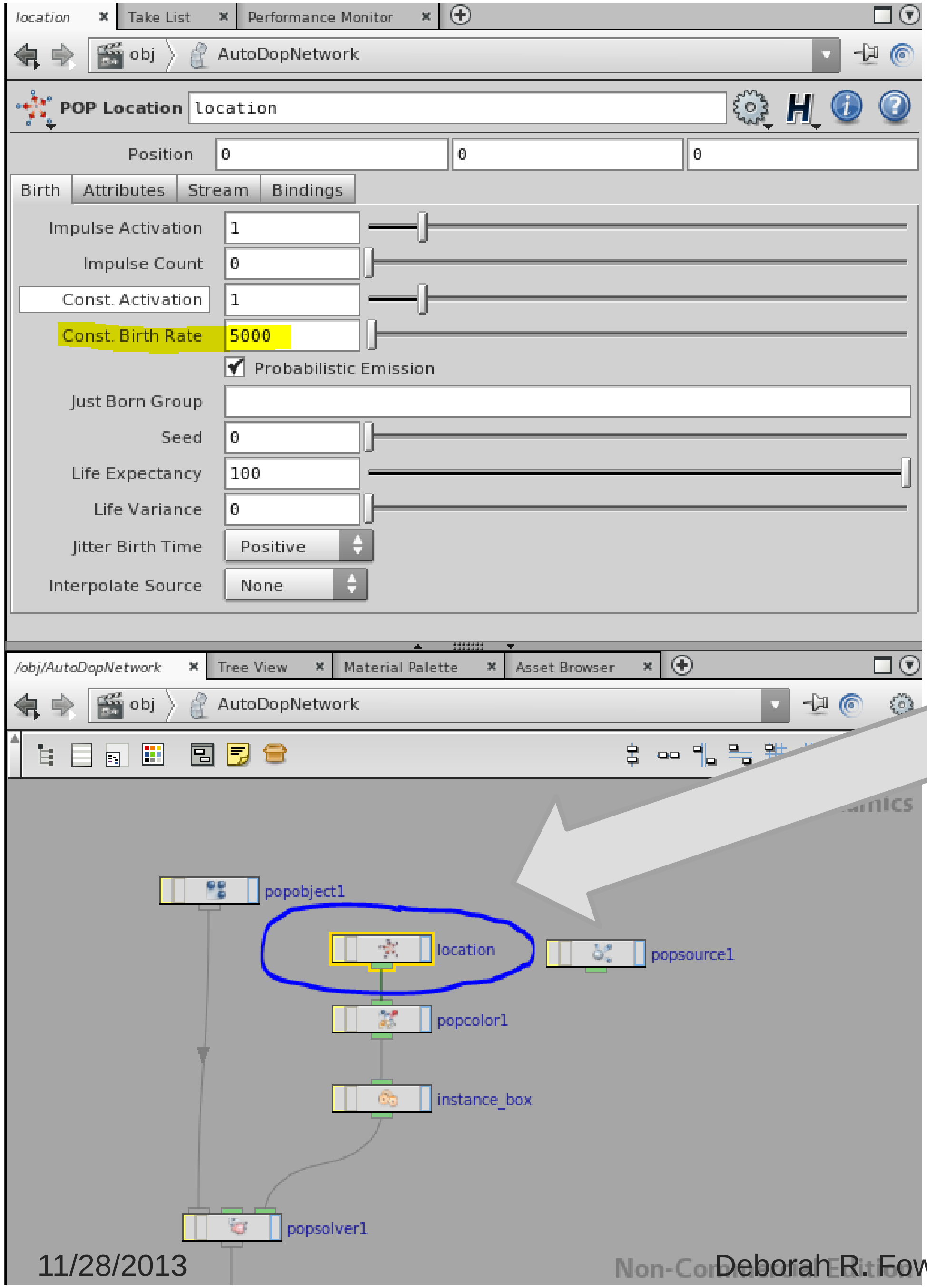
when we get to fluids, don’t forget to look at the documentation examples as a resource

Assorted Tips:

- In order to render instanced geometry, you must append an Instance SOP after the DOP I/O node inside the renderable object.



- Default birth rate is now 5000 (used to be 100) (see next slide)



Position: 0, 0, 0

Birth | Attributes | Stream | Bindings

Impulse Activation: 1

Impulse Count: 0

Const. Activation: 1

Const. Birth Rate: 5000

☑ Probabilistic Emission

Just Born Group:

Seed: 0

Life Expectancy: 100

Life Variance: 0

Jitter Birth Time: Positive

Interpolate Source: None

birth rate default 5000

here we are using a POP Location

could have instead used a POP Source (more common - node shown beside)