



Next Steps: Procedural Animation

M01 - Introduction to CHOPs

Ari Danesh
ari@sidefx.com

SIDE EFFECTS
SOFTWARE

Agenda

Course Overview and Prerequisites

What Are CHOPs?

Noise Toy Example

Triggers, Generators, and Filters

**SIDE EFFECTS
SOFTWARE**

Prerequisites

At a minimum have watched Houdini First Steps: Introduction or have equivalent knowledge

Understand the Basics of VOPs

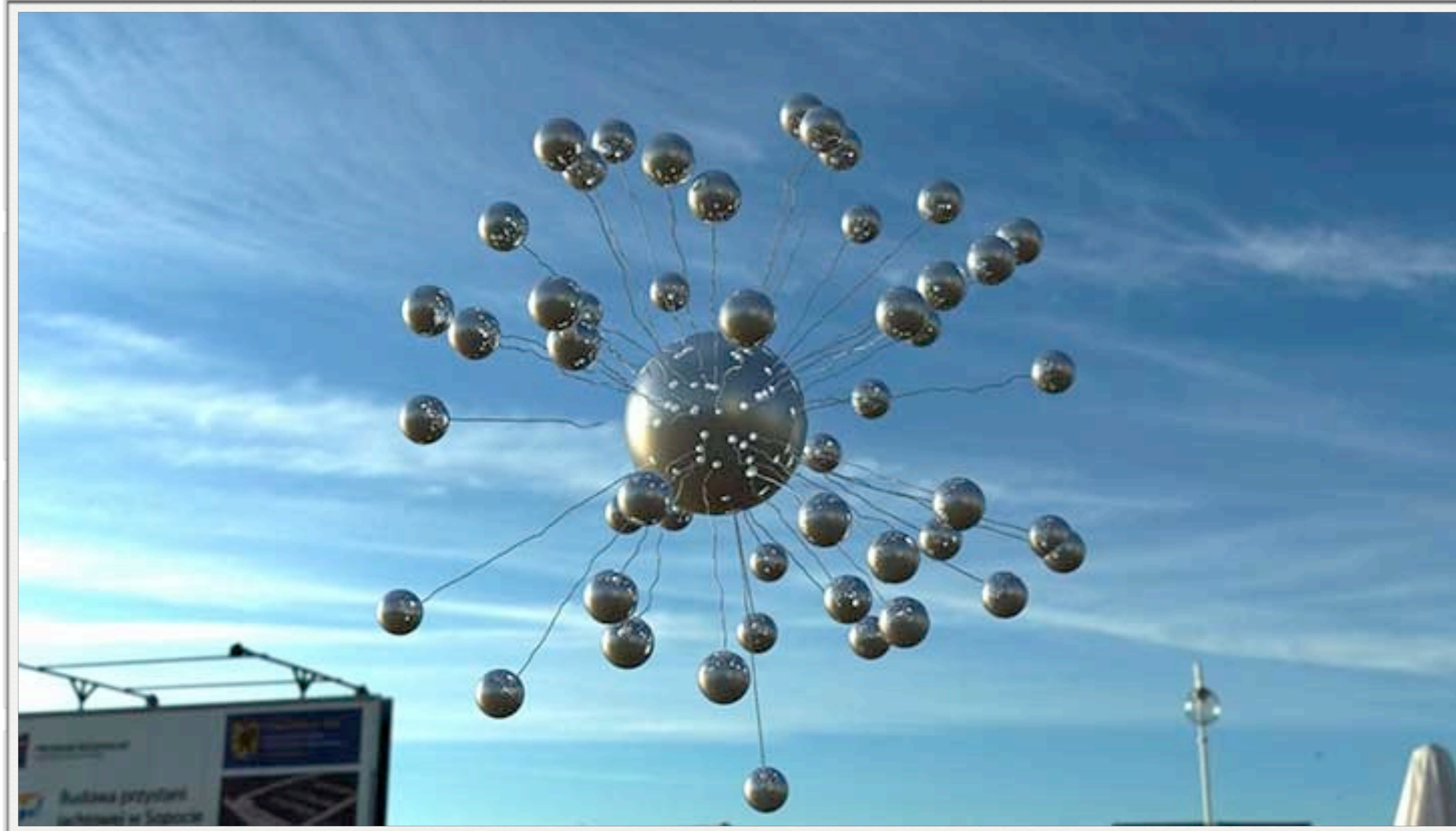
Suggested: Watch “Working with Noise”

SIDE EFFECTS
SOFTWARE

Course Overview

This eight week course is focused on getting the digital artists up and running with Houdini's procedural animation and simulation tools. The new VFX artist will also find this course invaluable as we leverage various technologies in Houdini such as Channel Operations, Particle Systems, Fluids, and Dynamics to create many different types of procedurally based animation projects.

Getting to Know CHOPs with a Motion FX Example

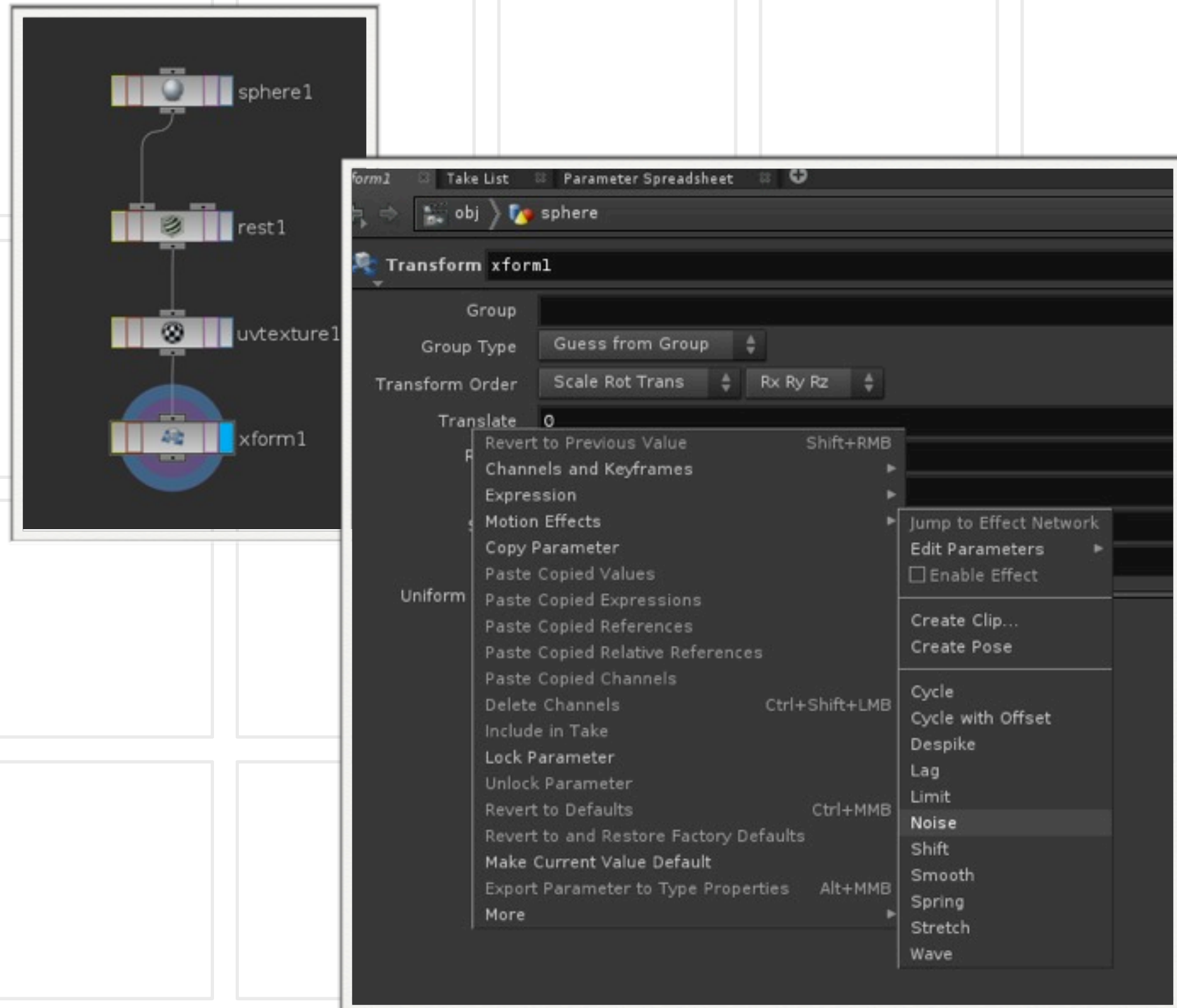


Animation we want to create

In our first example we will use Motion FX as an easy intro to CHOPs and then finish the lesson by creating our own CHOP Network

**SIDE EFFECTS
SOFTWARE**

Getting to Know CHOPs with a Motion FX Example



Drop Down a Sphere and dive inside

Make the Sphere a Polygon

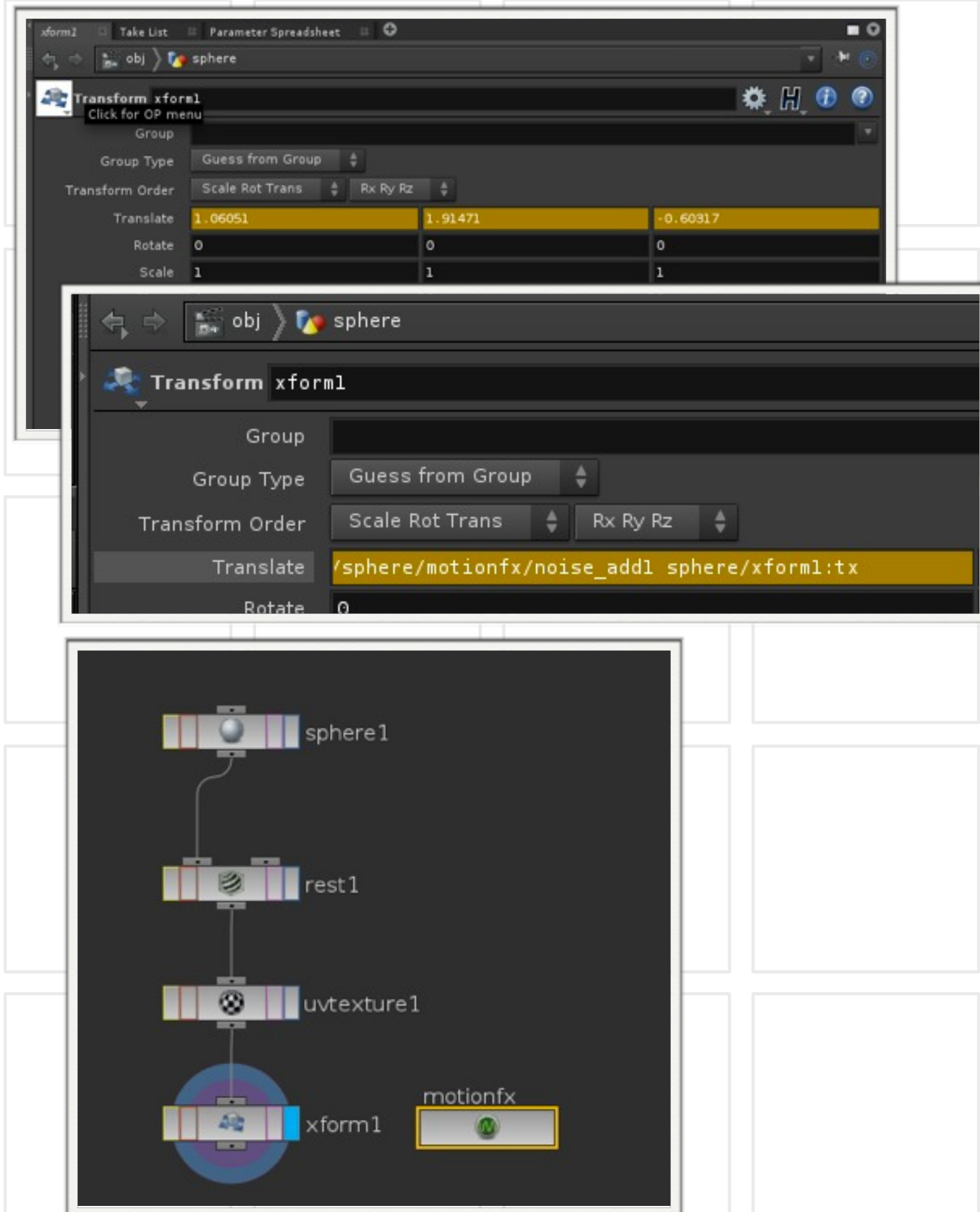
Add a rest Geometry and a UV Texture (Polar, Vertex)

Append a Transform

- ▶ Right Click on the Translate Parameter and choose
 - ▶ Motion FX --> Noise

Continued on Next Slide...

Getting to Know CHOPs with a Motion FX Example (cont.)



Notice the Translate Parameters are now highlighted in Orange

- ▶ This means the channels get their data overridden by a CHOP
- ▶ In this case by: /obj/sphere/motionfx/noise_add1 sphere/xform1:tx, ty, and tz

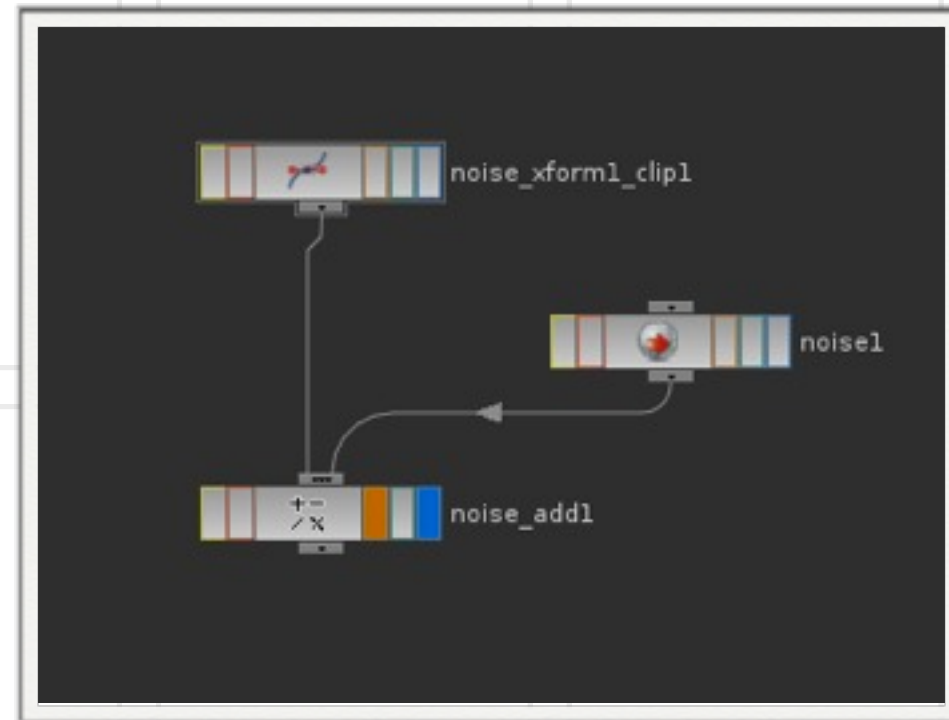
Houdini has added a CHOP network named motionfx

Add a rest Geometry and a UV Texture (Polar, Vertex)

Dive inside the CHOP network motionfx

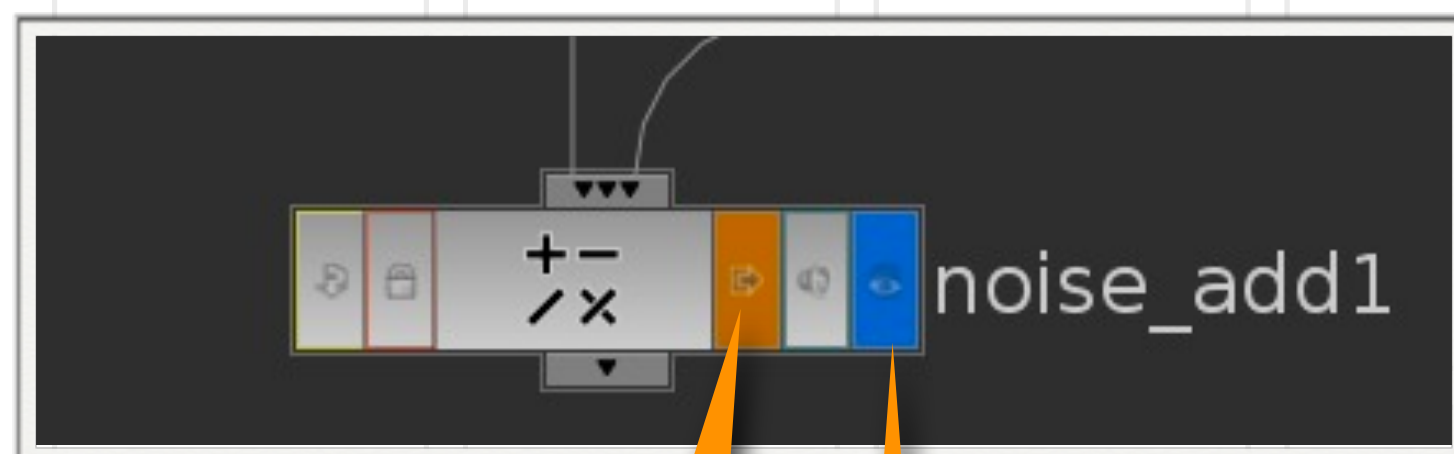
Continued on next slide...

Getting to Know CHOPs with a Motion FX Example (cont.)



Three Nodes are created for you

- ▶ Channel Node: Creates channels from the value of its parameters
- ▶ Makes an irregular wave that never repeats, with values approximately in the range -1 to +1.
- ▶ Perform a variety of arithmetic operations on and between channels. has added a CHOP network named motionfx

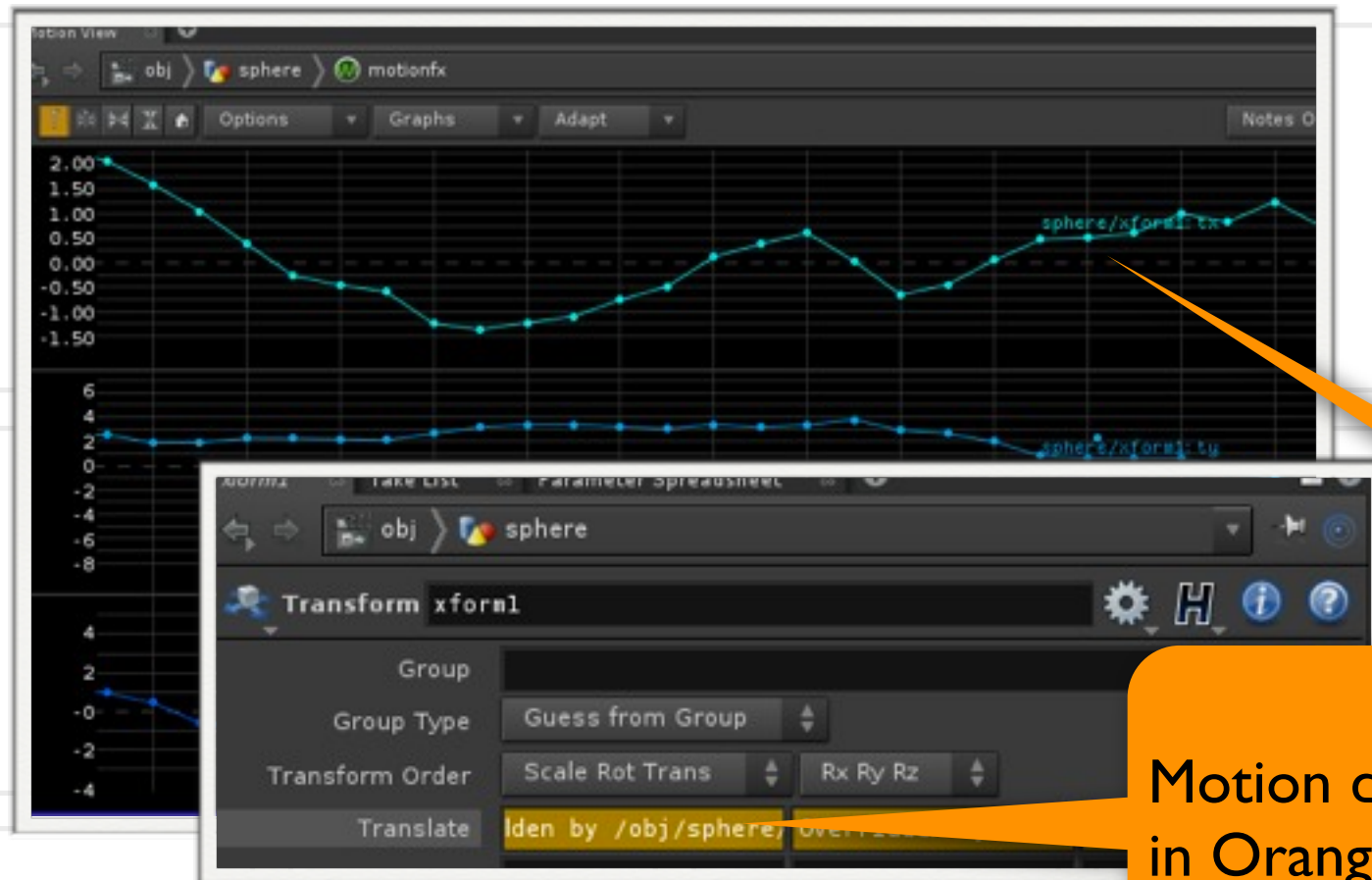


Export back to SOP Level

Display in Motion View

Notice the Flags on the Node are different then a SOP node

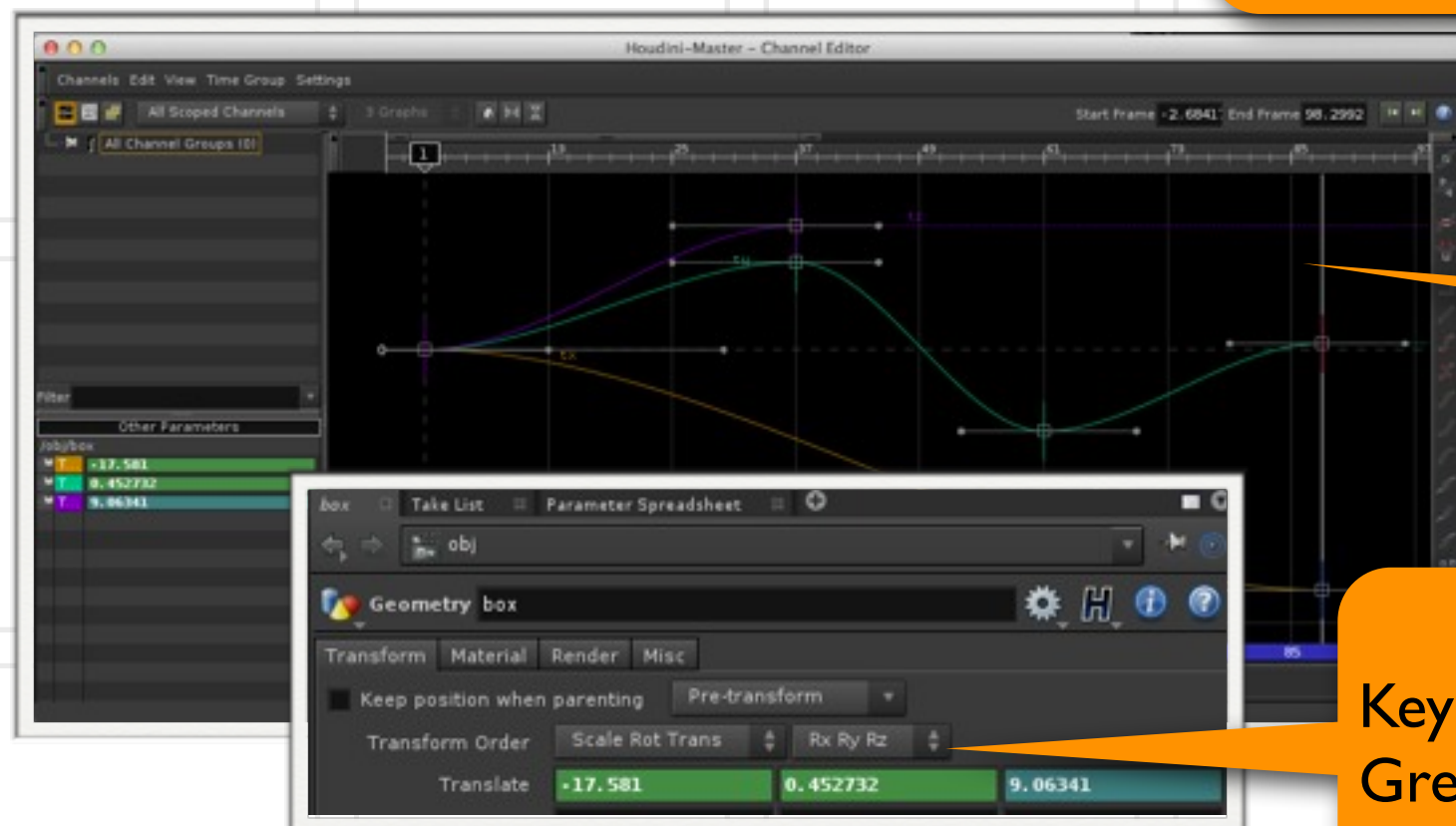
So What is a Channel?



Motion data shown in Orange

Channel Data - Discreet Sample Points at set distances apart

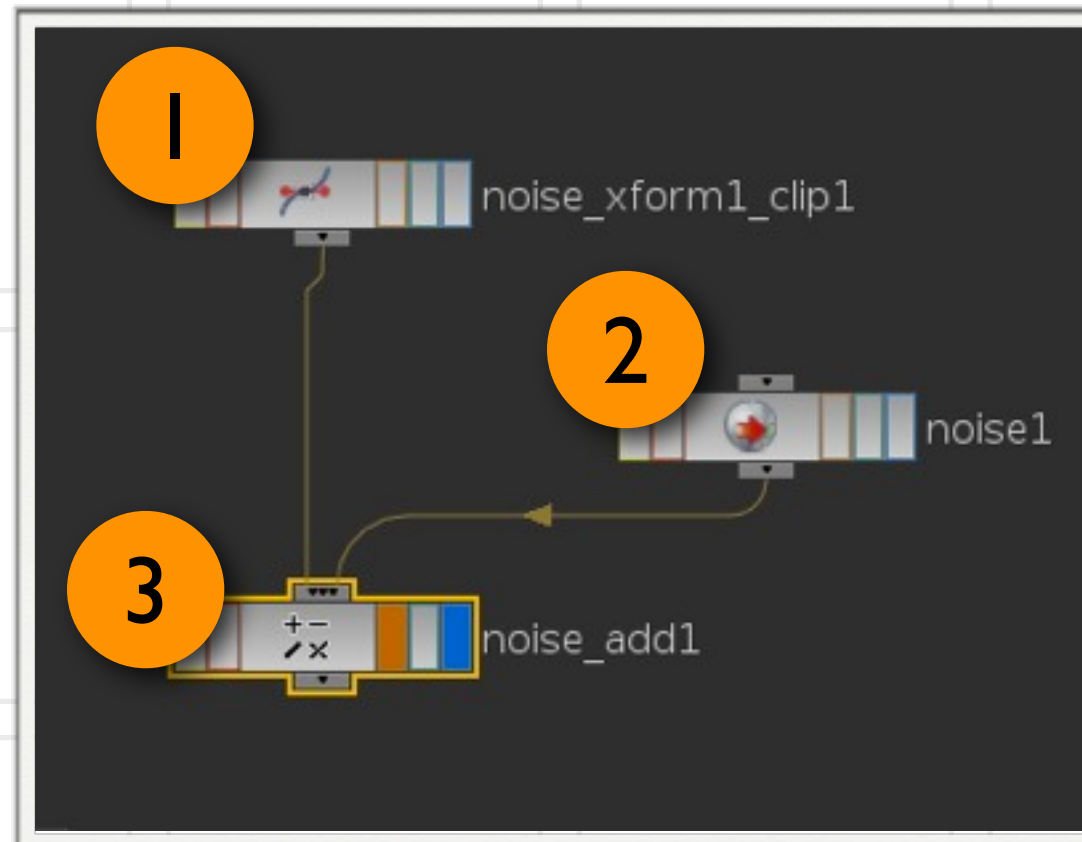
Discreet samples of data. Unlike Keyframes which contain two key frames and a function (such as a Bezier Function) between the two key frames a Channel is just a series of discreet sample points



Key framed data in Green

Keyframe data - Keyframes do not need to be set equal frames apart and bezier function interpolates between points

Basic Workflow of a CHOP



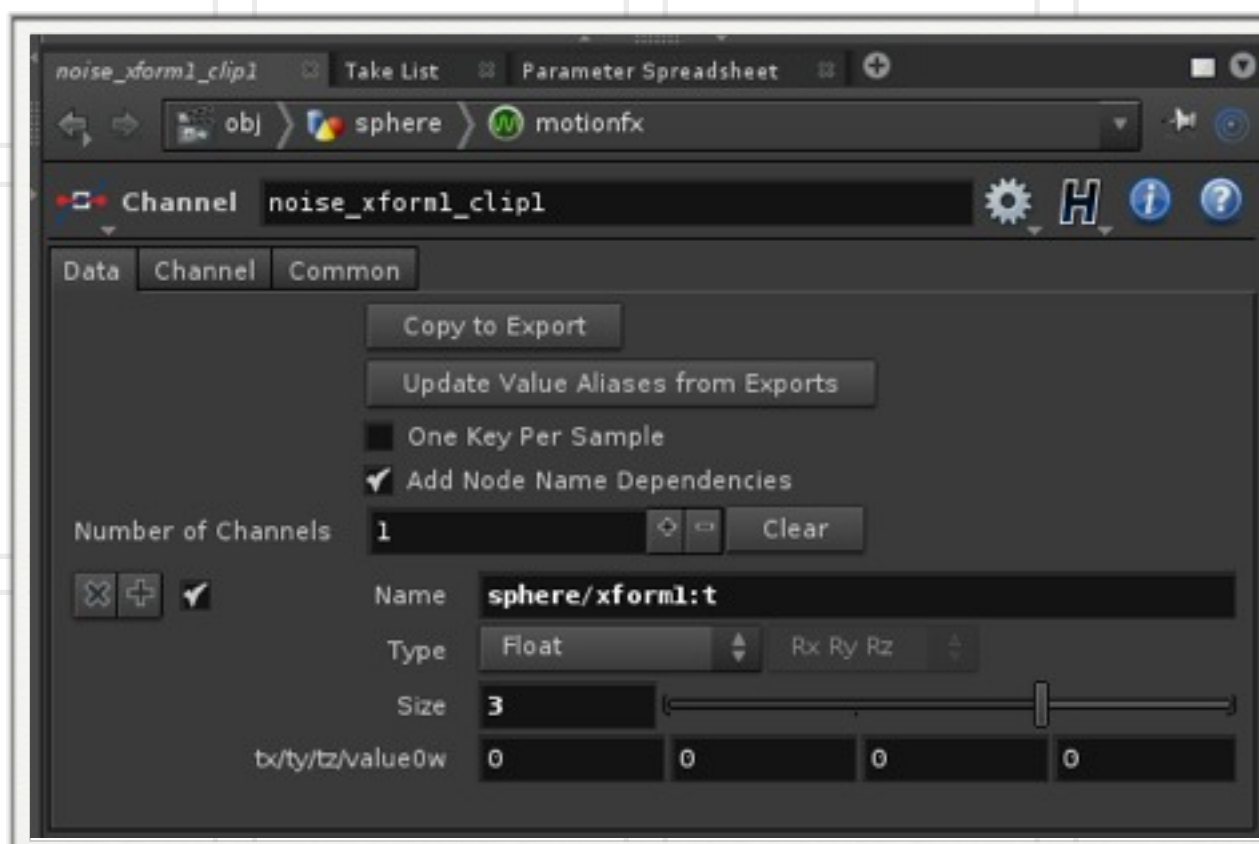
1. Use a Channel Node to Fetch data in this case from a SOP

- ▶ Use the “Name” parameter to specify where to grab data from. In this example - sphere/xform1:t

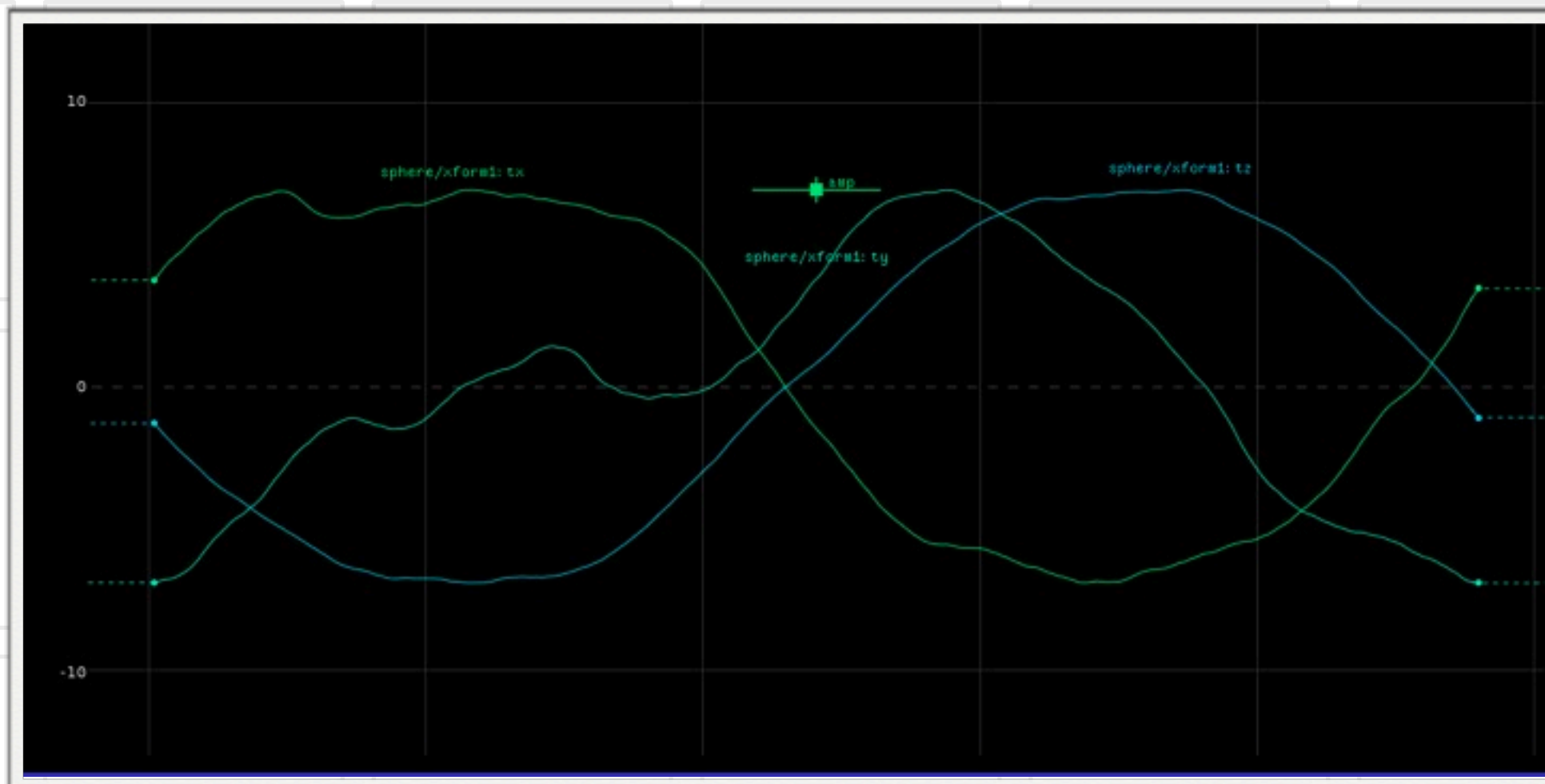
2. Use a Generator such a “Noise” to add motion without any keyframes

3. Add the Generator to the Channel data and set the export flag

We will shortly see another more common way to do this using the “Geometry” CHOP

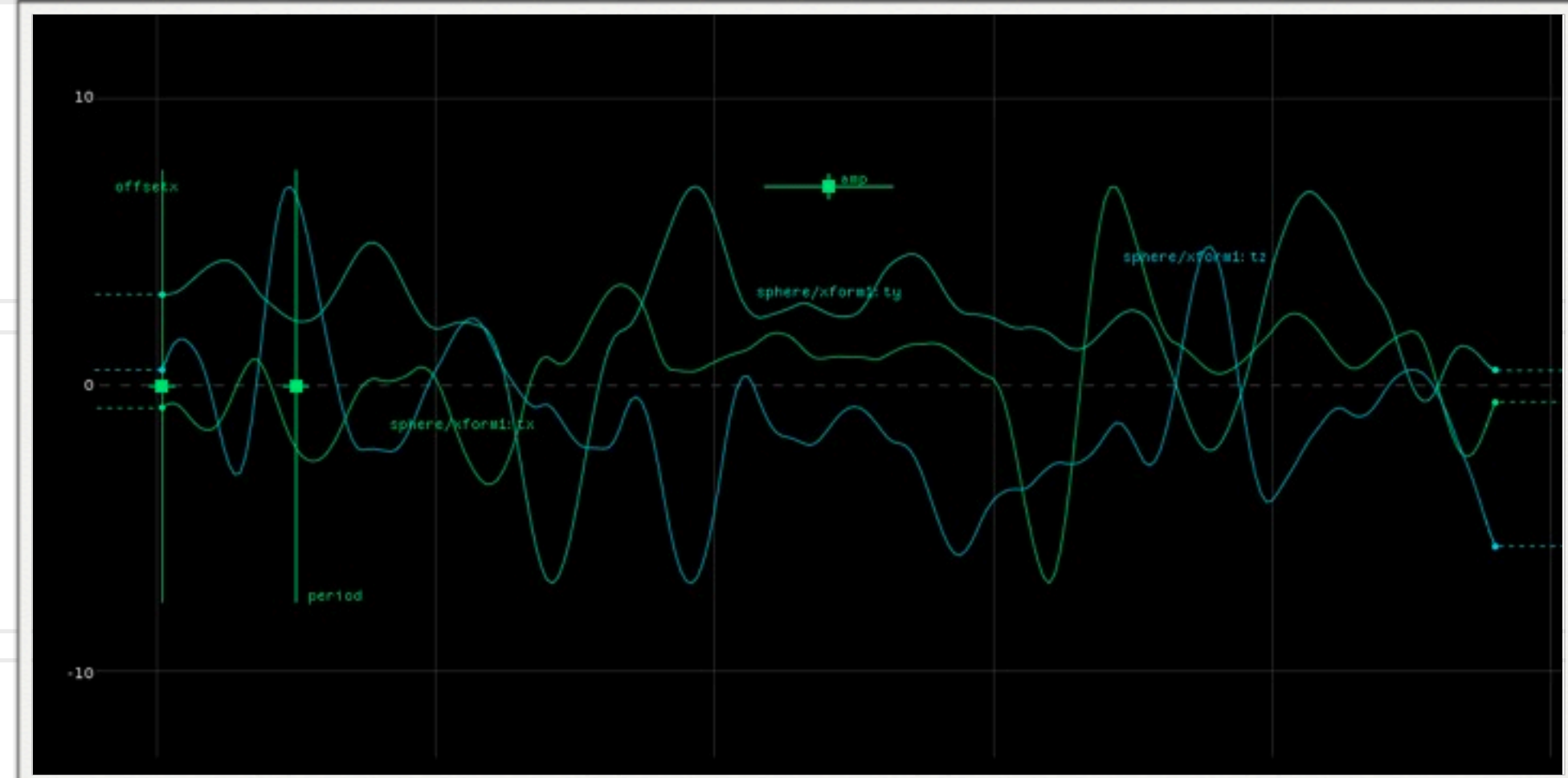


Brownian & Harmonic Summation



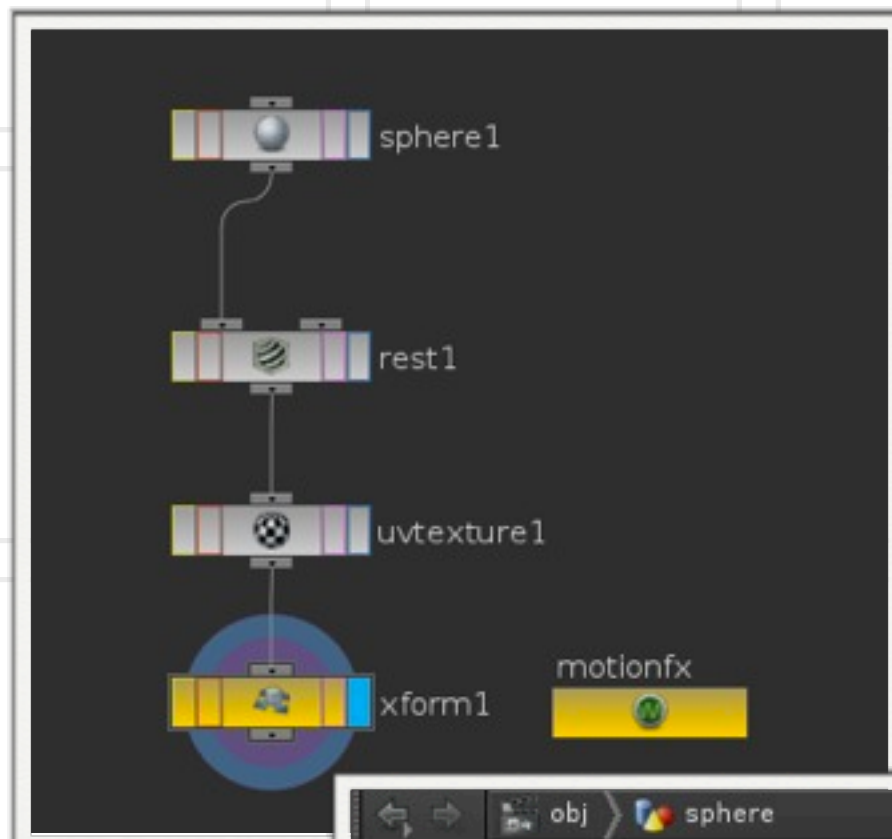
We have not talked about Brownian Noise yet

Notice the soft curves that like to stay at the plateaus



Harmonic Summation gives us nice loopy animation

Adding Weight to Our Motion



Right now the sphere is animating but seems weightless. There is no “real” ease in or ease out because our motion lacks weight. Let us add it now.

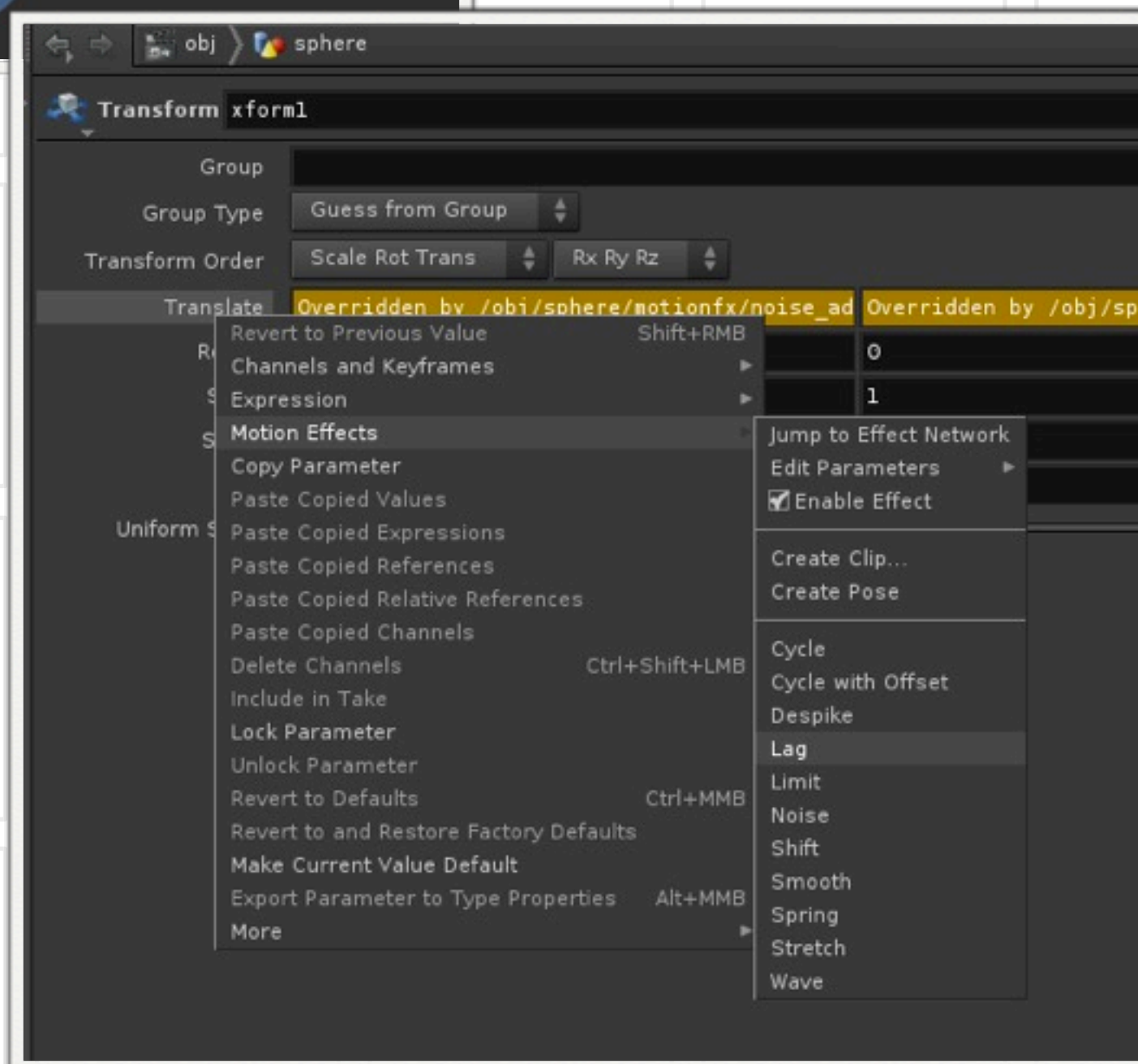
The nodes responsible for motion I will color “yellow” just so we can keep track

Right Click on the Translate parameter of the transform again and select:

- ▶ Motion Effects-->Lag

Dive Inside the MotionFX node (CHOP Net)

- ▶ Continued on next slide...

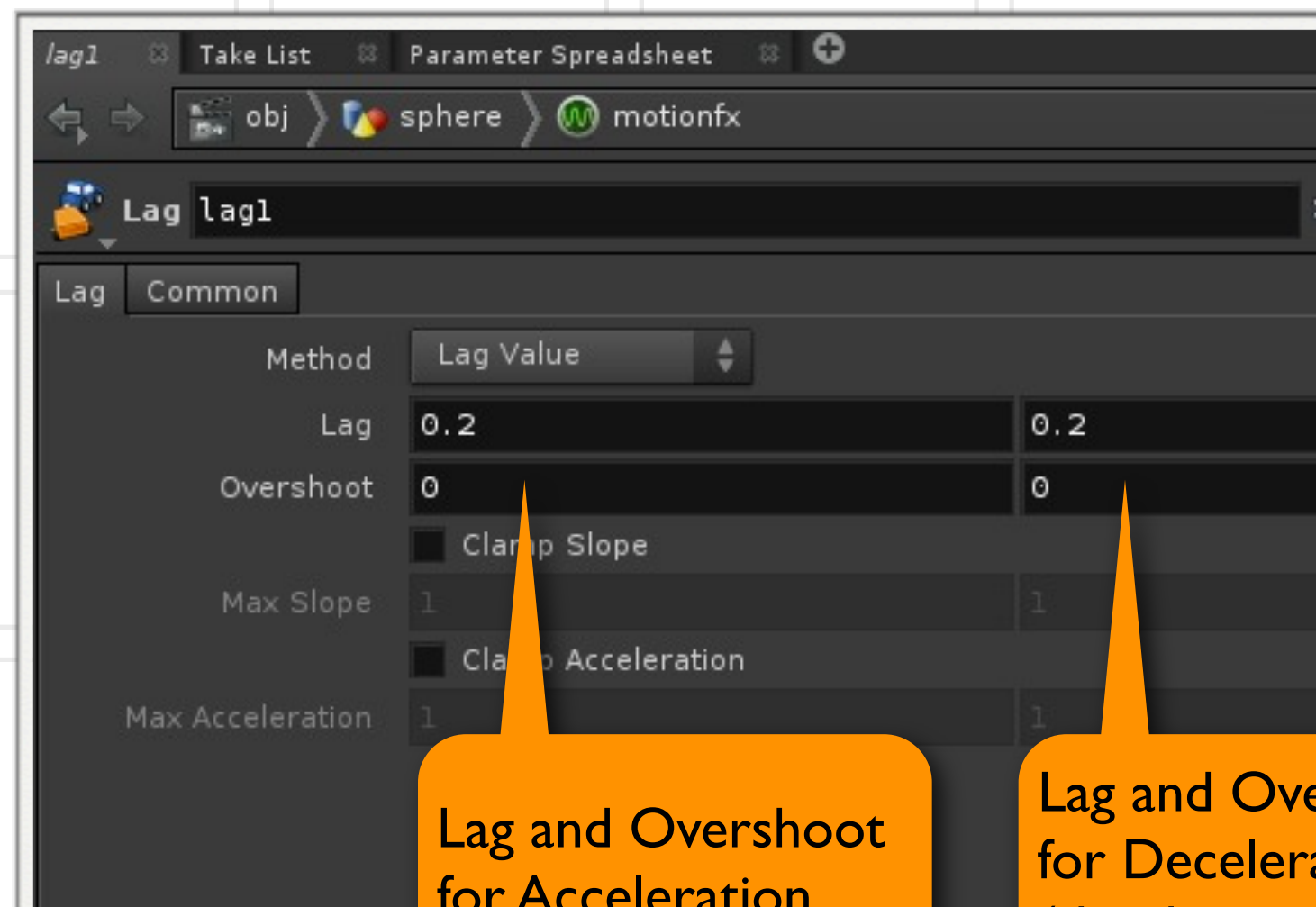
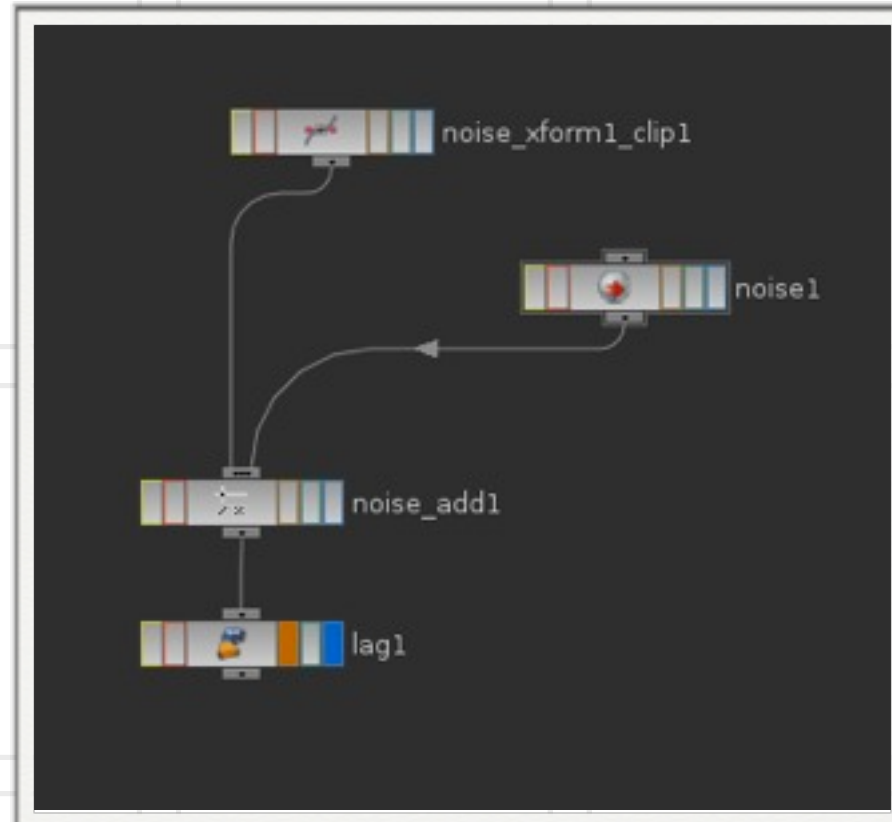


Adding Weight to Our Motion (cont.)

The “Lag” node has been added to the end of the network and the export and visibility flags have been moved to the Lag

The “Lag” node adds lag and overshoot to the animation

- ▶ If you have a heavy object and try to make it move with a given force it will take time to get up to speed compared to a light object - It lags behind
- ▶ Similarly when you try to stop a heavy object from moving in will be slow to stop. It may even “overshoot” the target

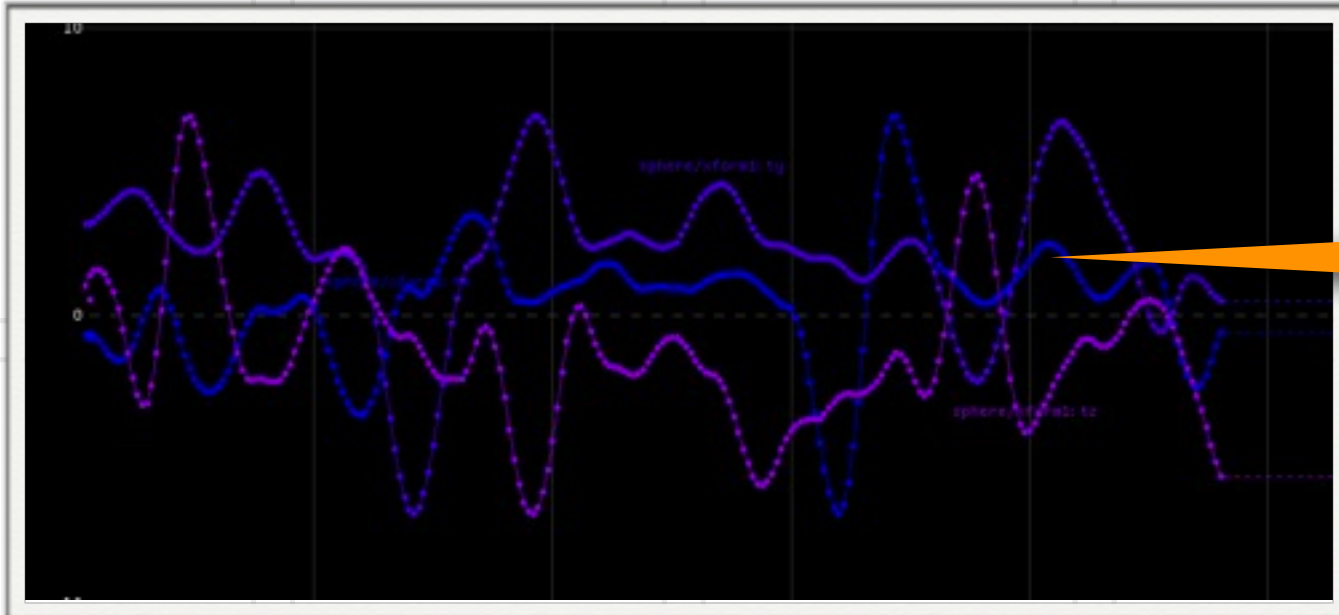


Lag and Overshoot
for Acceleration
(the upward curve)

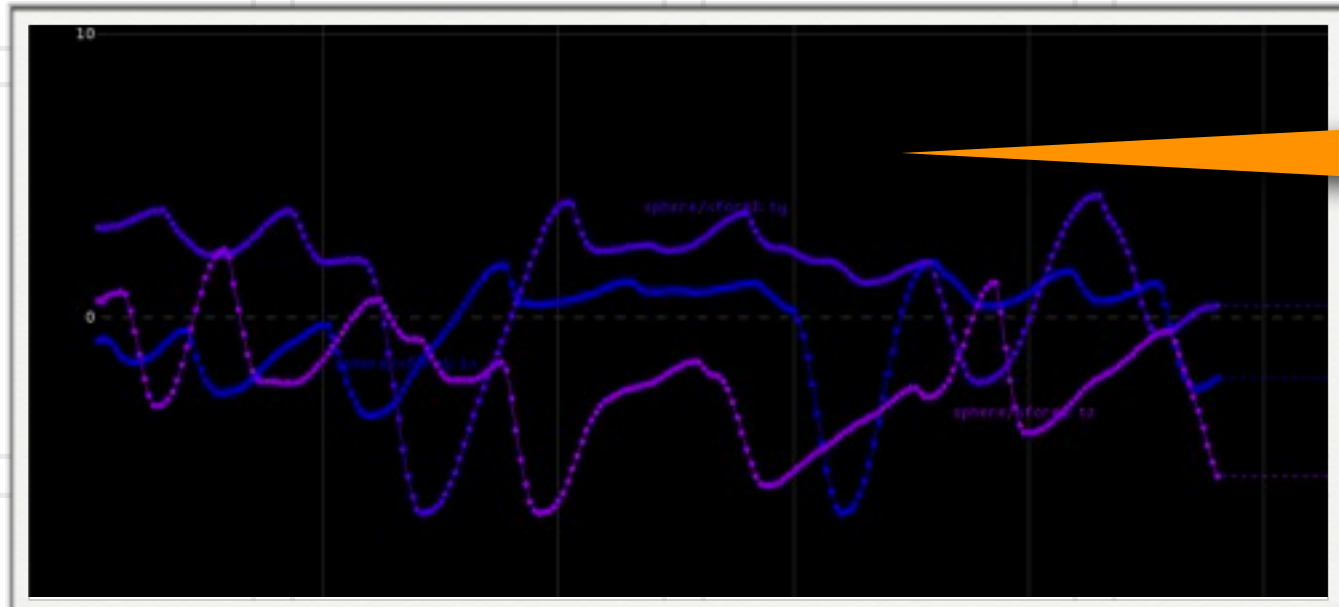
Lag and Overshoot
for Deceleration
(the downward
curve)

**SIDE EFFECTS
SOFTWARE**

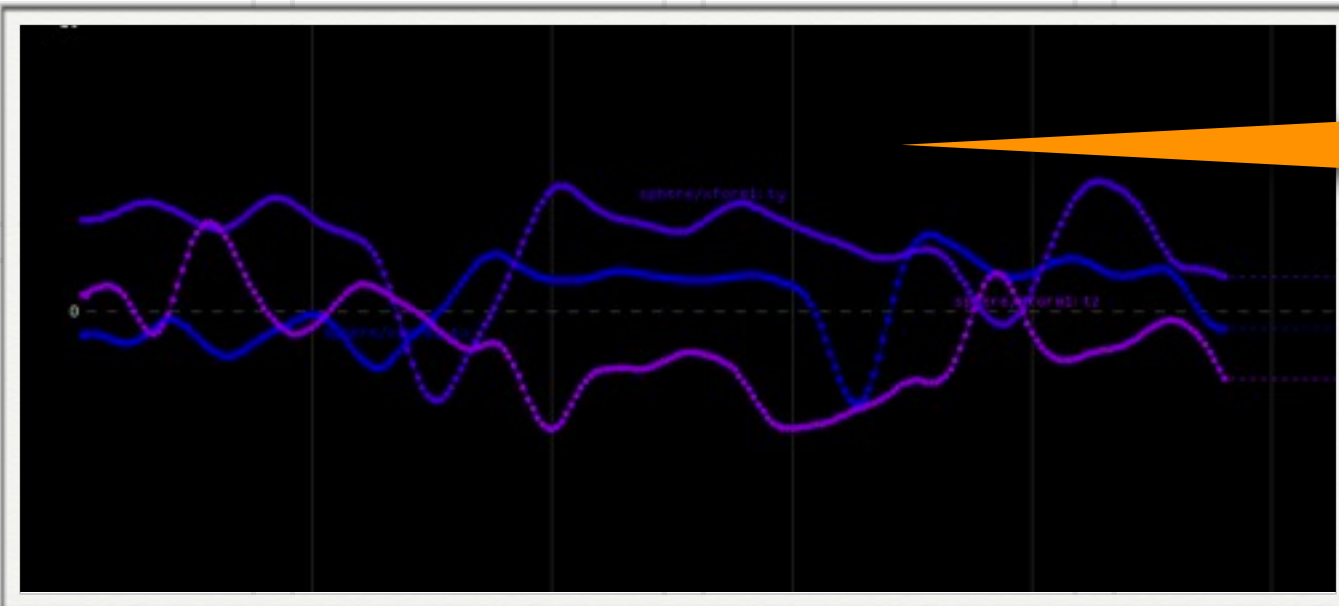
Understanding the Lag and Overshoot Curves



Lag 0,0
Overshoot 0,0

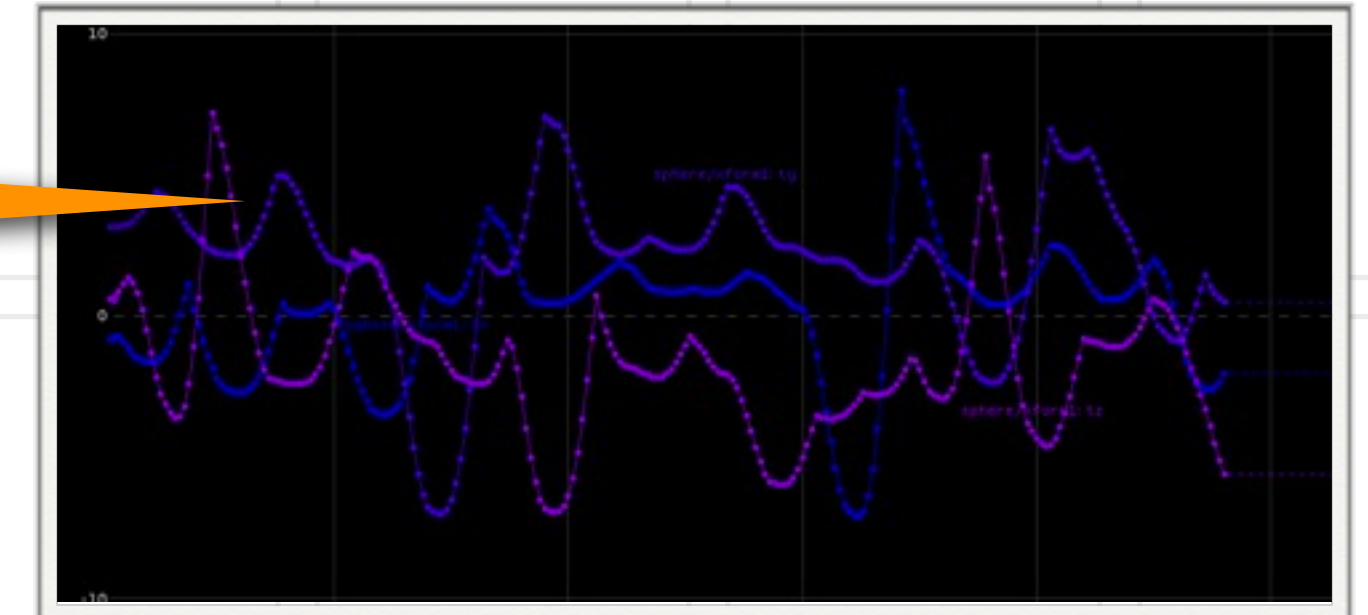


Lag 1,0
Overshoot 0,0
Notice the shallower slope going up. This indicates slower acceleration

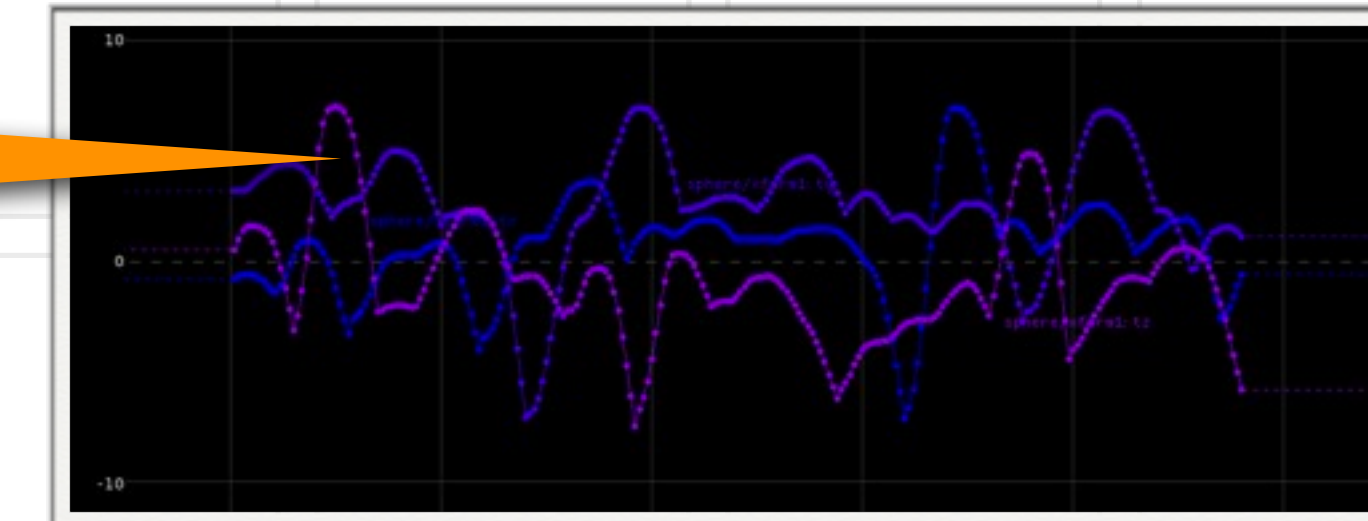


Lag 1,1
Overshoot 0,0
Notice the shallower slope going up and down. This indicates longer duration to a complete stop

Lag 0,0
Overshoot 1,0
Notice the upward plateau is no longer continuous



Lag 0,0
Overshoot 0,1
Notice the downward plateau is no longer continuous



SIDE EFFECTS
SOFTWARE

Continuing the Project

Up until now we have used the Motion FX presets to create a smooth motion with with lag on the main sphere

We will now create our own CHOP Network to define the motion of the “tentacle” spheres that will follow the primary motion of the main sphere

Our goal is not to just let the tentacle spheres follow the main sphere but give each sphere its own lag, and spring. This will allow the animation to have a more organic look

**SIDE EFFECTS
SOFTWARE**

Continuing With Our Project

First. Let us use a Point Wrangle to define Normals

- ▶ `@N = @N; //Initialize the Normals`

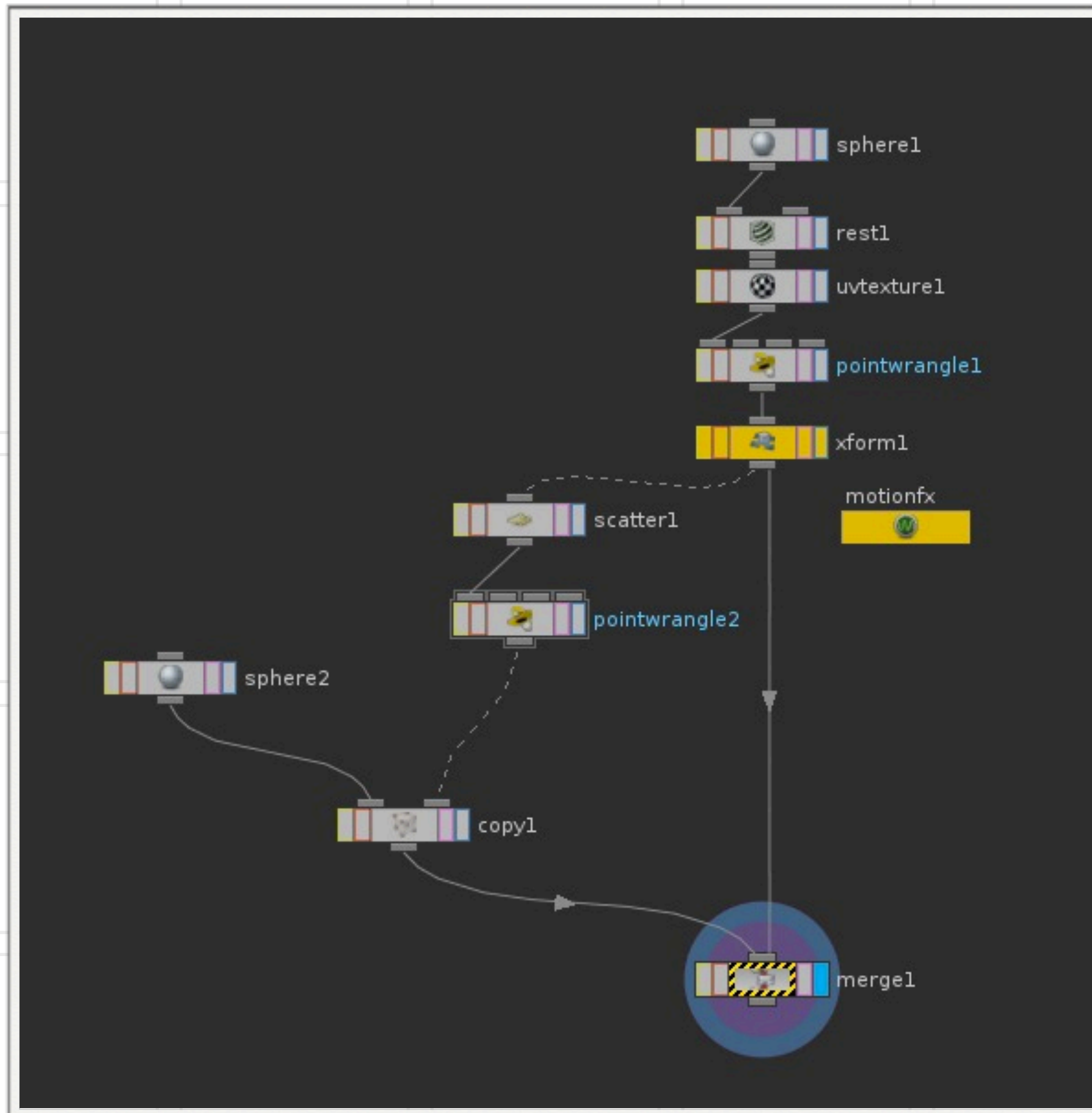
Append a Scatter SOP to the Transform

- ▶ Scatter 50 points

Append another Point Wrangle to the Scatter

- ▶ `@P += @N;`
- ▶ Create a spare float parameter on the point Wrangle name "scale_factor"
- ▶ `@P += @N*ch("scale_factor");`

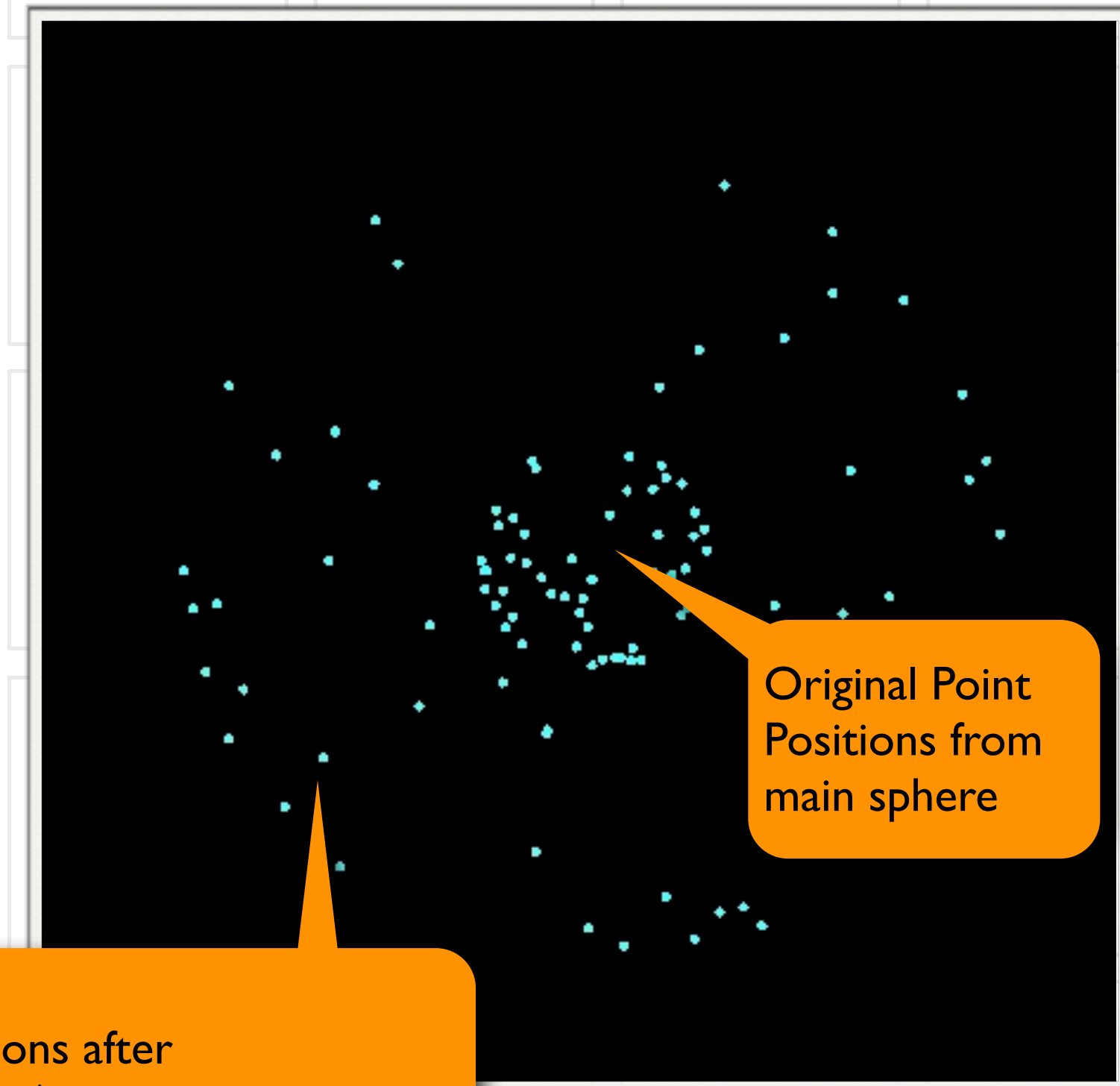
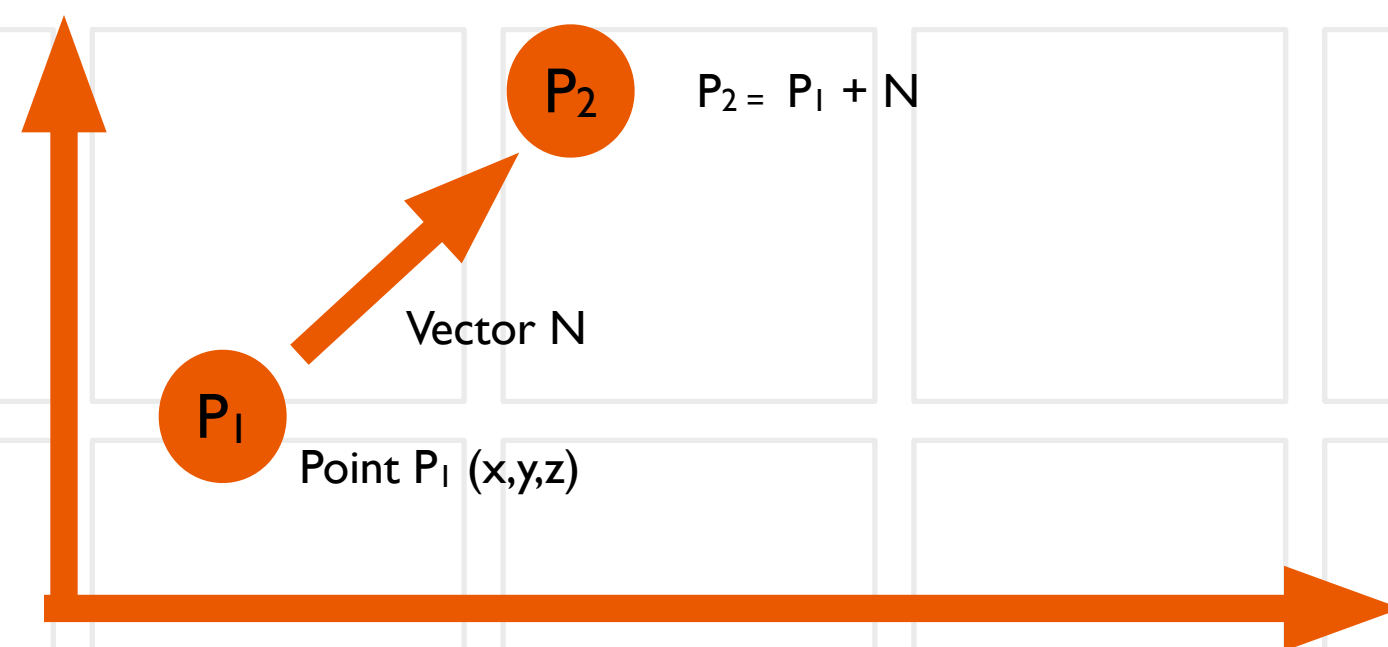
Copy small spheres to the points and merge with original sphere



```
@P += @N*ch("scale_factor");
```

What are we doing with this expression

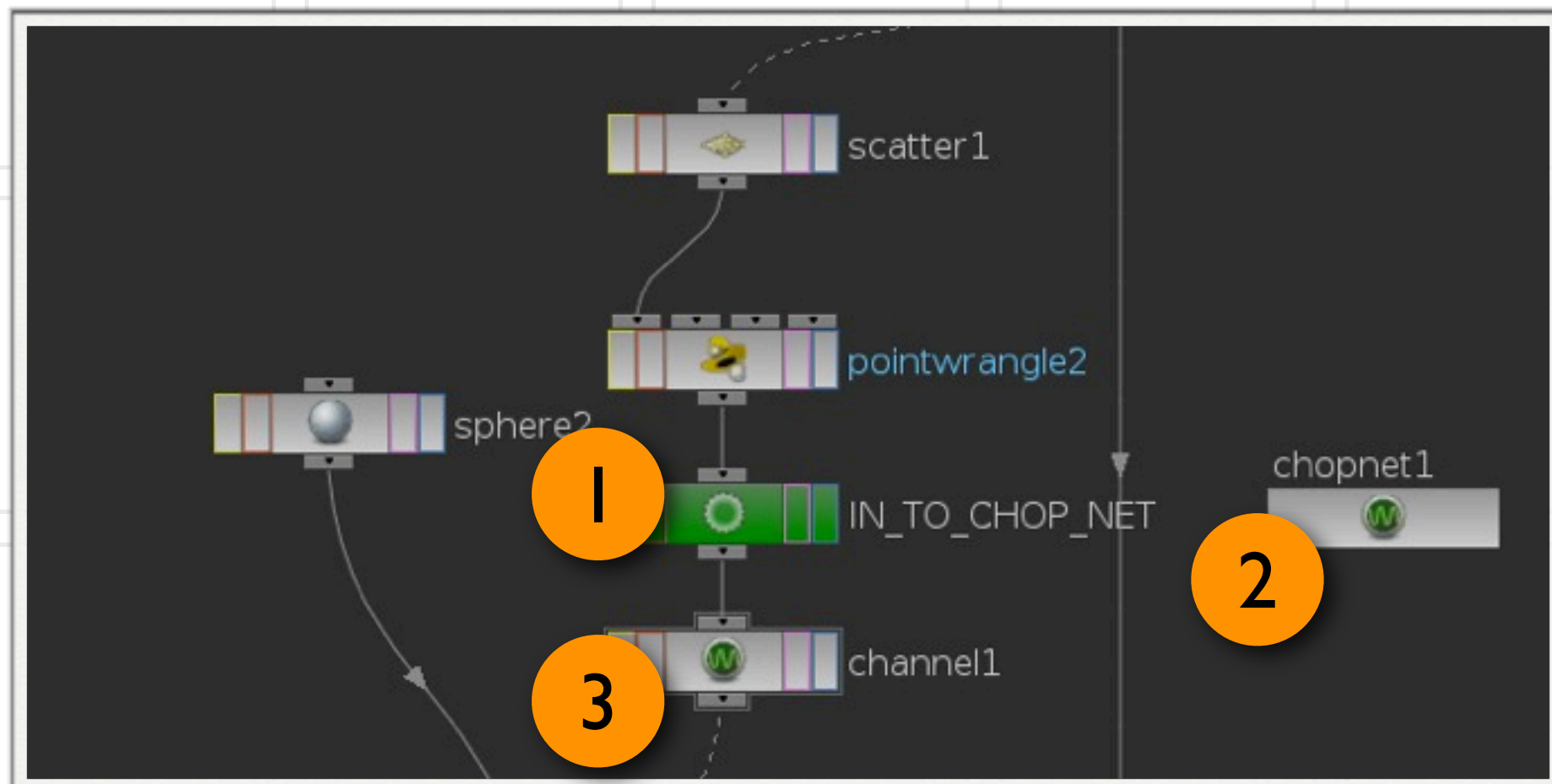
It is perfectly legal to add a vector to a point position



**SIDE EFFECTS
SOFTWARE**

Rolling Your Own CHOP Network

Three Steps to Creating Your own CHOPs



- ▶ 1. Append a NULL to the Node that has the data you want to manipulate with CHOPs
- ▶ 2. Drop down a CHOPNET
- ▶ 3. Append a Channel Node to the NULL to read back the CHOPNET manipulated data

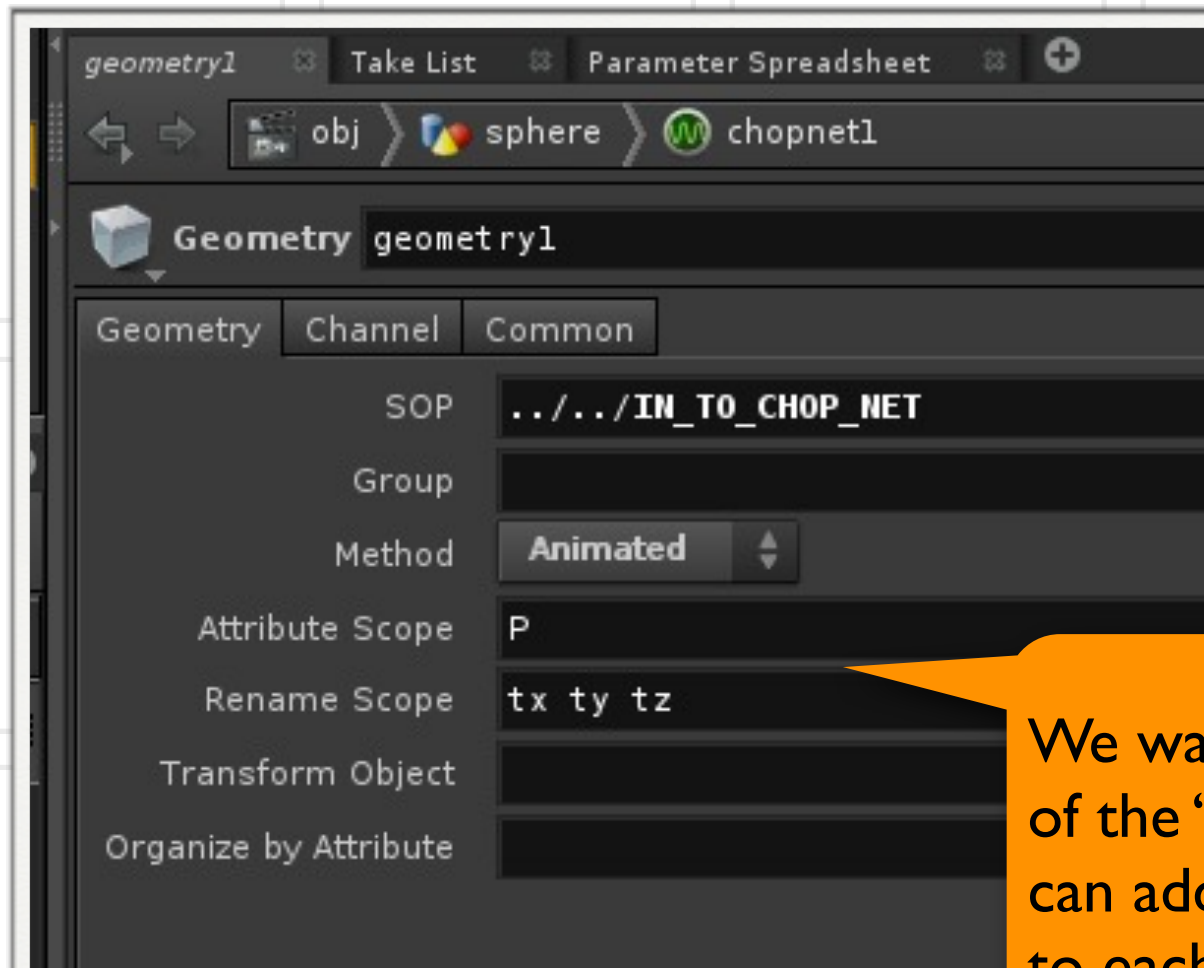
NOTE: The network will error out because you are not pointing to any data inside the CHOPNET Yet.

Rolling Your Own CHOP Network (cont.)

Dive Inside the CHOPNET

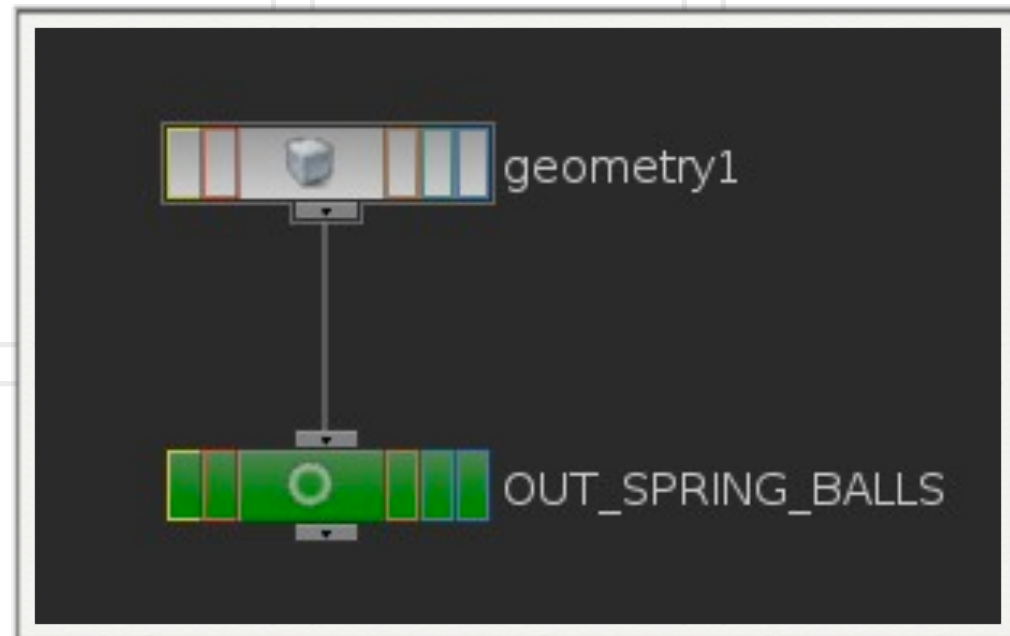
Drop Down a Geometry CHOP - The Geometry CHOP uses a geometry object to choose a SOP from which the channels will be created. The channels are created from the point attributes of a SOP, such as the X, Y and Z of the point position.

- ▶ The SOP should be pointing to the NULL you created earlier. The Geometry CHOP will fetch the data.
- ▶ Set the Method to “Animated”
 - ▶ Static - Creates one channel for each attribute, and all points use this channel (the first point resides at index 0, the next at 1, and so on). The length of the channels will be the number of points fetched.
 - ▶ Animated - Creates one channel per attribute per point. The channels show the animation of each point's position/attribute values.



We want to fetch the position data of the “tentacle” spheres so we can add different lags and springs to each individual sphere

Completing the Cycle



Right now the network is still has an error

Let us complete the CHOP cycle so we do not have errors

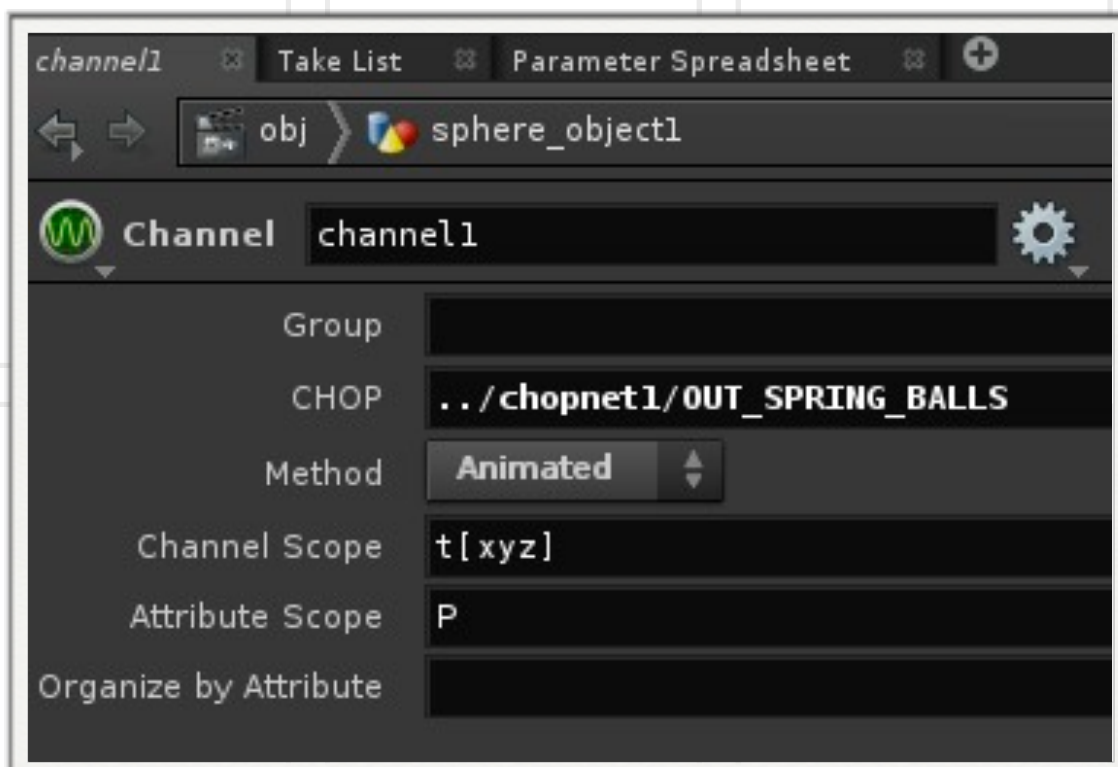
Append a NULL to the Geometry CHOP

- ▶ Name it “OUT_SPRING_BALLS”

Go Up to the Geometry Level

In the Channel SOP

- ▶ CHOP ../chopnet1/OUT_SPRING_BALLS



Rolling Your Own CHOP Network (cont.)

Attribute Scope - What data do we want to manipulate.

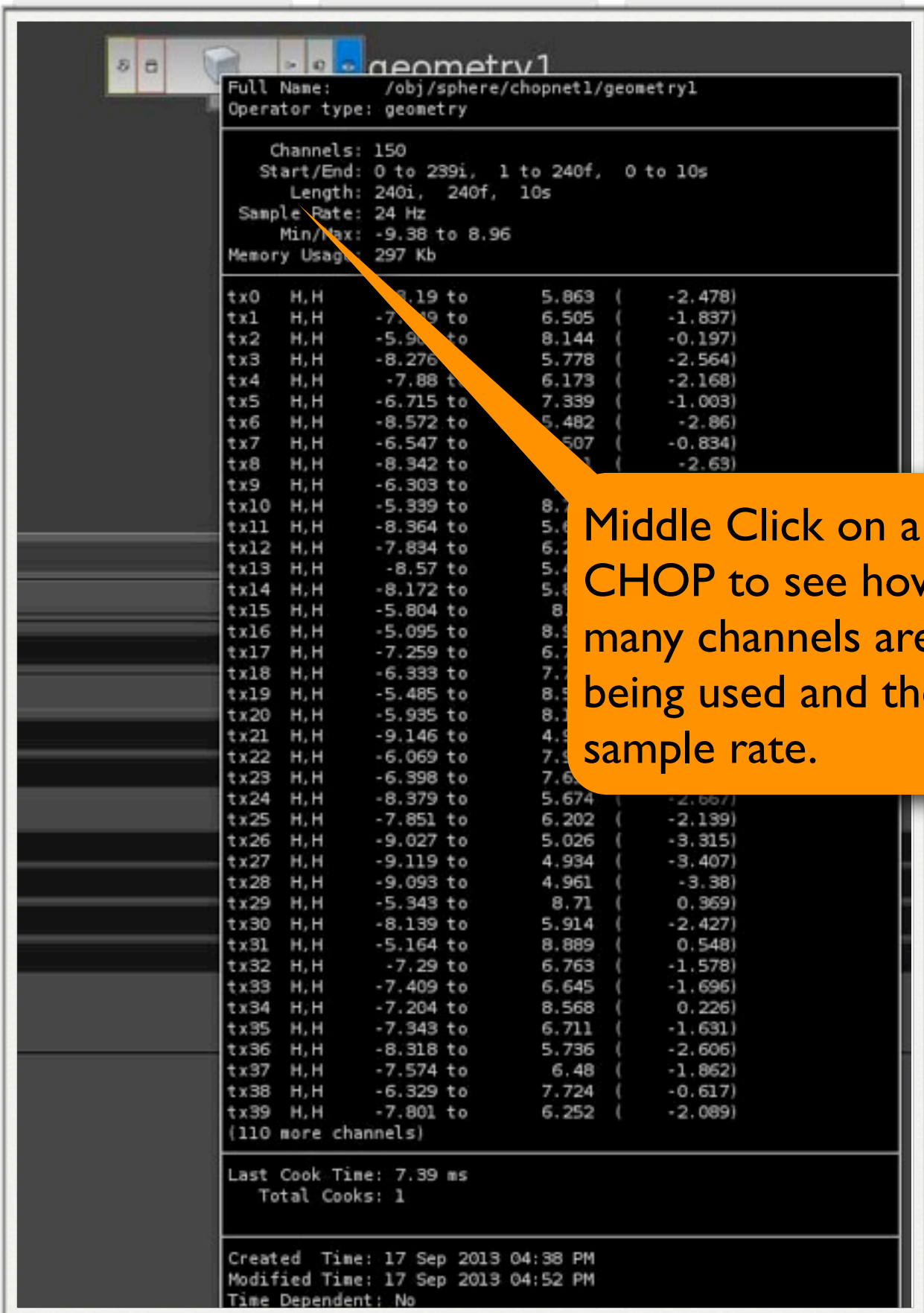
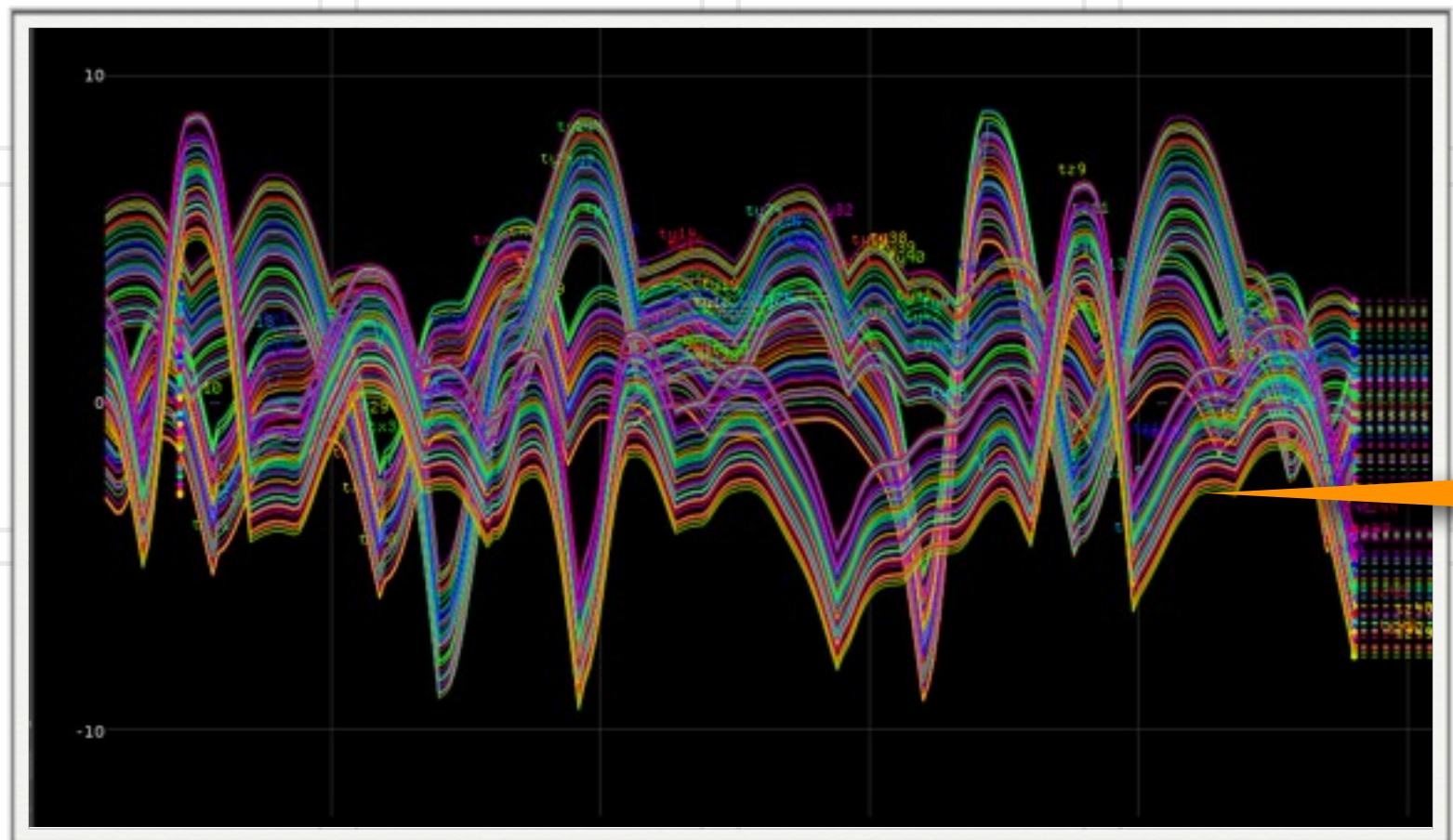
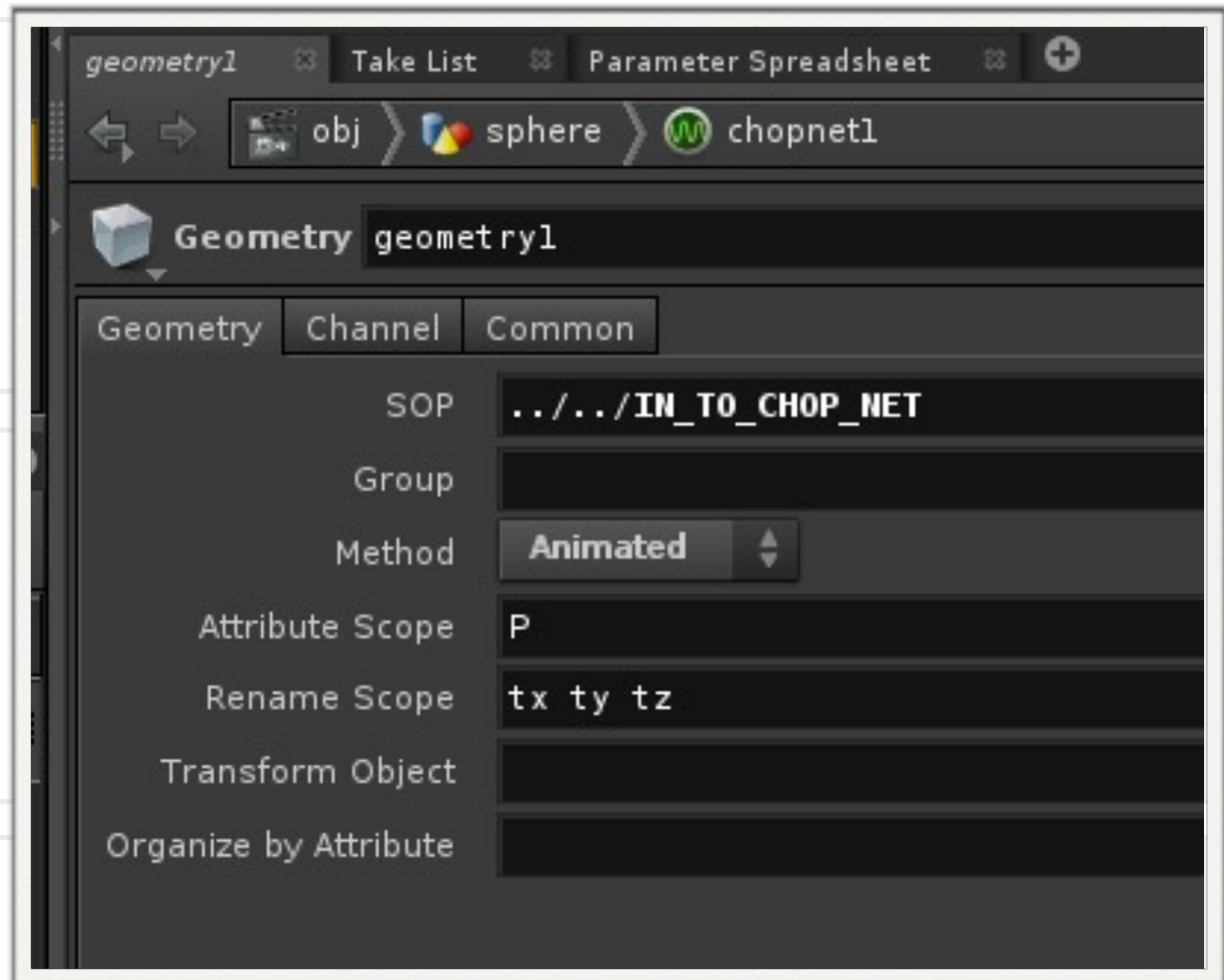
- In our case position “P”, later we can play with Rotation

Rename Scope - What will we name each channel

- tx,ty,tz

Motion View Displaying all tx,ty, and tz data

Middle Click on a CHOP to see how many channels are being used and the sample rate.



Help Card for Geometry CHOP

Parameters

Geometry

SOP	Retrieves the geometry from the SOP pointed to by this path.												
Group	Fetch only the points within this point group. If blank, all points are fetched.												
Method	The geometry fetch method. <div><div>Static</div><div>Creates one channel for each attribute, and all points use this channel (the first point resides at index 0, the next at 1, and so on). The length of the channels will be the number of points fetched.</div><div>Animated</div><div>Creates one channel per attribute per point. The channels show the animation of each point's position/attribute value over time.</div></div>												
Attribute Scope	<div>A string list of attributes to fetch from the SOP. The common attributes are:</div> <table><tr><td>P</td><td>point position (X, Y, Z) - 3 values.</td></tr><tr><td>Pw</td><td>point weight with a value of 1.</td></tr><tr><td>Cd</td><td>point color (red, green, blue) - 3 values.</td></tr><tr><td>Alpha</td><td>point alpha with a value of 1.</td></tr><tr><td>N</td><td>point normal value (X, Y, Z) - 3 values.</td></tr><tr><td>uv</td><td>point texture coordinates (U,V,W) - 3 values.</td></tr></table>	P	point position (X, Y, Z) - 3 values.	Pw	point weight with a value of 1.	Cd	point color (red, green, blue) - 3 values.	Alpha	point alpha with a value of 1.	N	point normal value (X, Y, Z) - 3 values.	uv	point texture coordinates (U,V,W) - 3 values.
P	point position (X, Y, Z) - 3 values.												
Pw	point weight with a value of 1.												
Cd	point color (red, green, blue) - 3 values.												
Alpha	point alpha with a value of 1.												
N	point normal value (X, Y, Z) - 3 values.												
uv	point texture coordinates (U,V,W) - 3 values.												
Rename Scope	The names to use for the newly fetched attributes. There must be one name per attribute, so an attribute scope of "P" should have a rename scope with three names (like tx ty tz).												
Transform Object	If a transform object is specified, the point values will be represented relative to that object's origin and rotation.												
Organize by Attribute	(Applies to 'Animated Method' only) Will rebuild the fetched channels by the value of this attribute. A common example is the 'id' attribute found in particles. A channel is built for each unique id since the number of points may vary.												

Common attributes to fetch into CHOPs

In CHOPs each component of a position or vector becomes a channel therefore we must rename the scope as an example from P to tx, ty, tz

Adding a Little Noise to the Spheres

Append a Math CHOP to the Geometry CHOP

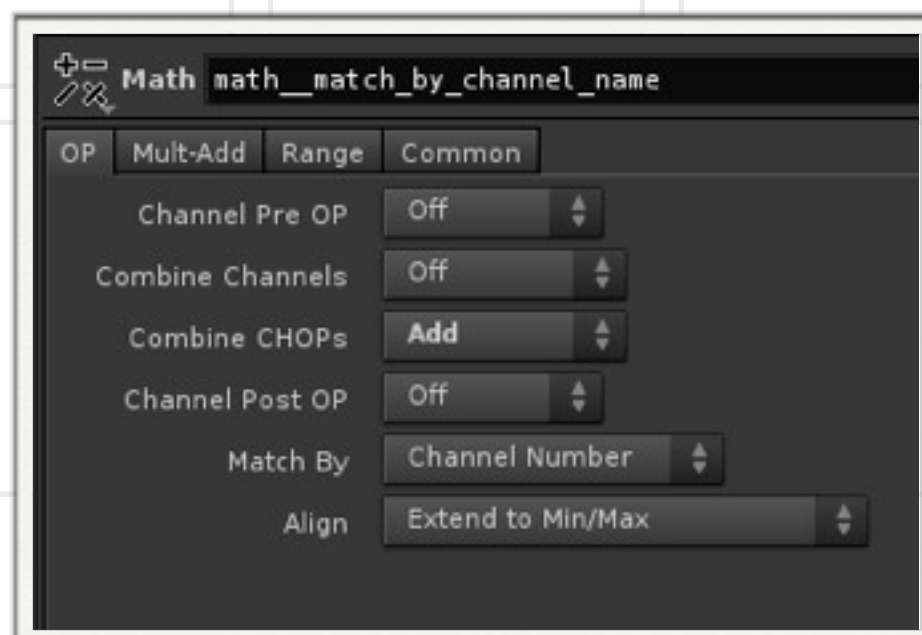
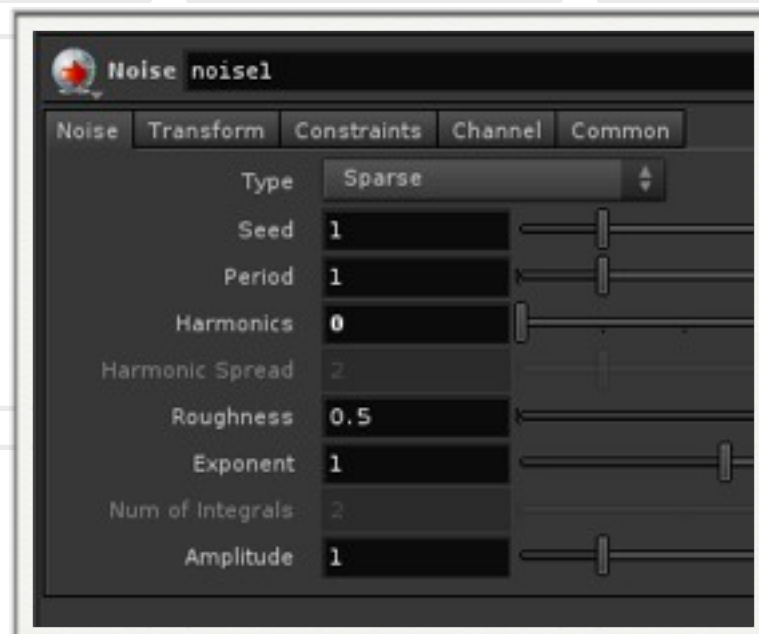
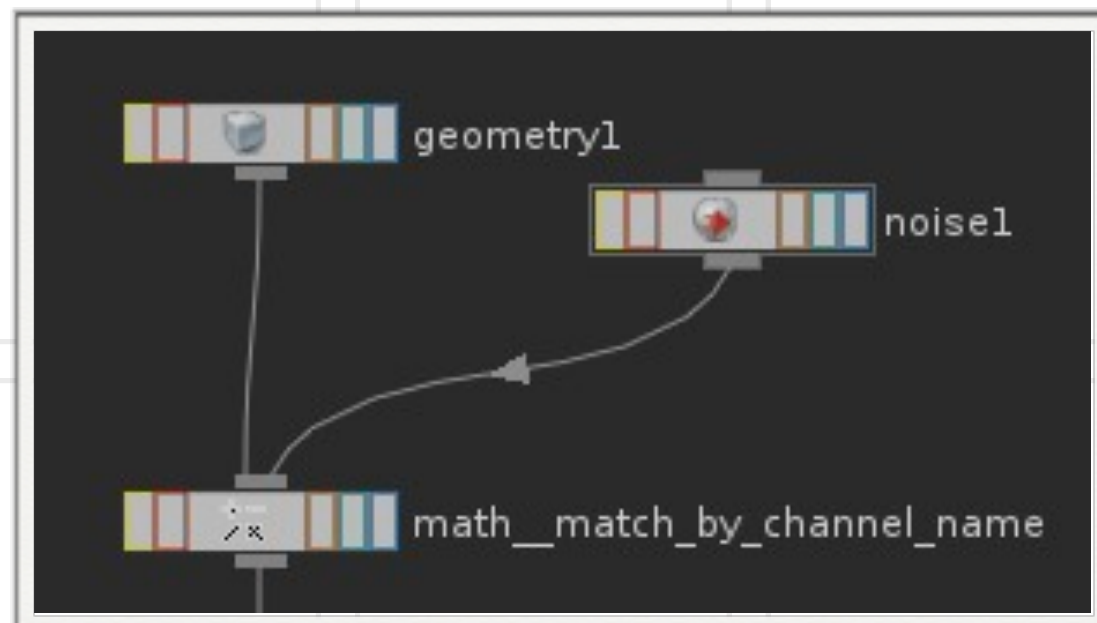
Drop Down a Noise CHOP and wire it to the Math CHOP

On the Noise CHOP

- ▶ Type - Sparse
- ▶ Seed 1
- ▶ Harmonics 0 - This will give us a smooth noise

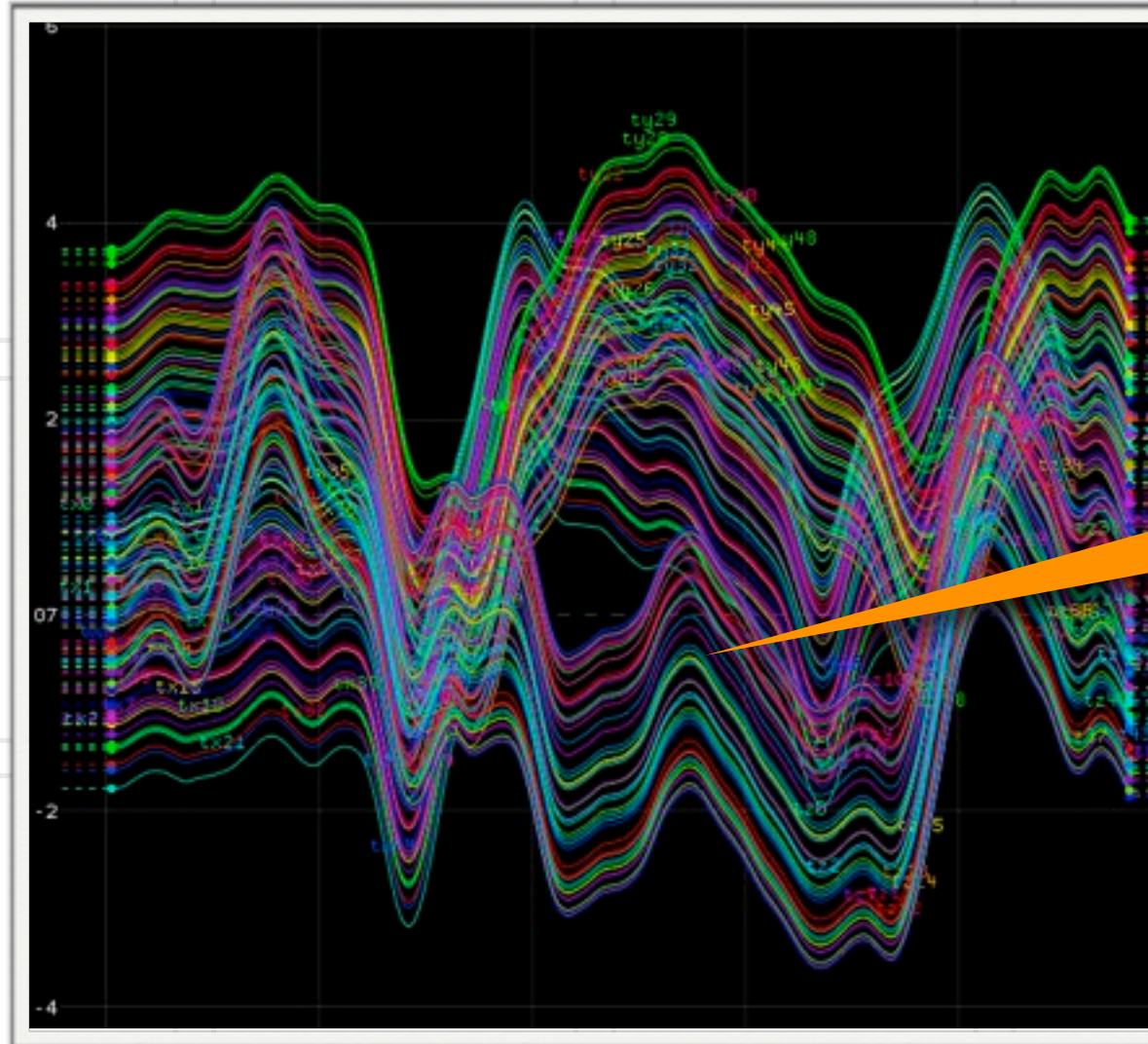
On the Math Node

- ▶ Combine CHOPs - Set to "Add"
- ▶ We want to add the Noise Channels to the Geometry Channels



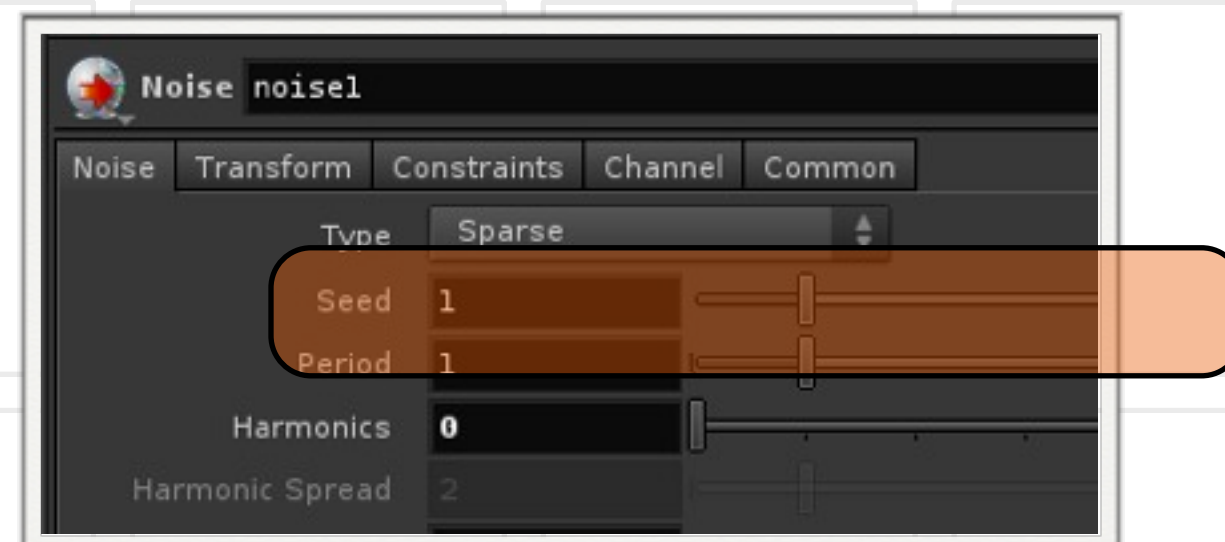
SIDE EFFECTS
SOFTWARE

Adding a Little Noise to the Spheres (cont.)



All Position have equal noise applied. We want each sphere to have different noises applied

Unfortunately this creates a motion that is identical for each sphere. We can see this in the Motion View



We can fix this. In the Noise CHOP there is a “seed” parameter. We need to be able to vary the seed value per channel. If we only knew the local variables

Continued on next slide

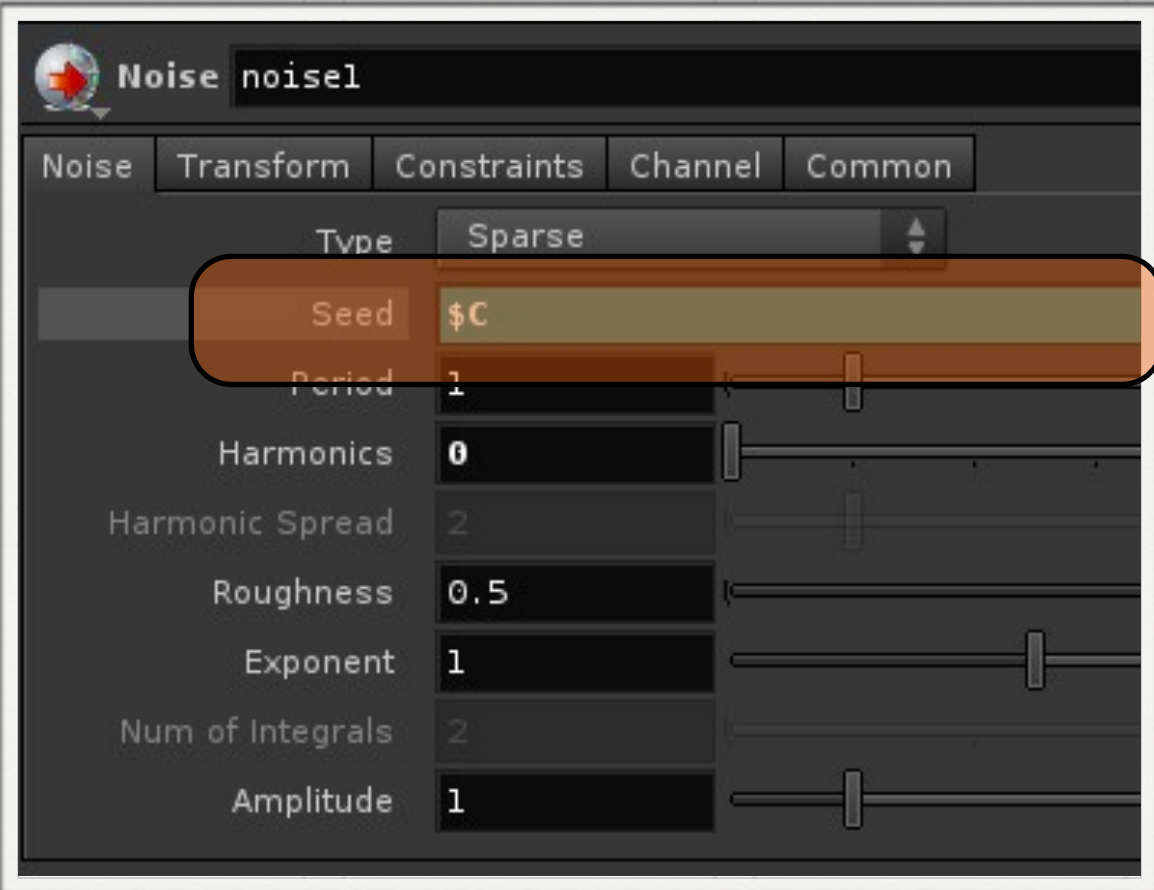
Adding a Little Noise to the Spheres (cont.)

Local variables

I	The current index.
C	The current channel (0 to NC-1).
NC	The total number of channels.
S,E	The start and end indices of the noise curve.

Pulling up the help card for Noise we can find several local variables

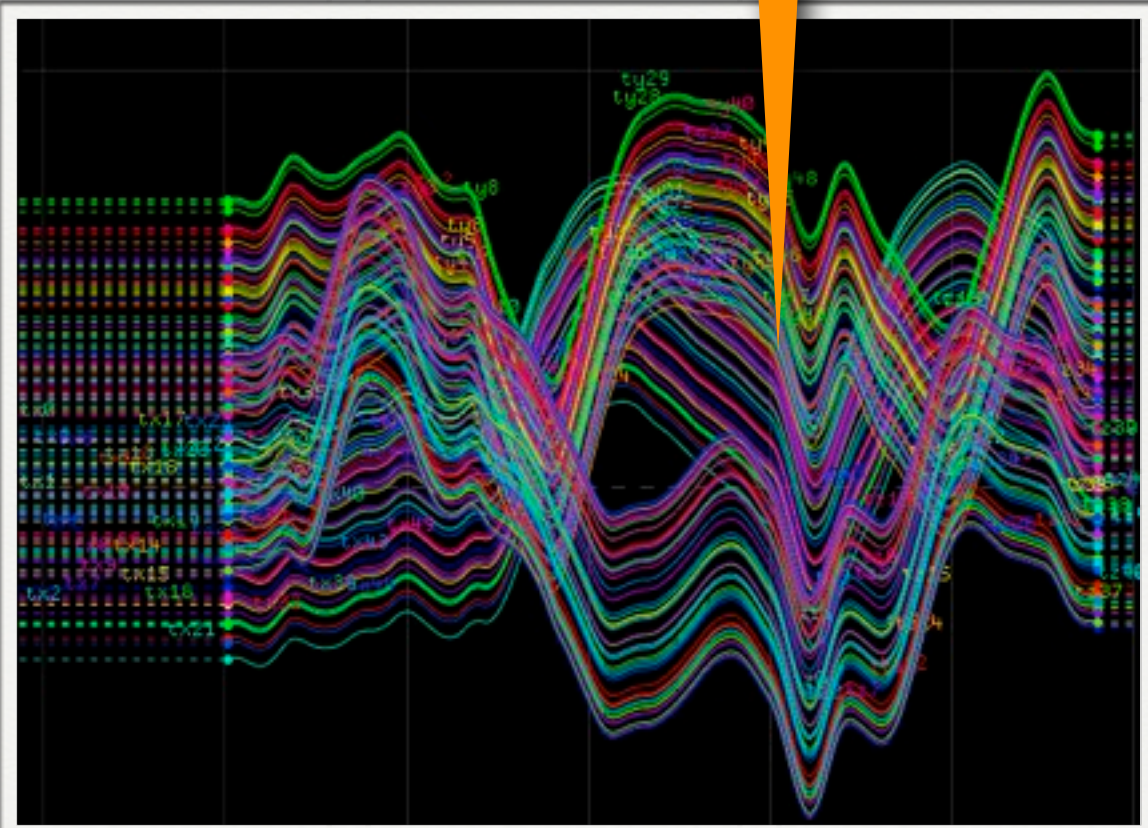
No Real Difference - Even with varying the seed value with \$C there is not much of a difference. There is still one thing to do



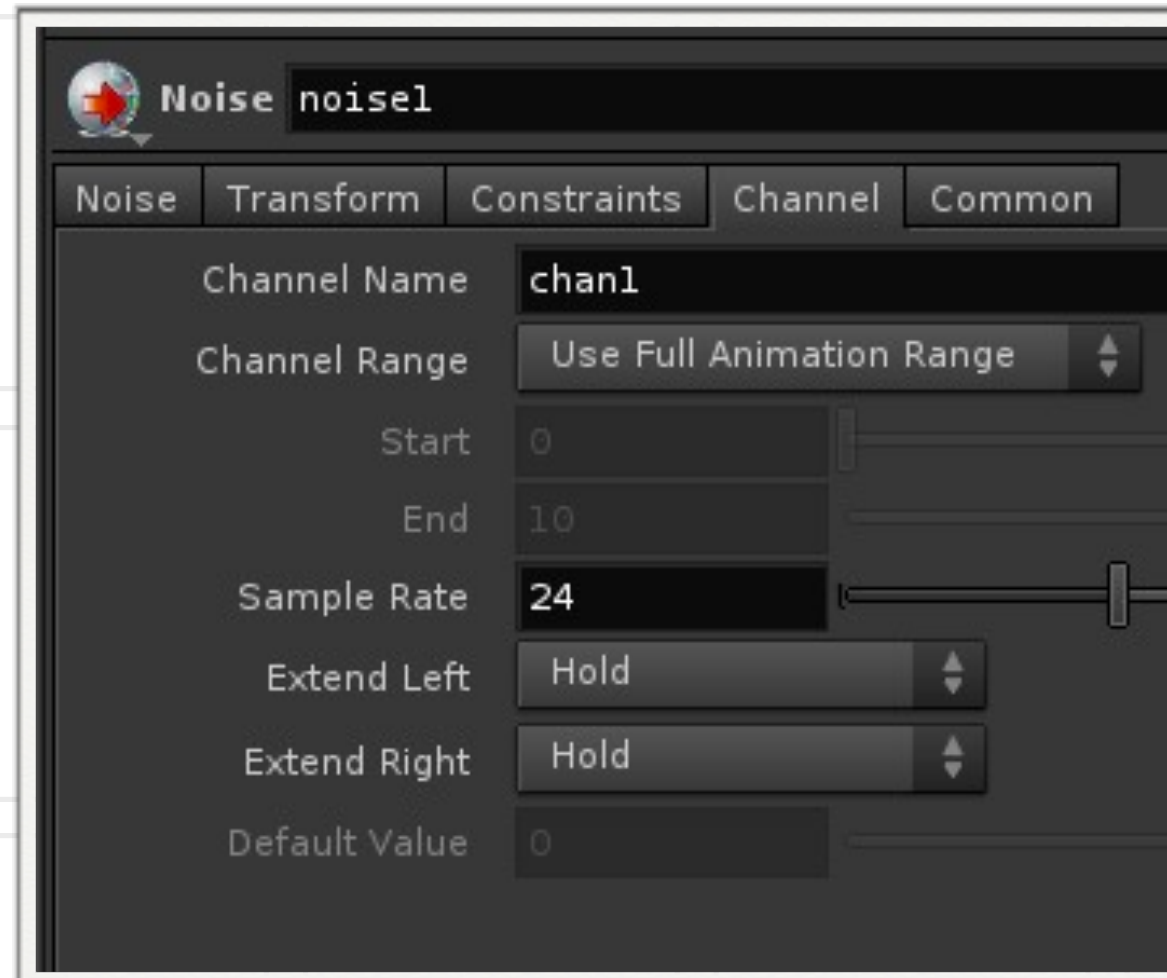
Let us use the Current Channel local variable

In the seed parameter - \$C

Continued on next slide...



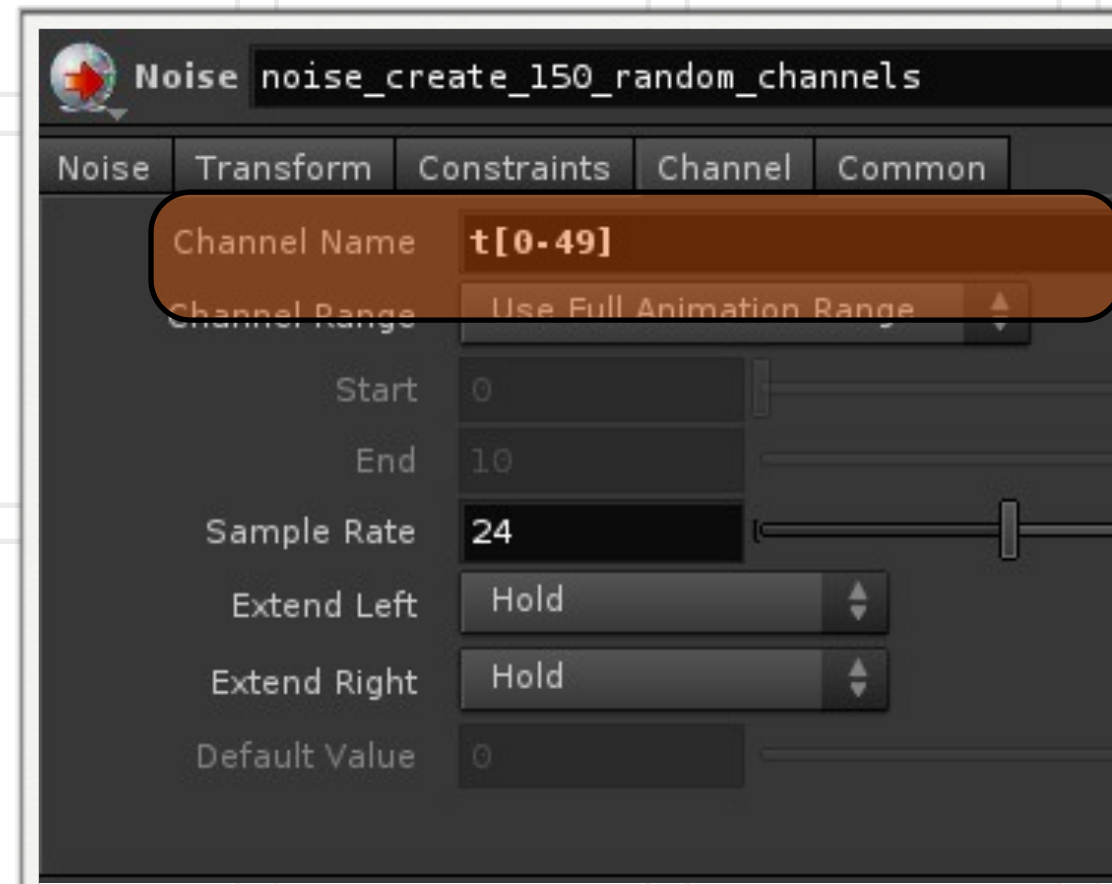
Adding a Little Noise to the Spheres (cont.)



In the Noise CHOP switch to the Channel Tab

Currently the Channel Name is set to “chan1”

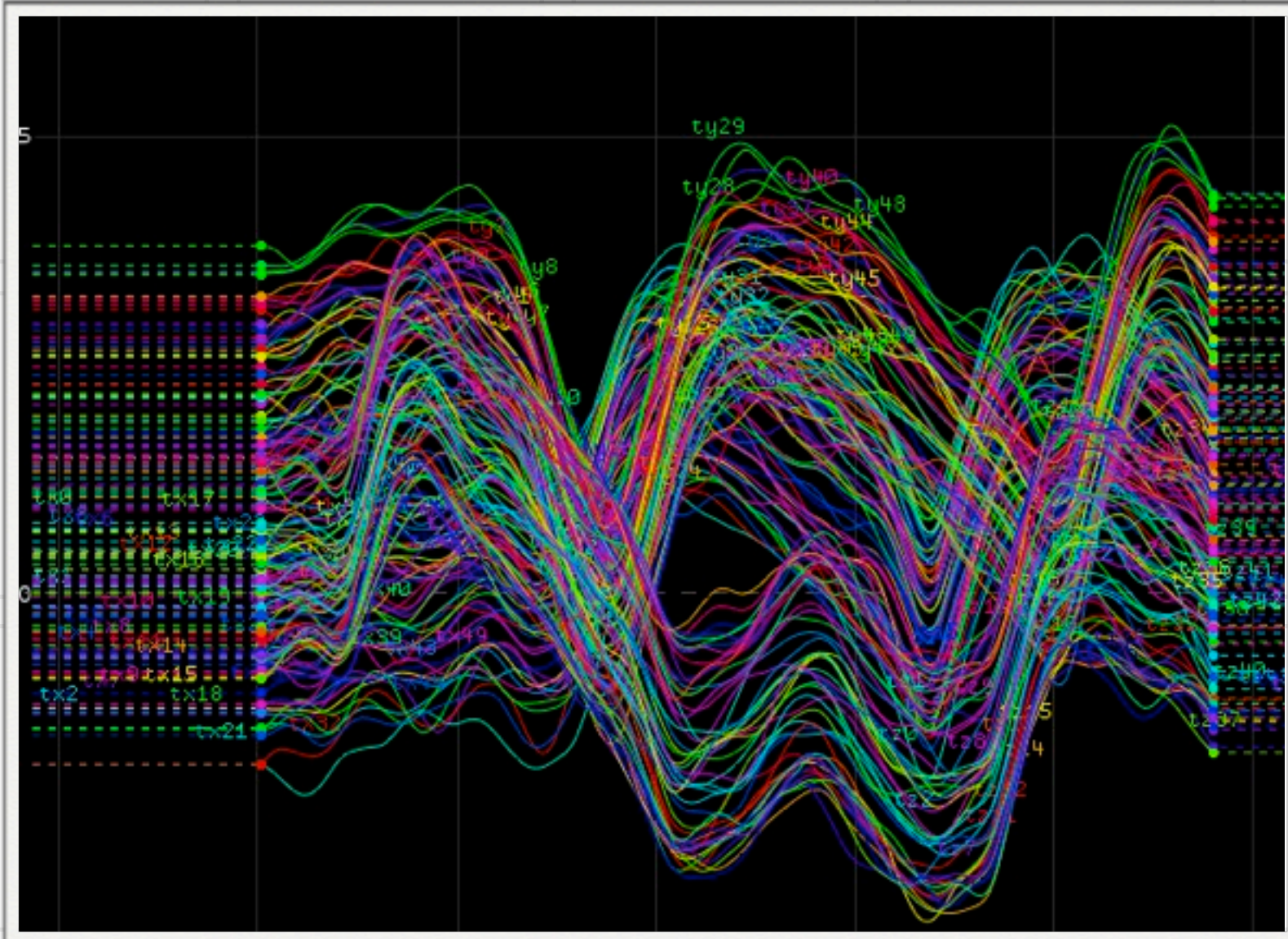
This mean it will only create noise for the first channel



Change the Channel Name Parameter - t[0-49]

Now it knows to create noise for a total of 50 channels

Adding a Little Noise to the Spheres (cont.)



Now with Channel Name set to `t[0-49]`

Each sphere has a unique animation

A more formal way to write this would be `t[xyz][0-49]`

How did we know to
write `t[xyz][0-49]`?

How did we know to write t[xyz][0-49]?



Drop down a Rename CHOP (not attached to anything) and open the help card

Sample Naming Patterns

The *From* and *To* parameters are any channel name creation patterns like: `chan[1-9:2]` or `t[xyz]` . The *From* parameter also allows matching patterns like `chan*` and `ch?` .
For example:

Simple matching assuming an input of `chan1 chan2 chan3`:

From	To	Produces
<code>chan1 chan2 chan3</code>	<code>tx ty tz</code>	<code>tx ty tz</code>
<code>chan[1-3]</code>	<code>t[xyz]</code>	<code>tx ty tz</code>
<code>chan[1-3]</code>	<code>tx ty tz rx</code>	<code>tx ty tz</code>
<code>chan[1-2]</code>	<code>tx ty</code>	<code>tx ty chan3</code>

Simple matching assuming an input of `tx ty tz`:

<code>t[xyz]</code>	<code>[XYZ]trans</code>	<code>Xtrans Ytrans Ztrans</code>
---------------------	-------------------------	-----------------------------------

Change channels that start with "bob" to channels that start with "carol":

<code>bob*</code>	<code>carol*</code>
-------------------	---------------------

Replace channels with "hand" in the middle to channels with "foot" in same place:

<code>*hand*</code>	<code>*foot*</code>
---------------------	---------------------

Swap words in a name, where the words are separated by : `table>>tr>>td>>`**``

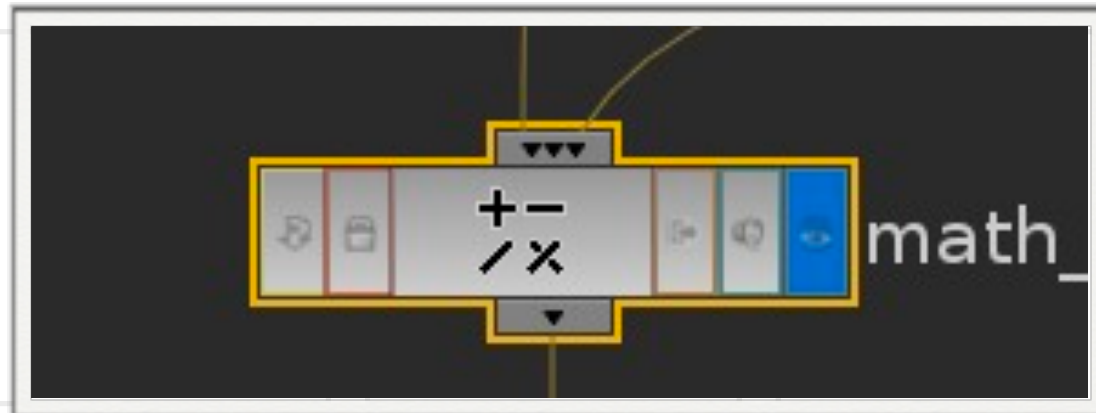
Wildcard matching, assuming an input of `ch1 ch2 ch3`:

<code>ch?</code>	<code>t?</code>	<code>t1 t2 t3</code>
<code>*</code>	<code>op:*</code>	<code>op:ch1 op:ch2 op:ch3</code>
<code>*h*</code>	<code>*han*</code>	<code>chan1 chan2 chan3</code>
<code>*[1-3:2]</code>	<code>foot[11-13:2]</code>	<code>foot11 ch2 foot13</code>
<code>*</code>	<code>chan[3,1,2]</code>	<code>chan3 chan1 chan2</code>

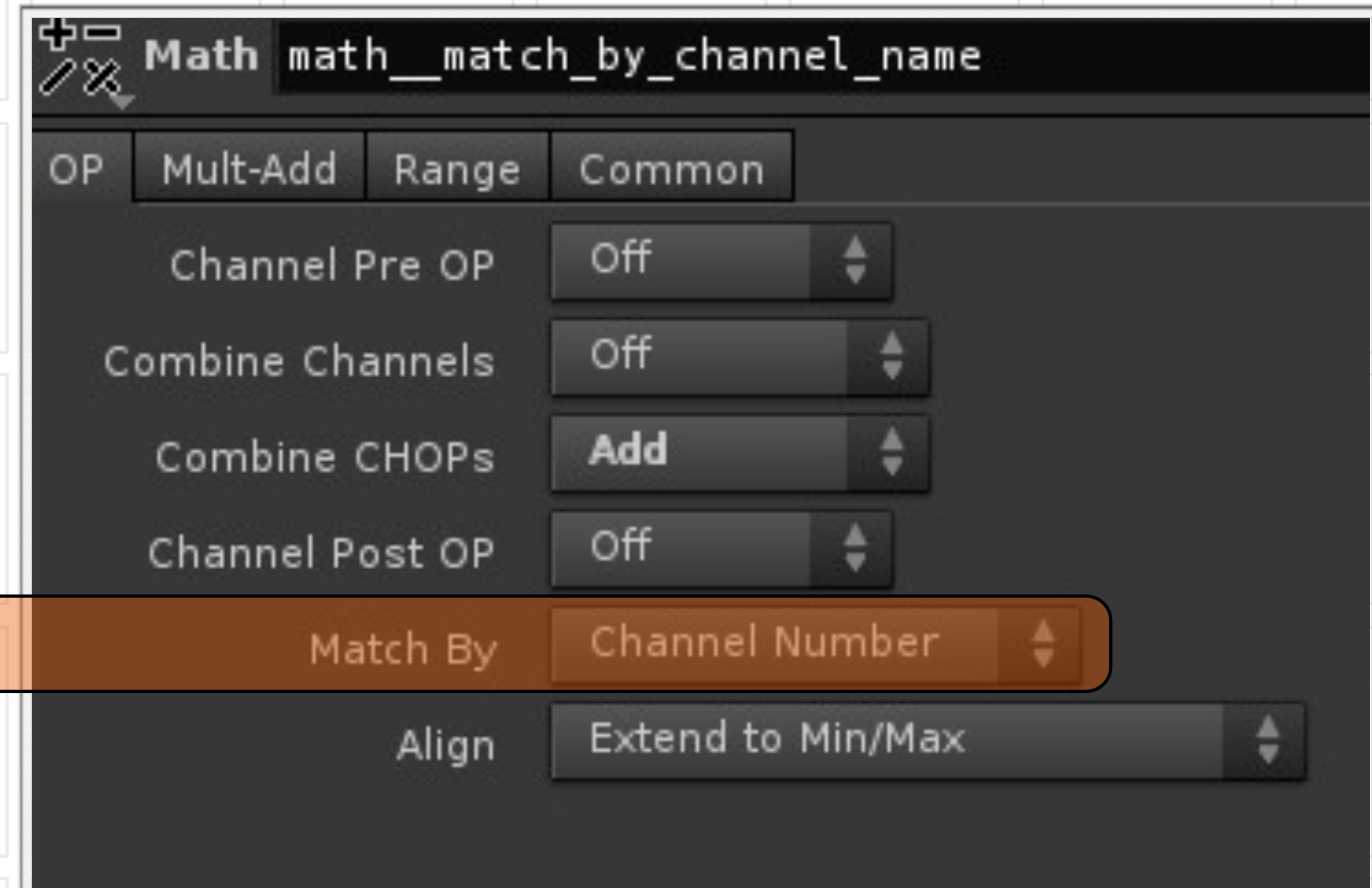
Parameters

The Rename Help card will show you all your channel name options

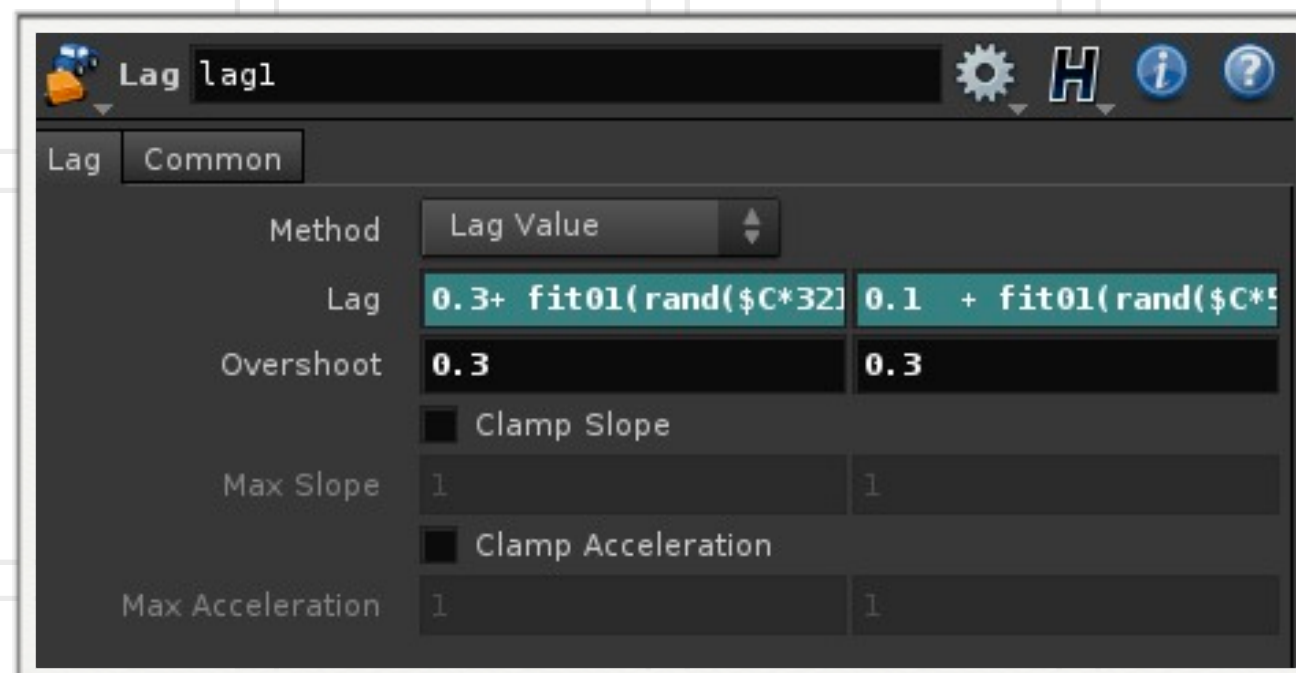
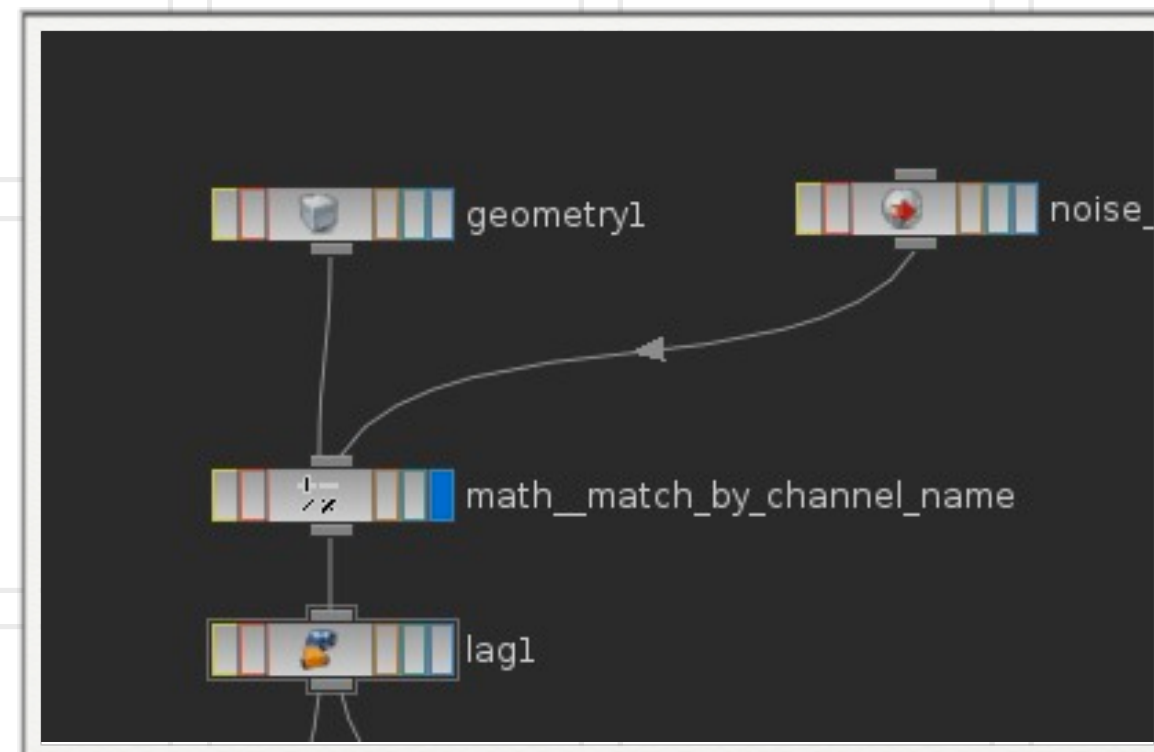
How does the Math CHOP know to add T1 of Noise to T1 of Geometry?



Set “Match By” to Channel Number to get Geometry and Noise Channels to add correctly



Adding Lag to Each Sphere



Append a Lag CHOP to the Match CHOP

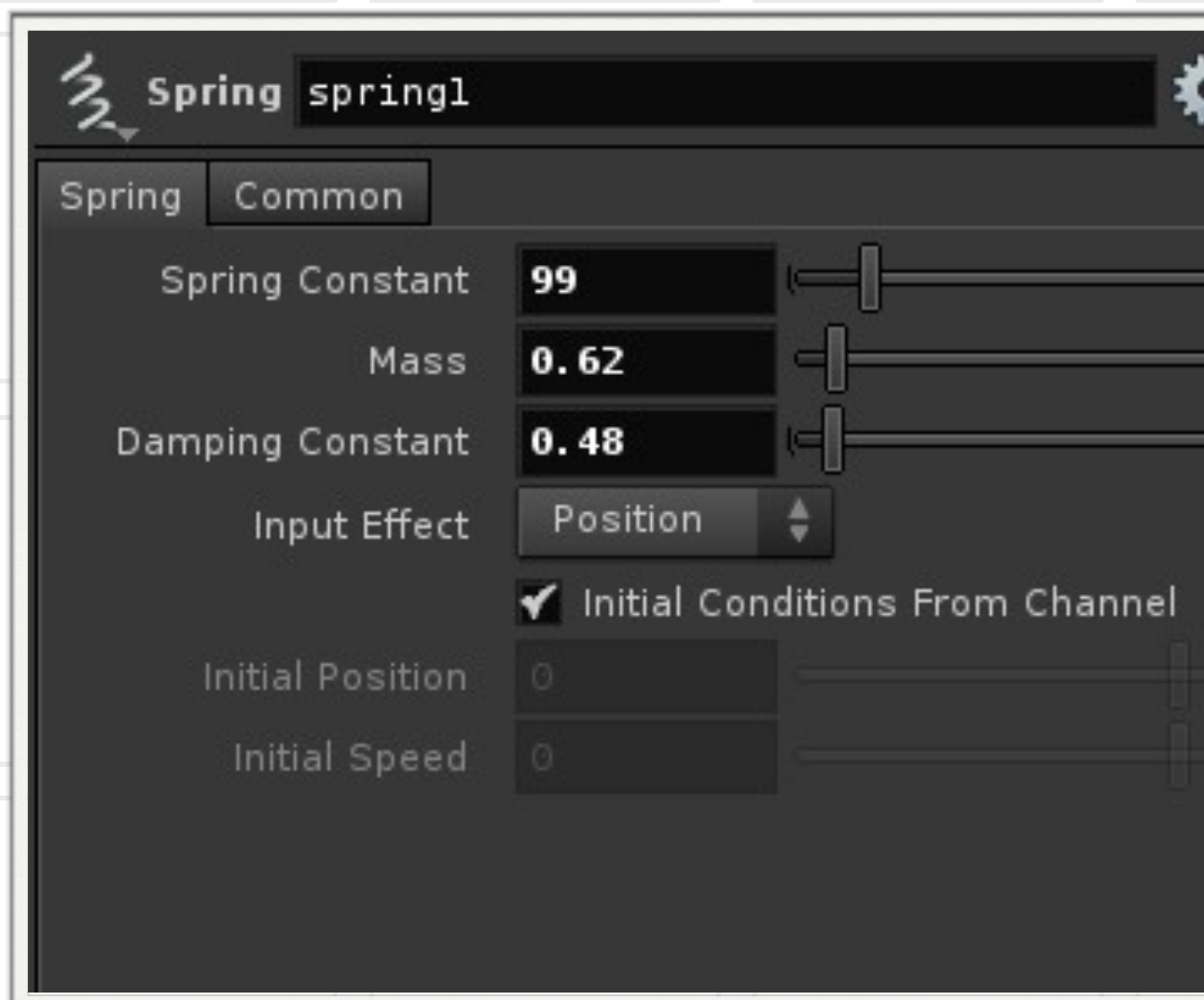
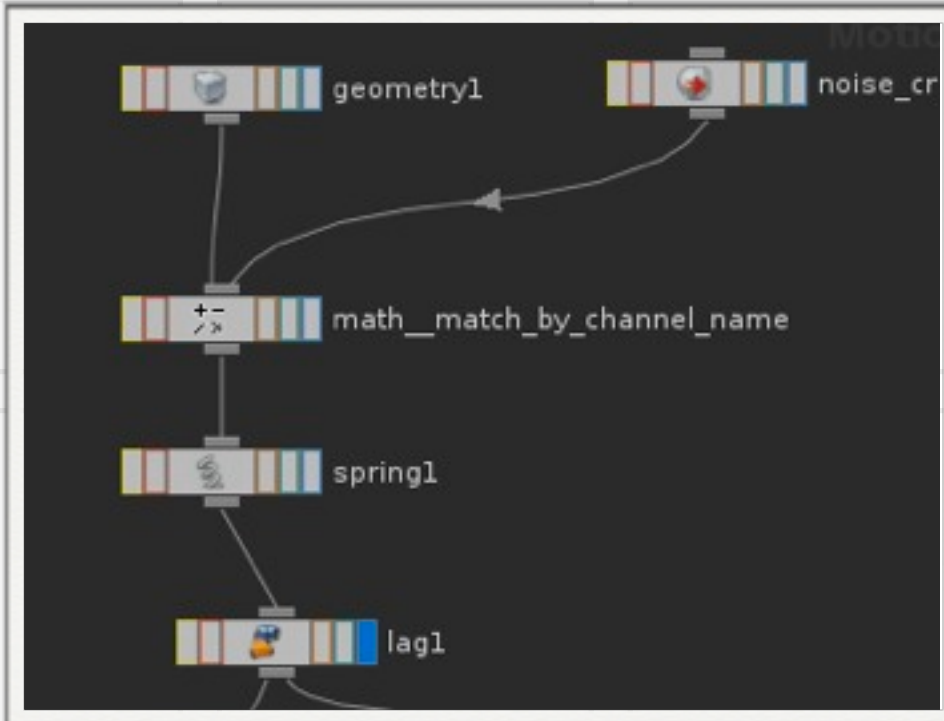
- ▶ This will give us the same lag for each sphere
- ▶ Let us write an expression based on Channel number to make the lag unique for each Sphere

Lag1 - 0.3+ fit01(rand(\$C*321.5), -0.05, 0.05)

Lag2 - 0.1 + 0.1 + fit01(rand(\$C*512.3), -0.05, 0.05)

Overshoot - 0.3

Adding Spring to Each Ball



We would like to add some springiness to each sphere so the balls snap back a bit and seem even more organic

Spring should always go before lag since you can think of Spring as an impulse function and lag as a steady state function

Adding Lines From Central Sphere to Tentacle Spheres

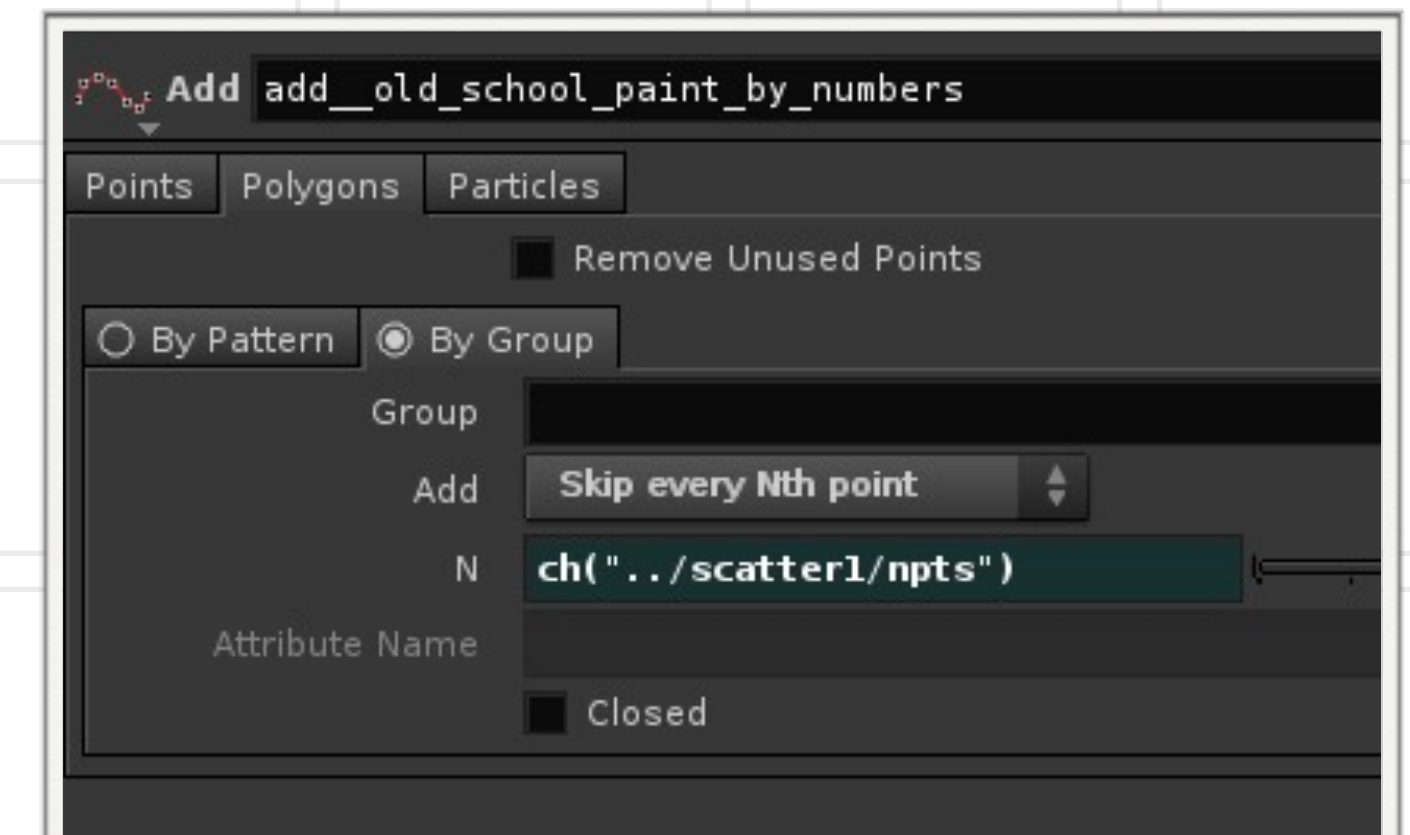
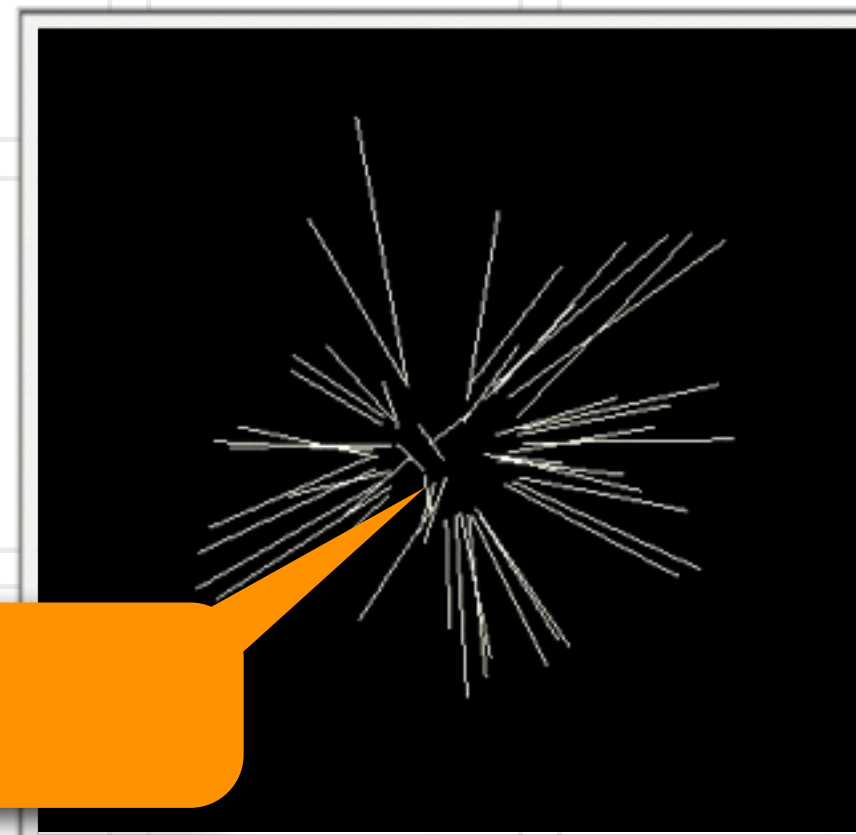
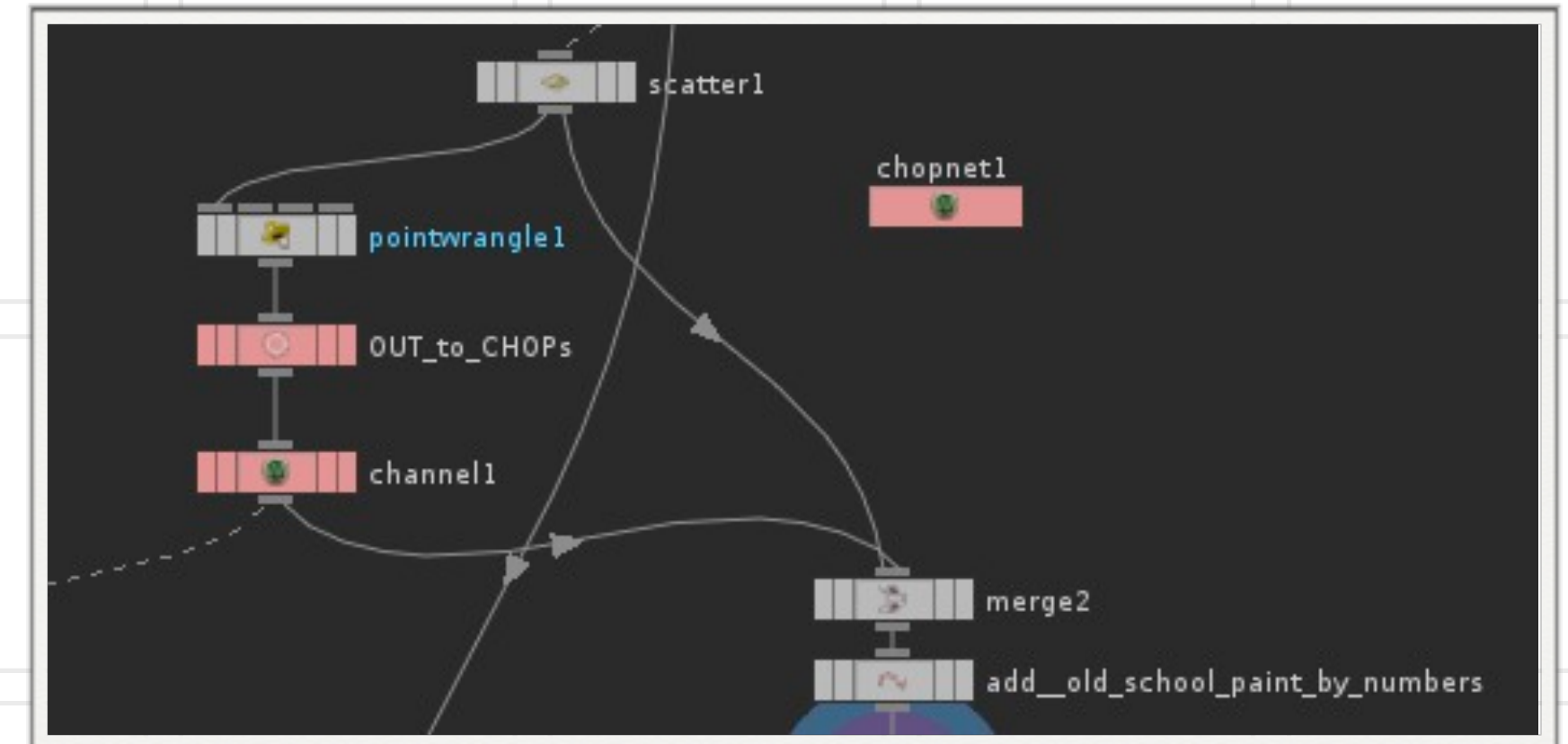
Drop Down a Merge SOP

- ▶ Wire both the Channel SOP and the Scatter SOP to the Merge

Append an “Add” SOP to the Merge

- ▶ We want connect a point on the main sphere to the corresponding point on the tentacle sphere
- ▶ Set the Add tab to “By Group”
- ▶ Add - “Skip every Nth point”
- ▶ N - `ch("../scatter1/npts")`

End result of lines created



Making The Lines Wiggle

Right now the lines move and have spring and lag but it would be nice to have the lines wiggle. Maybe like a little electrical sizzle

Append a Resample SOP to the Add SOP so we have enough points to wiggle the line

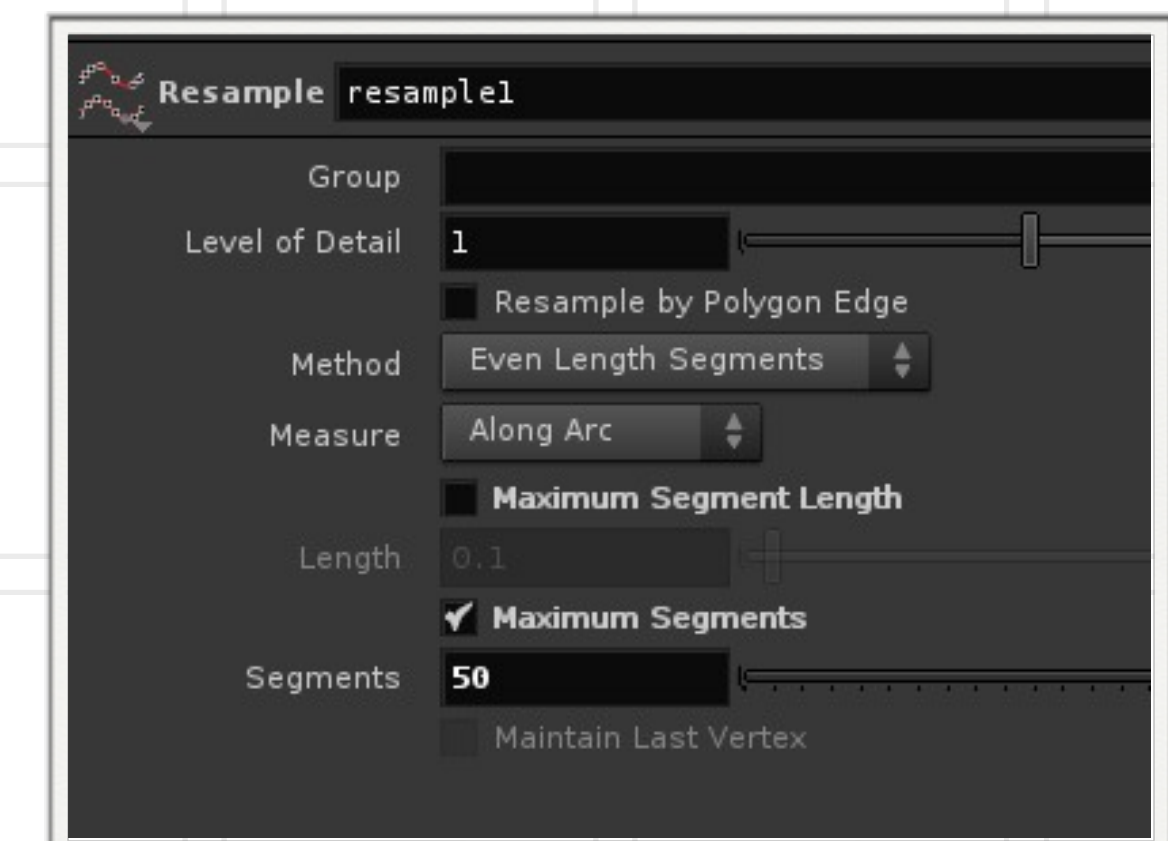
- ▶ Segments - 50

Append a NULL to the Resample SOP. We are going to create another CHOP network for the Wiggle

- ▶ Name it - IN_LINES_TO_CHOPS

Append a Channel SOP to the Null

- ▶ Method - Static



**SIDE EFFECTS
SOFTWARE**

Making The Lines Wiggle (cont.)

We can use the same CHOPNET we created before. We will just add a separate network within it

Dive inside Chopnet1

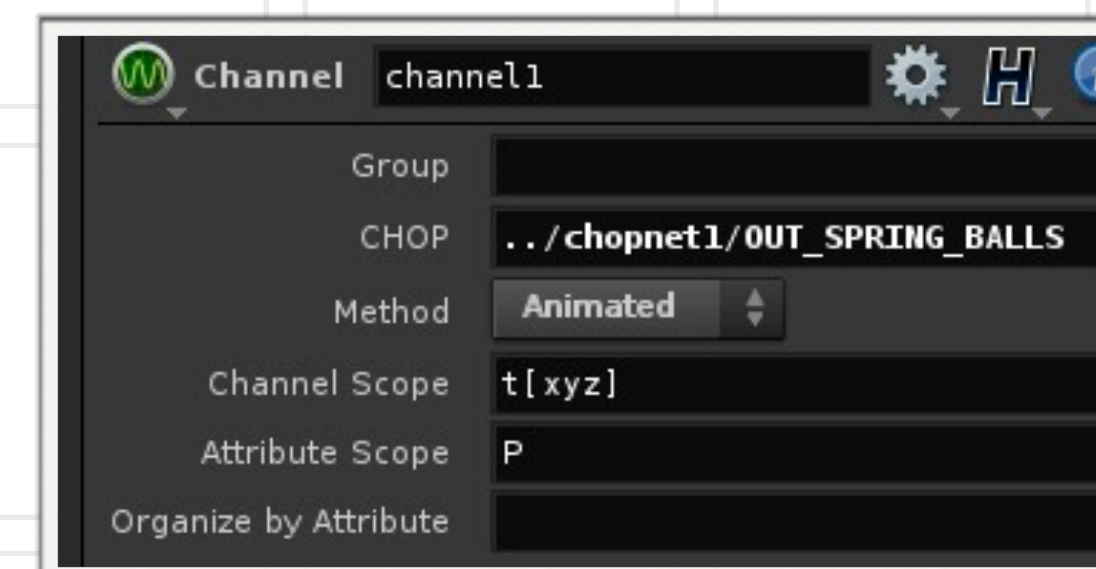
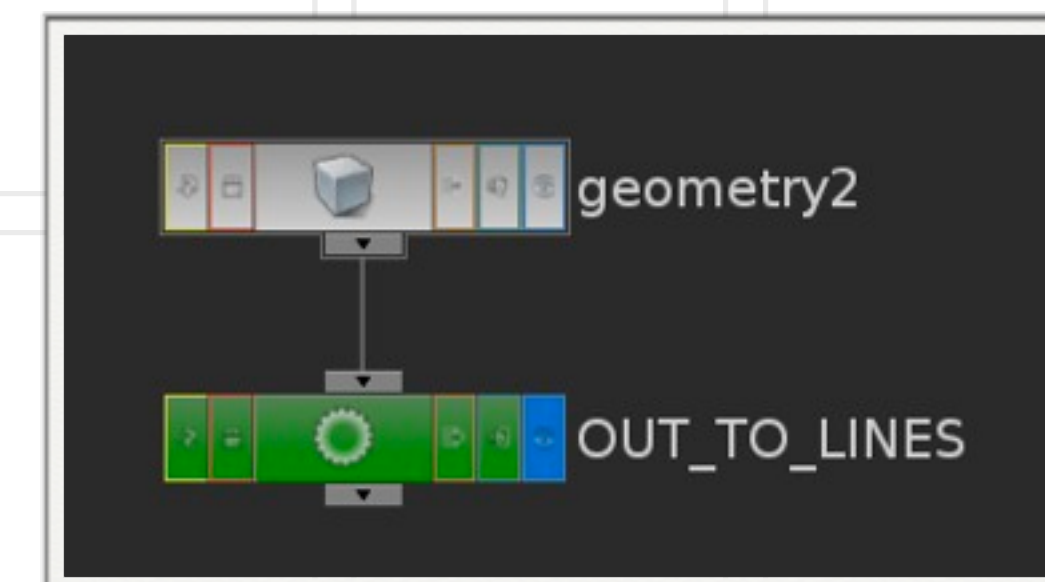
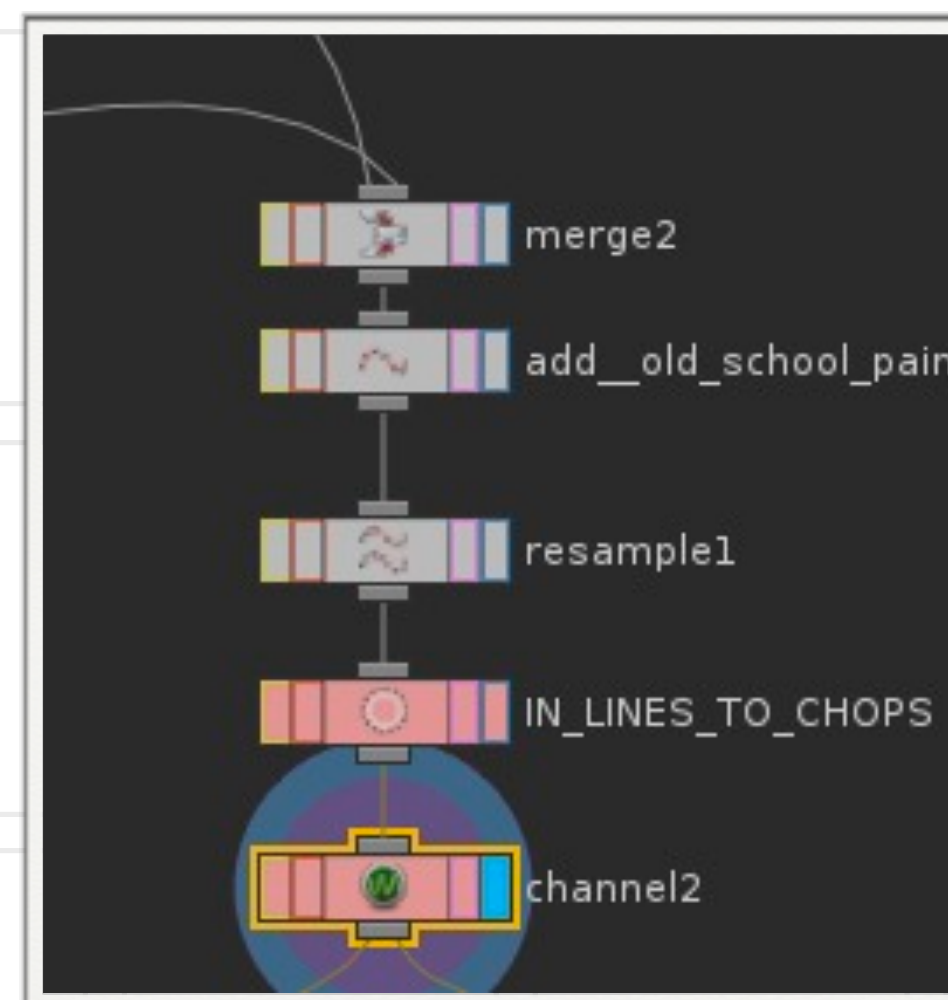
- ▶ Add a Geometry CHOP that is **NOT** attached to any node
- ▶ SOP - ../../IN_LINES_TO_CHOPS
- ▶ Method - Static

Append a NULL to the Geometry CHOP

- ▶ Name it - OUT_TO_LINES

Complete the Cycle

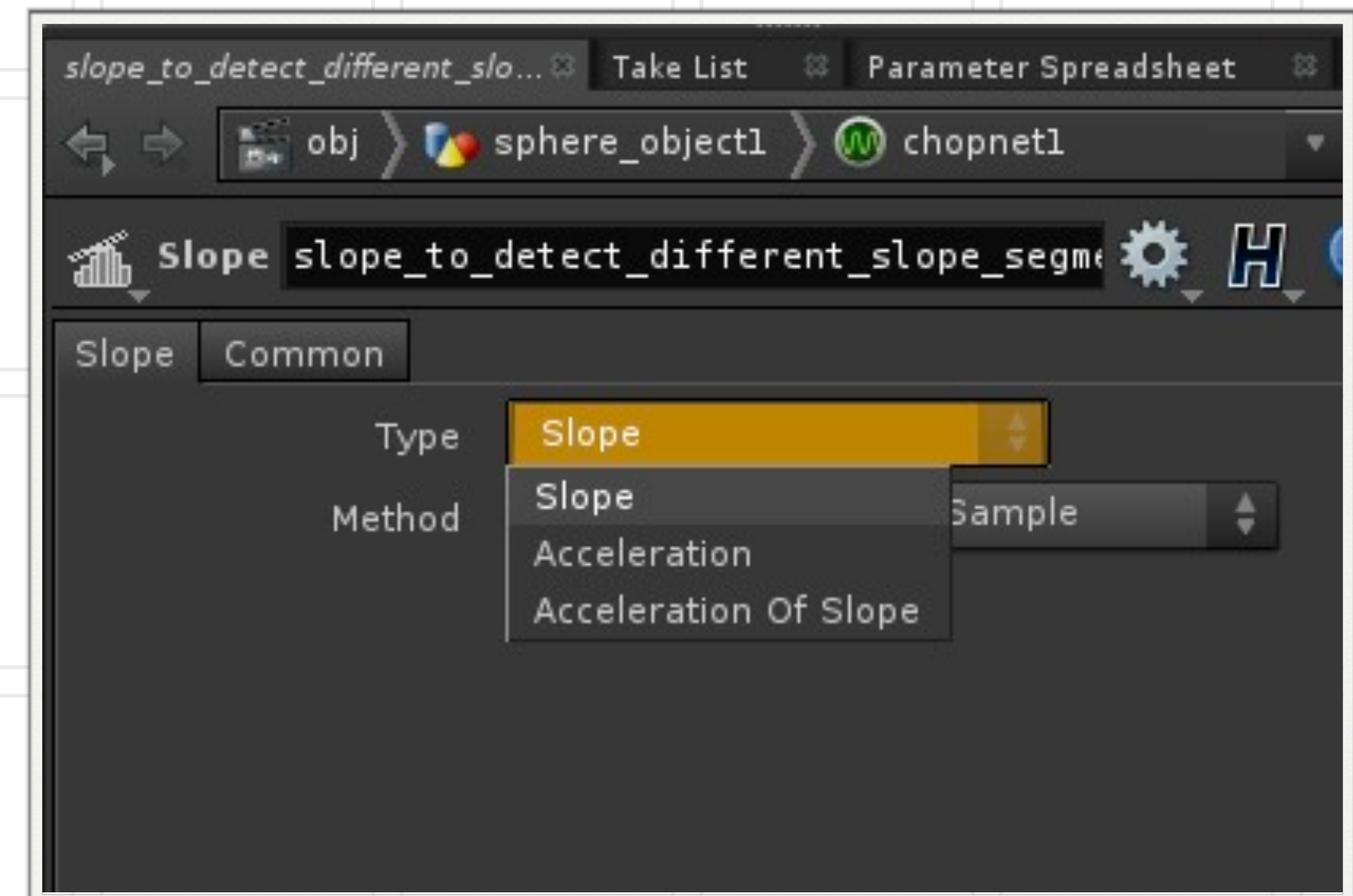
- ▶ Go up to the Geometry Level and in the Channel SOP
- ▶ CHOP - ../chopnet1/OUT_SPRING_BALLS



More on the Slope CHOP

The Slope CHOP has a drop down menu to allow you to calculate:

- ▶ First Derivative - Slope (velocity)
- ▶ Second Derivative - Acceleration
- ▶ Third Derivative - Acceleration of Slope



Making The Lines Wiggle (cont.)

Now Append a Trigger CHOP to the Slope CHOP

- ▶ Trigger CHOP adds an audio-style attack/decay/sustain/release (ADSR) envelope to all trigger points in the input channels. A trigger point occurs whenever the first input's channel increases across the trigger threshold value.

Trigger Tab

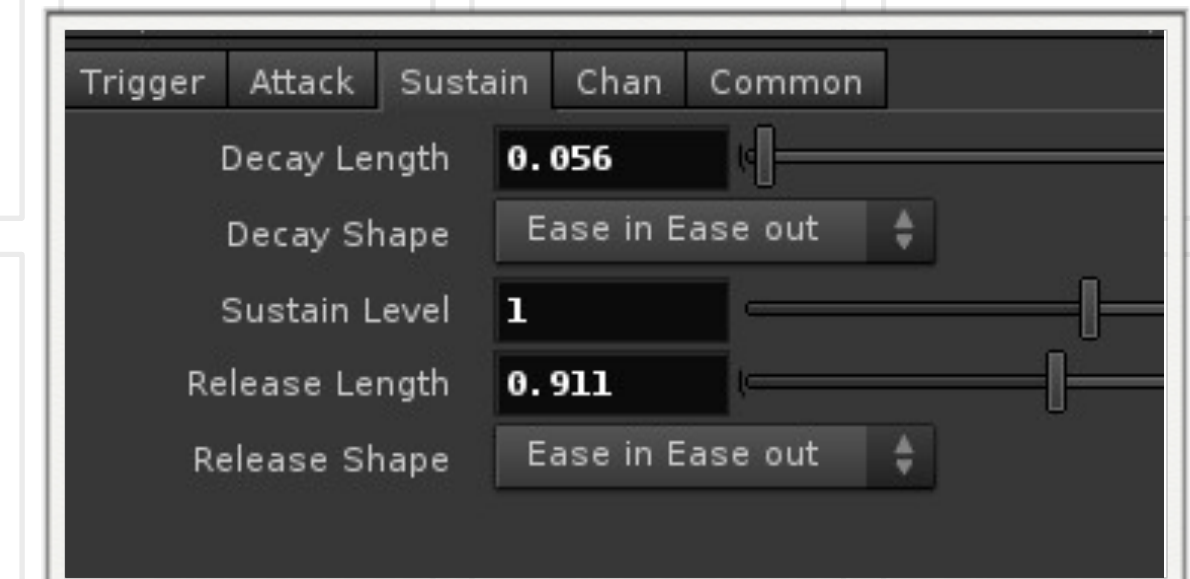
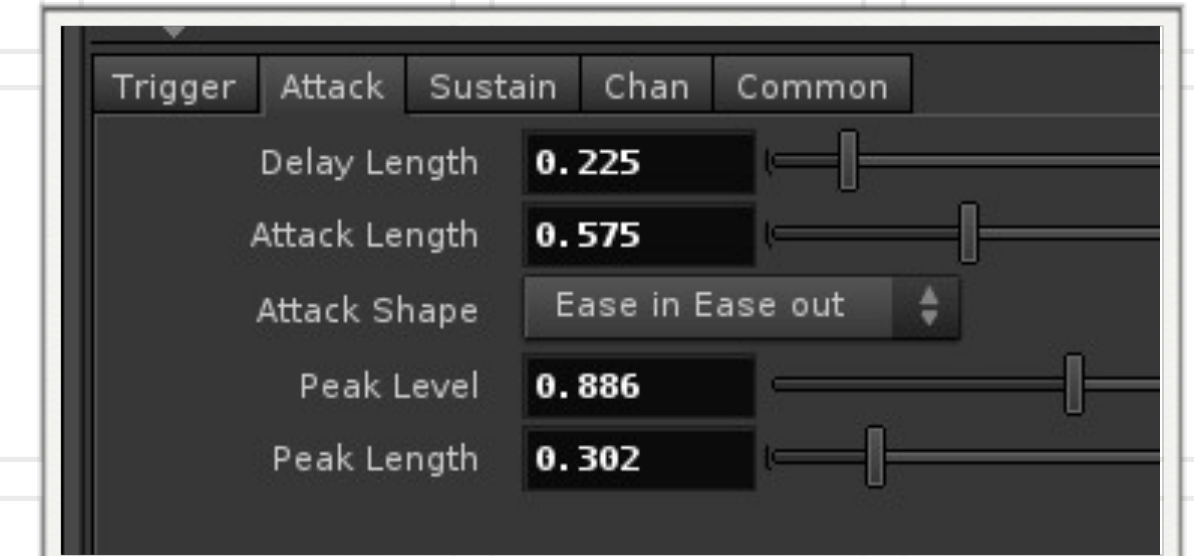
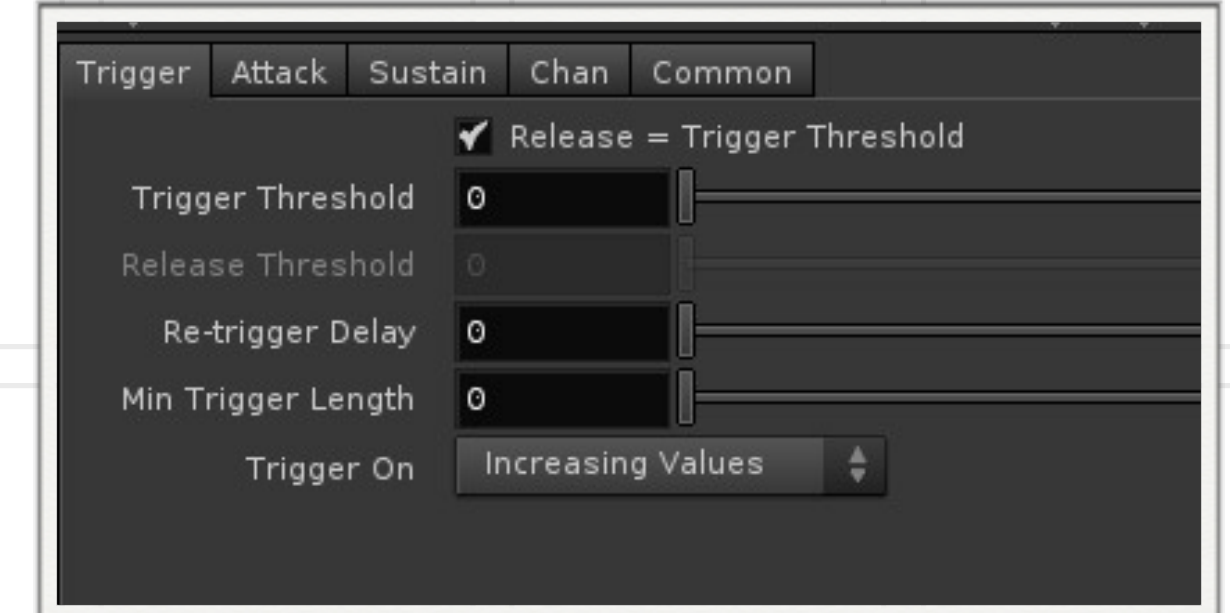
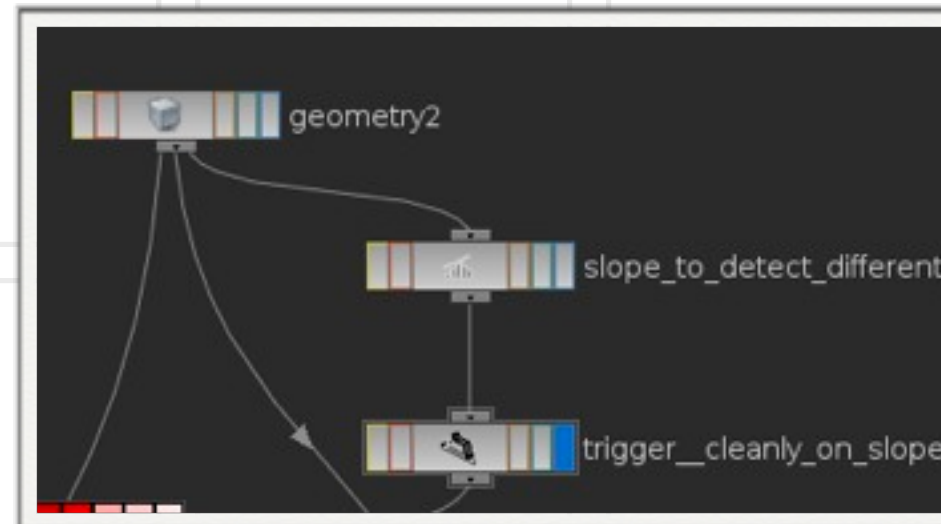
- ▶ Trigger On - Increasing Values

Attack Tab

- ▶ Delay Length - 0.225, Attack Length - 0.575
- ▶ Attack Shape - Ease In Ease Out
- ▶ Peak Level - 0.88, Peak Length 0.3

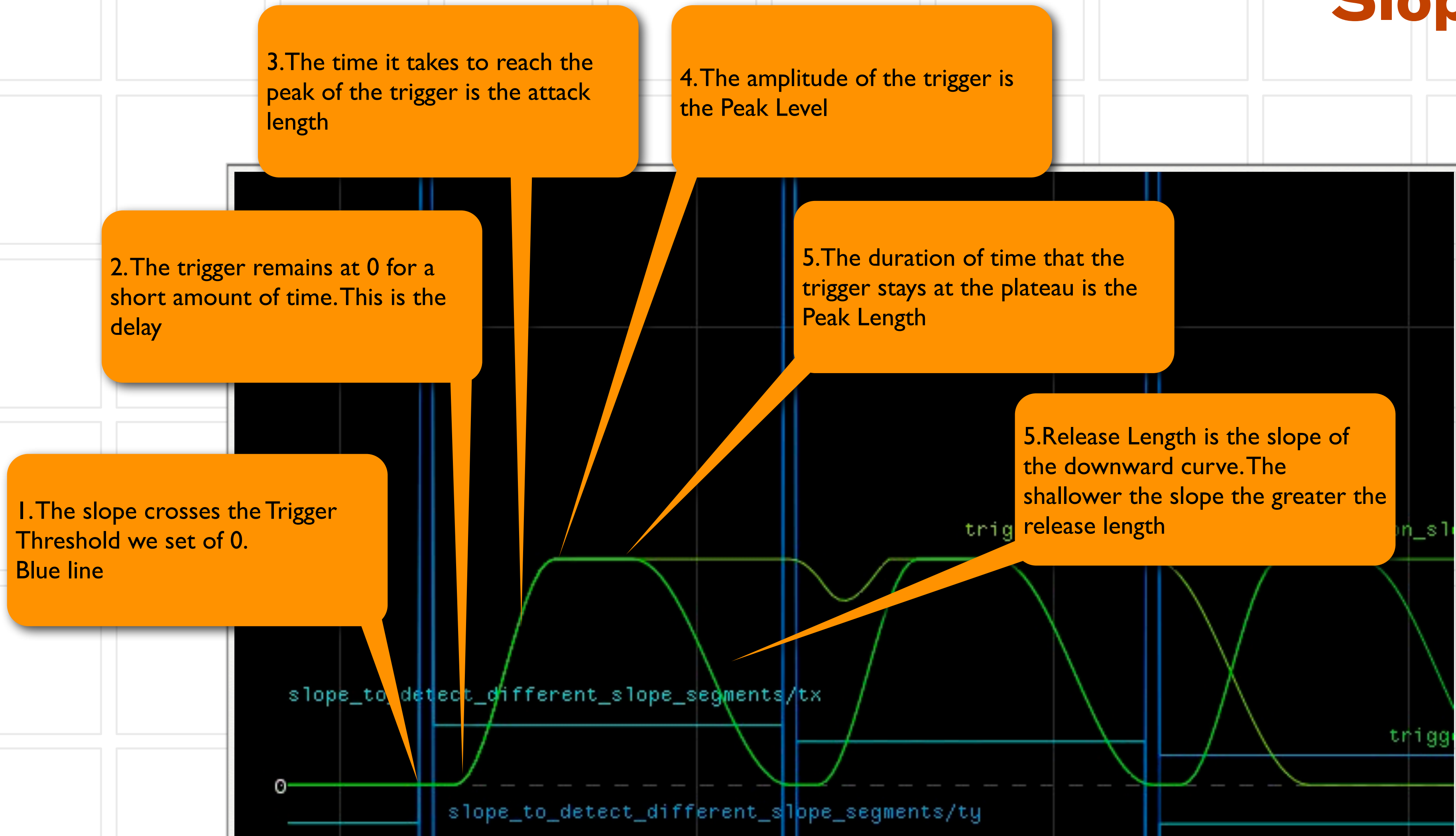
Sustain Tab

- ▶ Decay Length - 0.056, Decay Shape - Ease In Ease Out, Release Length 0.911



SIDE EFFECTS
SOFTWARE

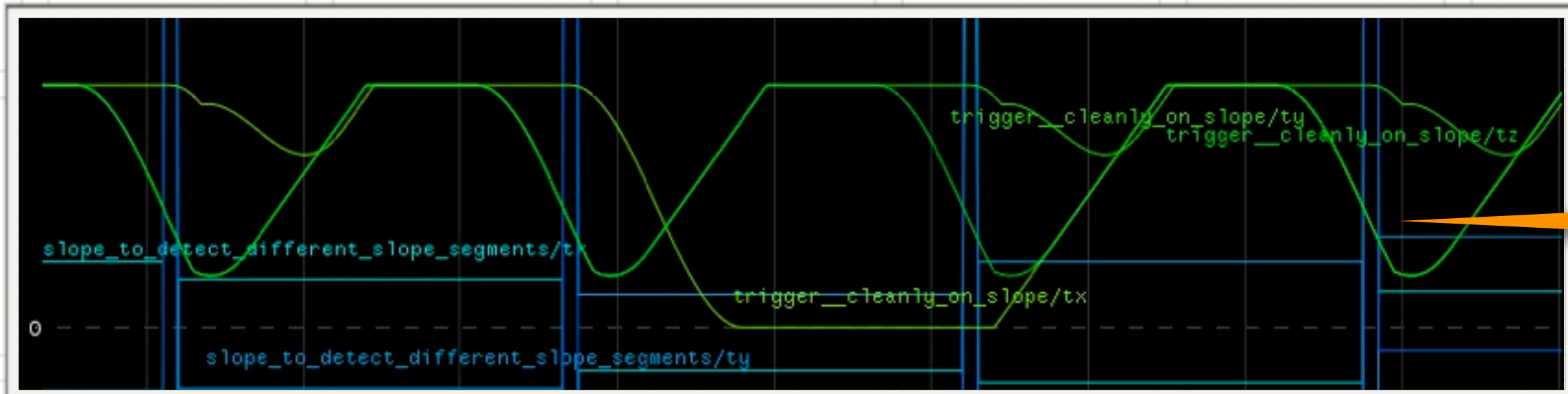
Slope and Trigger



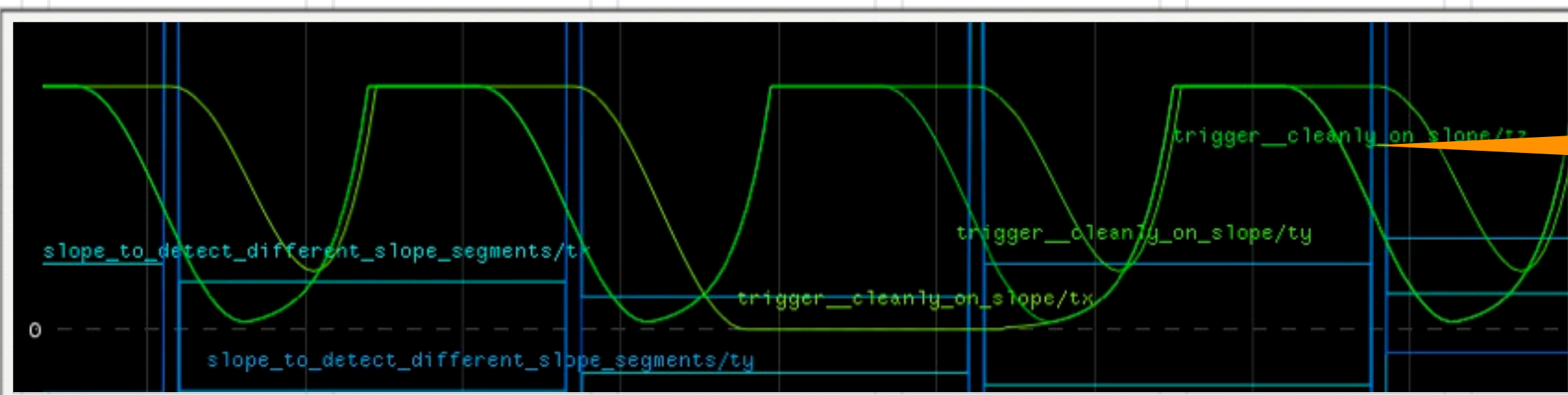
Trigger Shapes



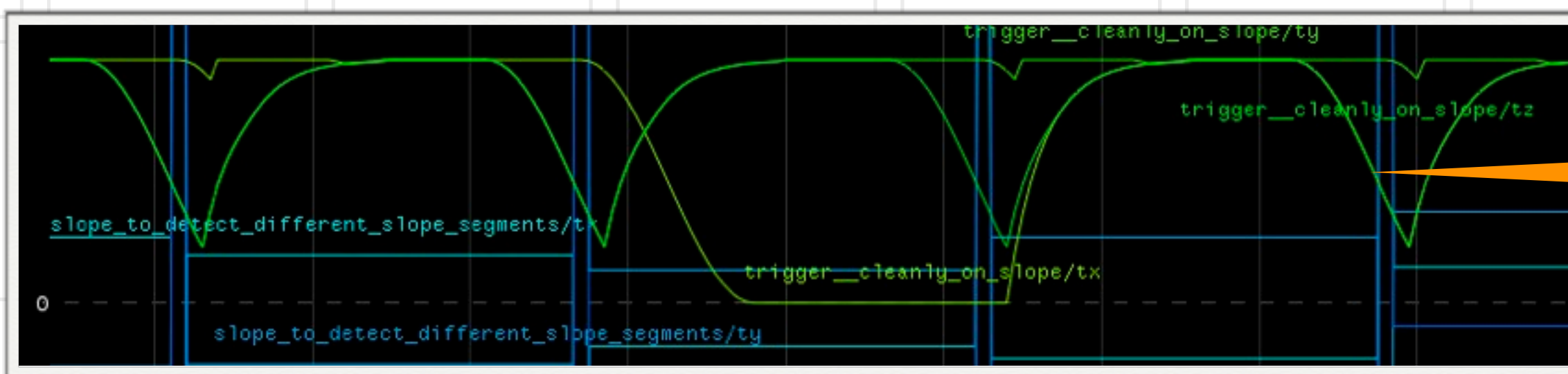
Ease In Ease Out



Linear



Ease In



Ease Out

SIDE EFFECTS
SOFTWARE

Using a Math CHOP to do a Fit Range

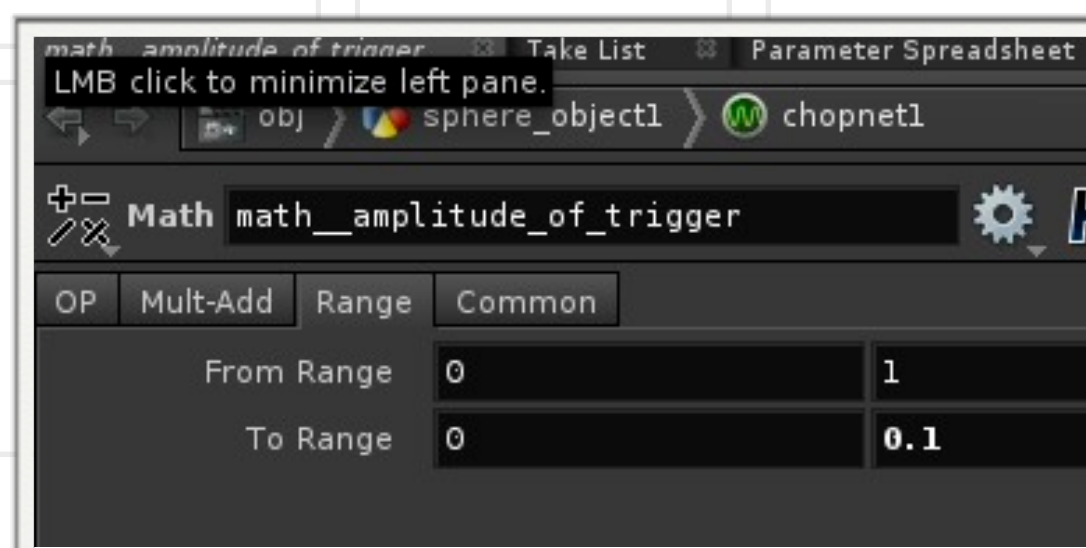
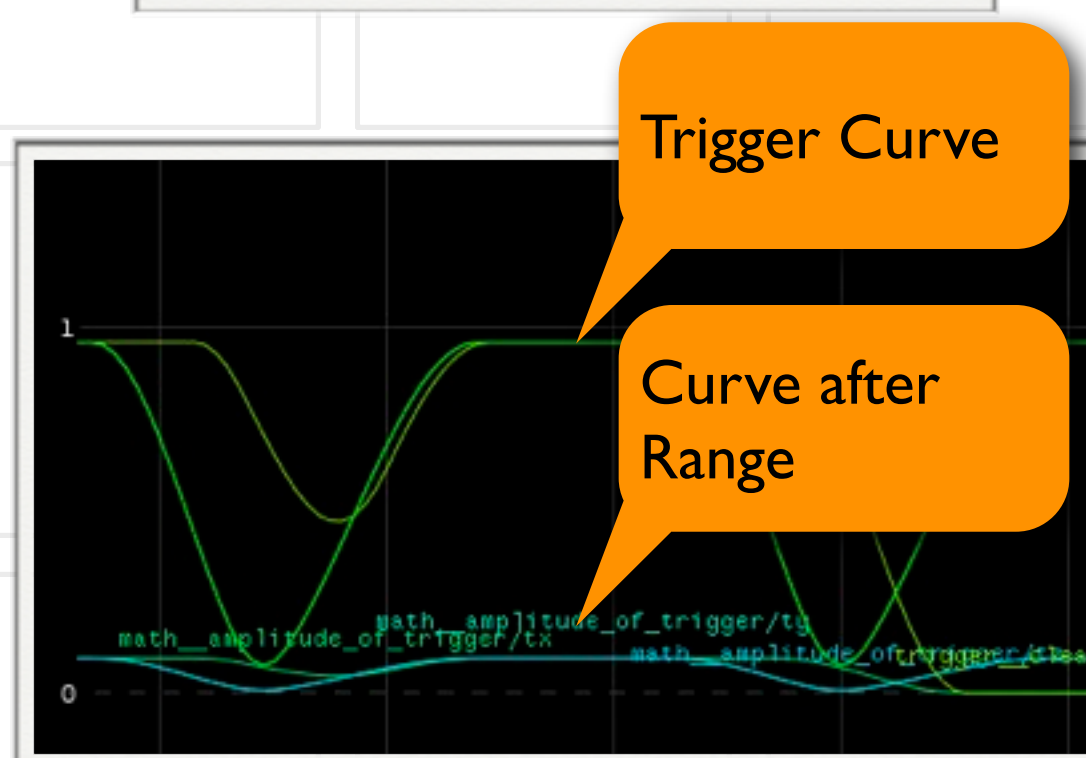
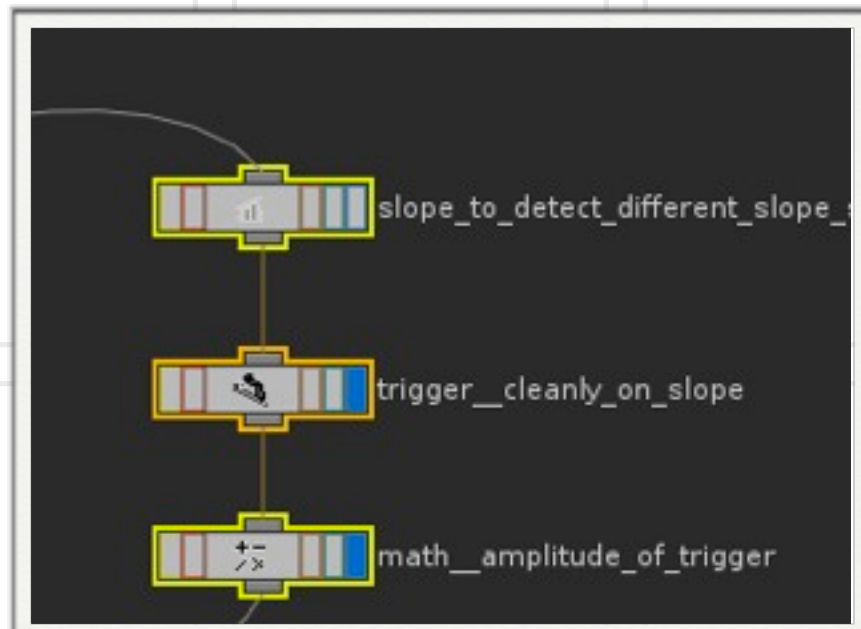
Append a Math CHOP to the Trigger

- ▶ Currently the trigger is giving values between 0 and 1
- ▶ We want the wiggle to be small vibrations compared to the main Motion

Select the Range Tab

- ▶ From Range - 0,1
- ▶ To Range - 0, 0.1

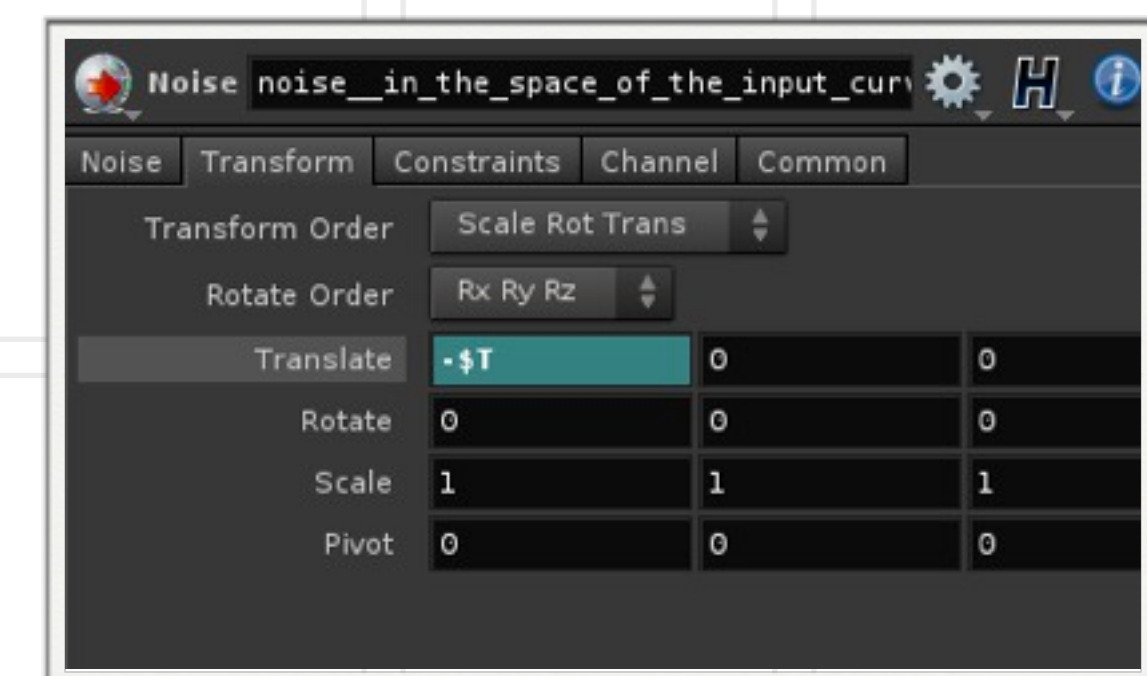
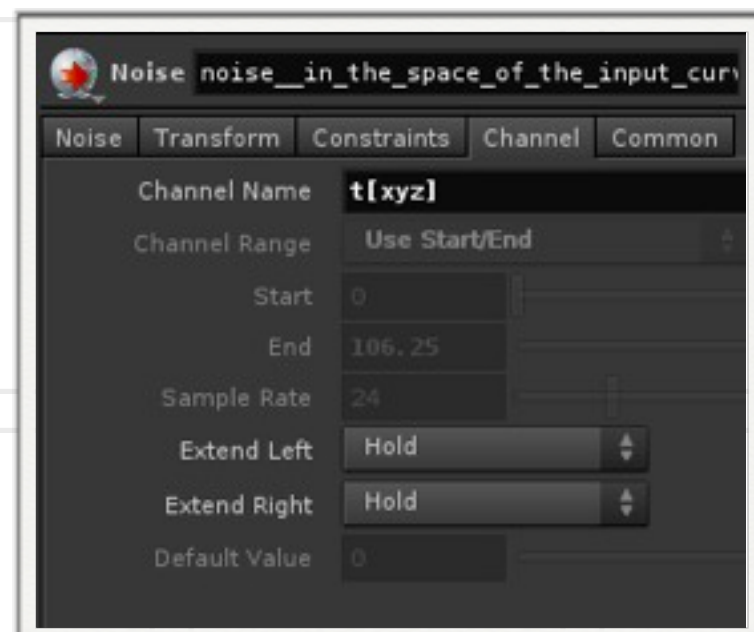
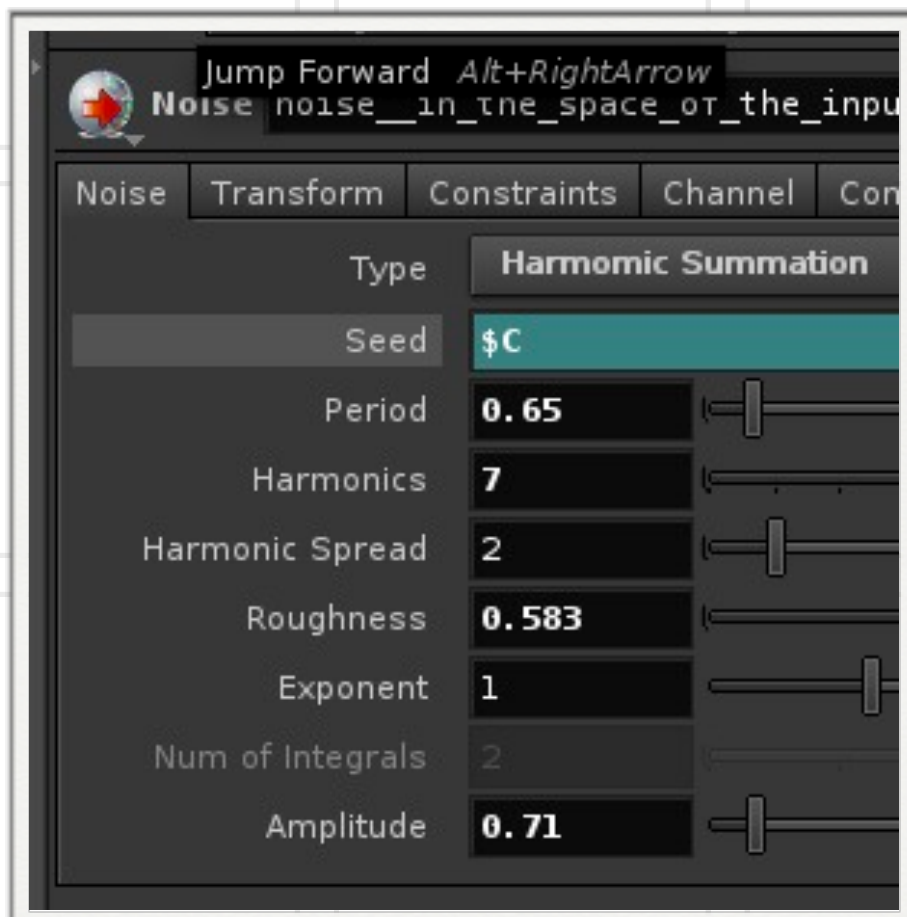
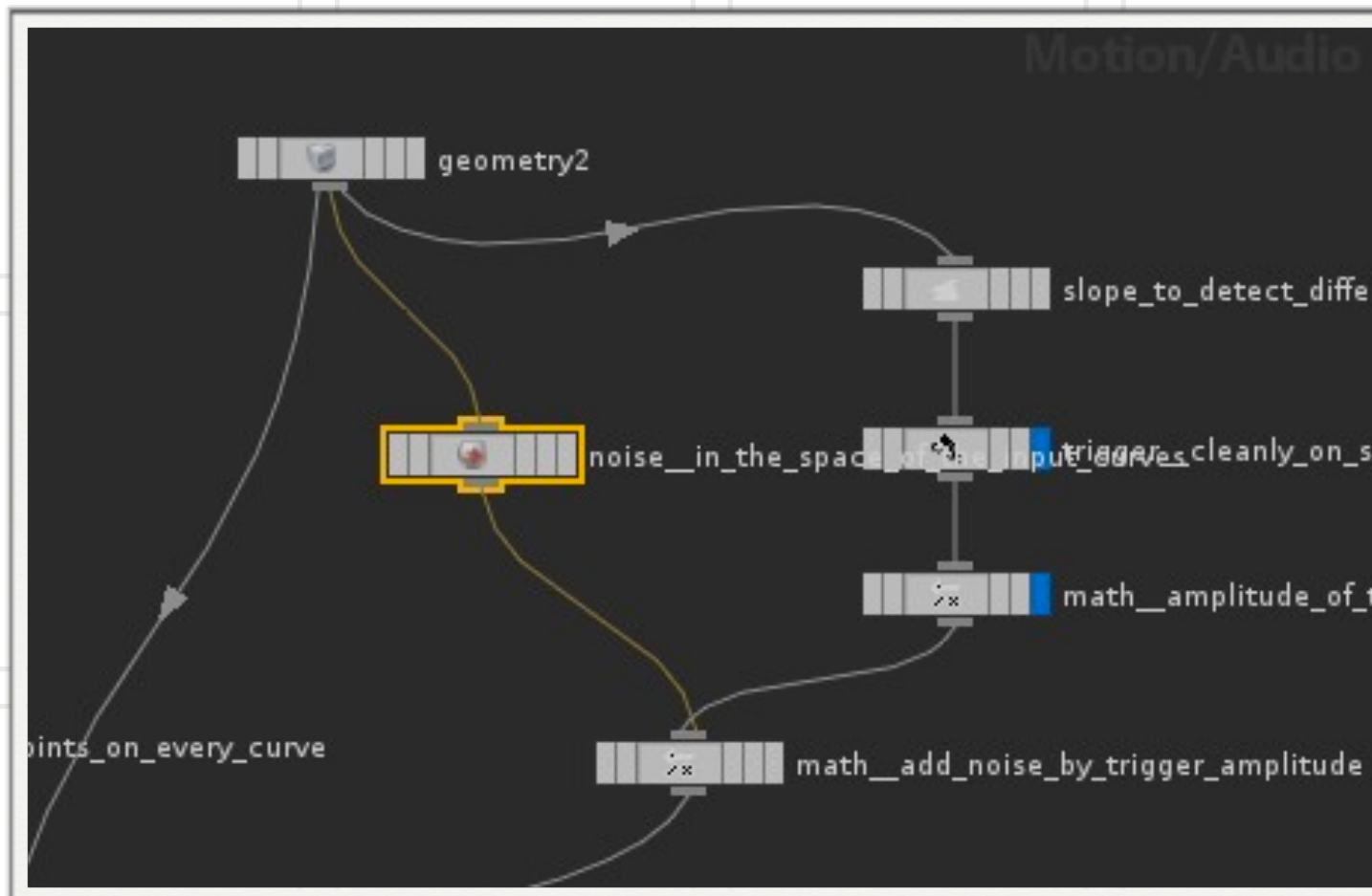
Remember the Trigger is only use to modulate the on/off of the noise we are going to build now



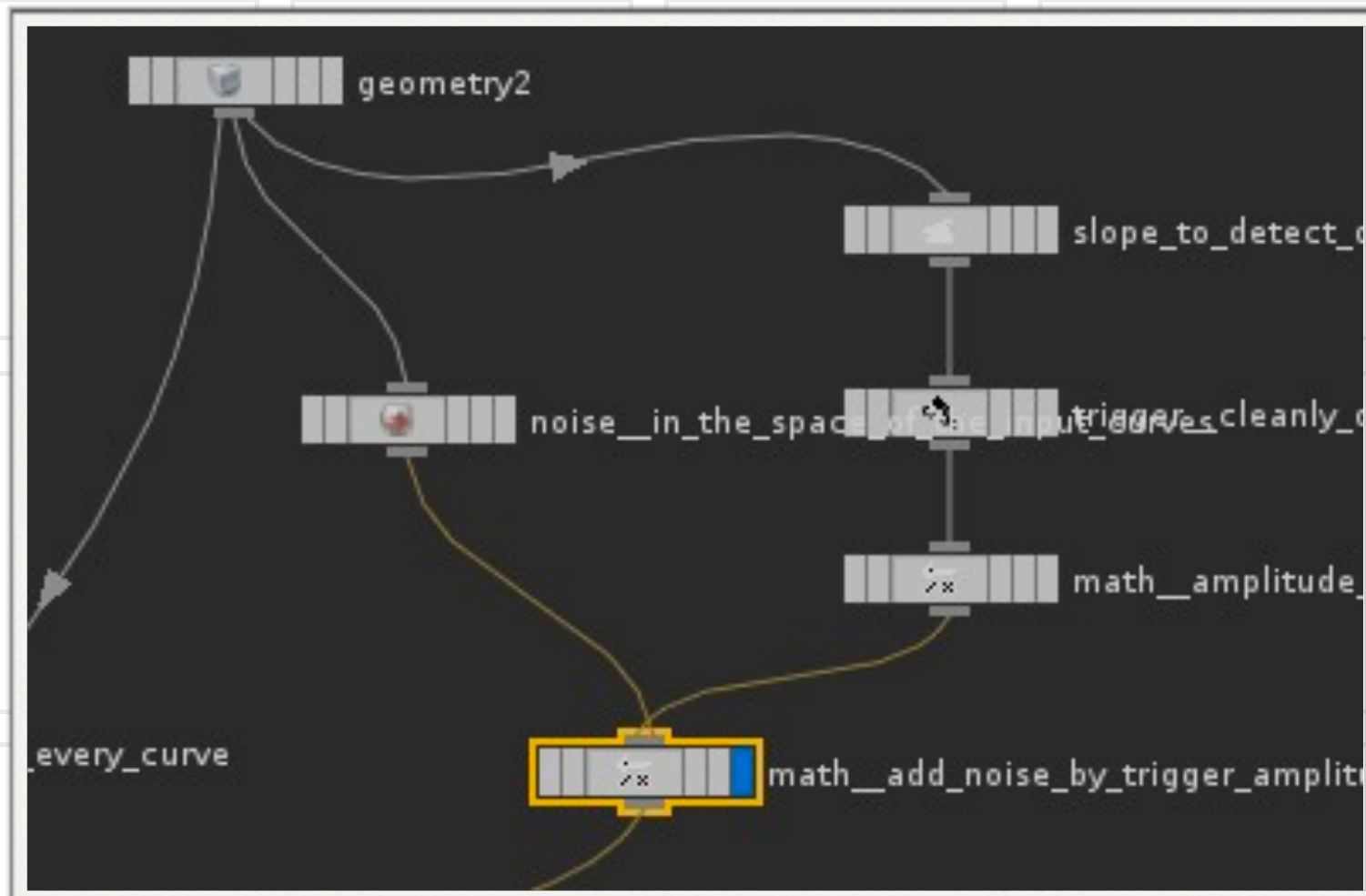
Building the Noise of the Wires

Append a Noise Chop to the Geometry

- ▶ Seed - $\$C$
- ▶ Play with other values
- ▶ In the Transform Tab
 - ▶ Translate X - $-\$T$ (have the noise move down the wire)
- ▶ In the Channel Tab
 - ▶ Channel Name - $t[xyz]$



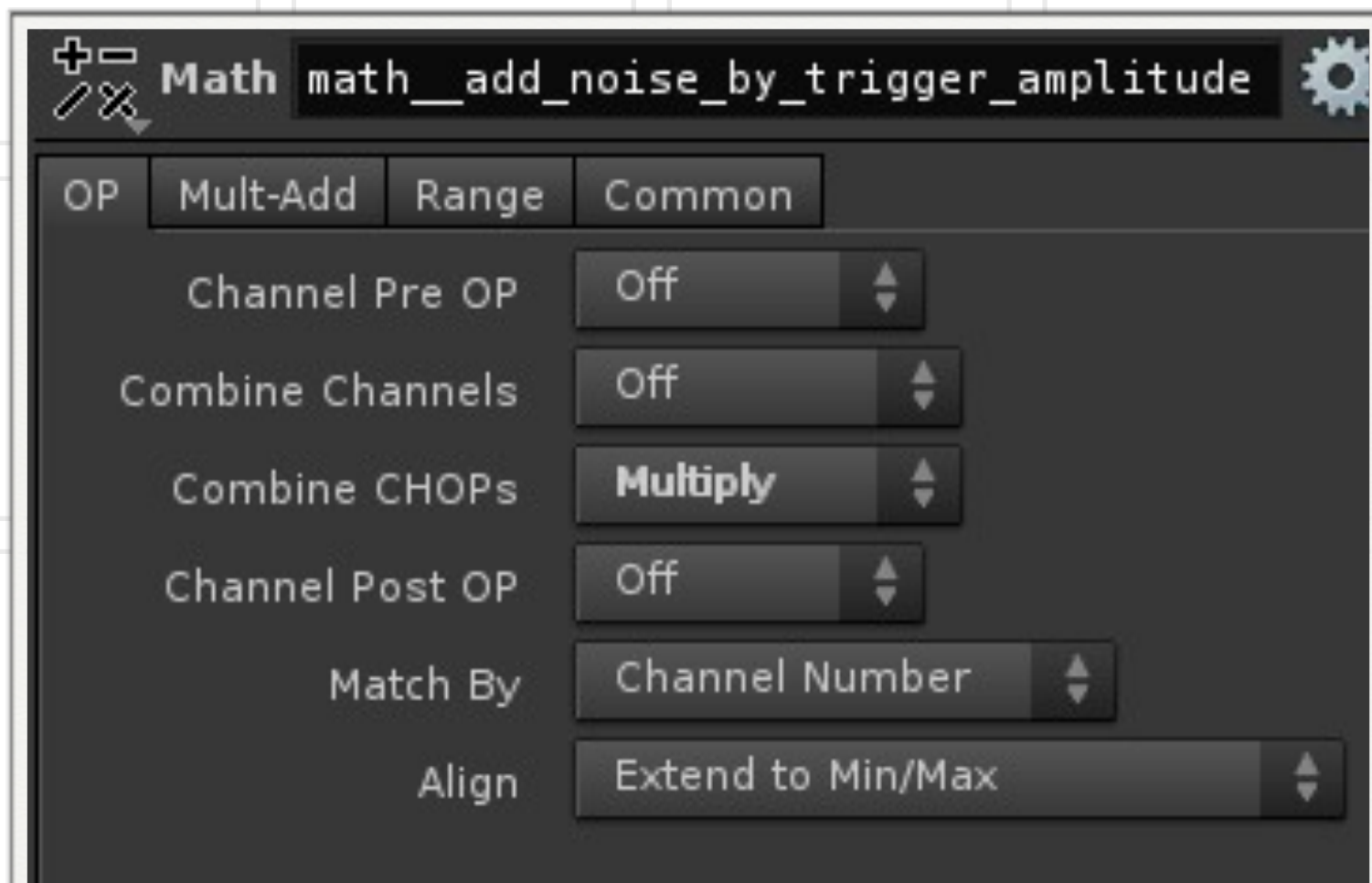
Multiplying the Noise by the Trigger to Modulate the Noise



Drop down a Math CHOP

Wire the Noise CHOP and the Math CHOP from the Trigger to the new Math CHOP

Set Combine CHOP - Multiply

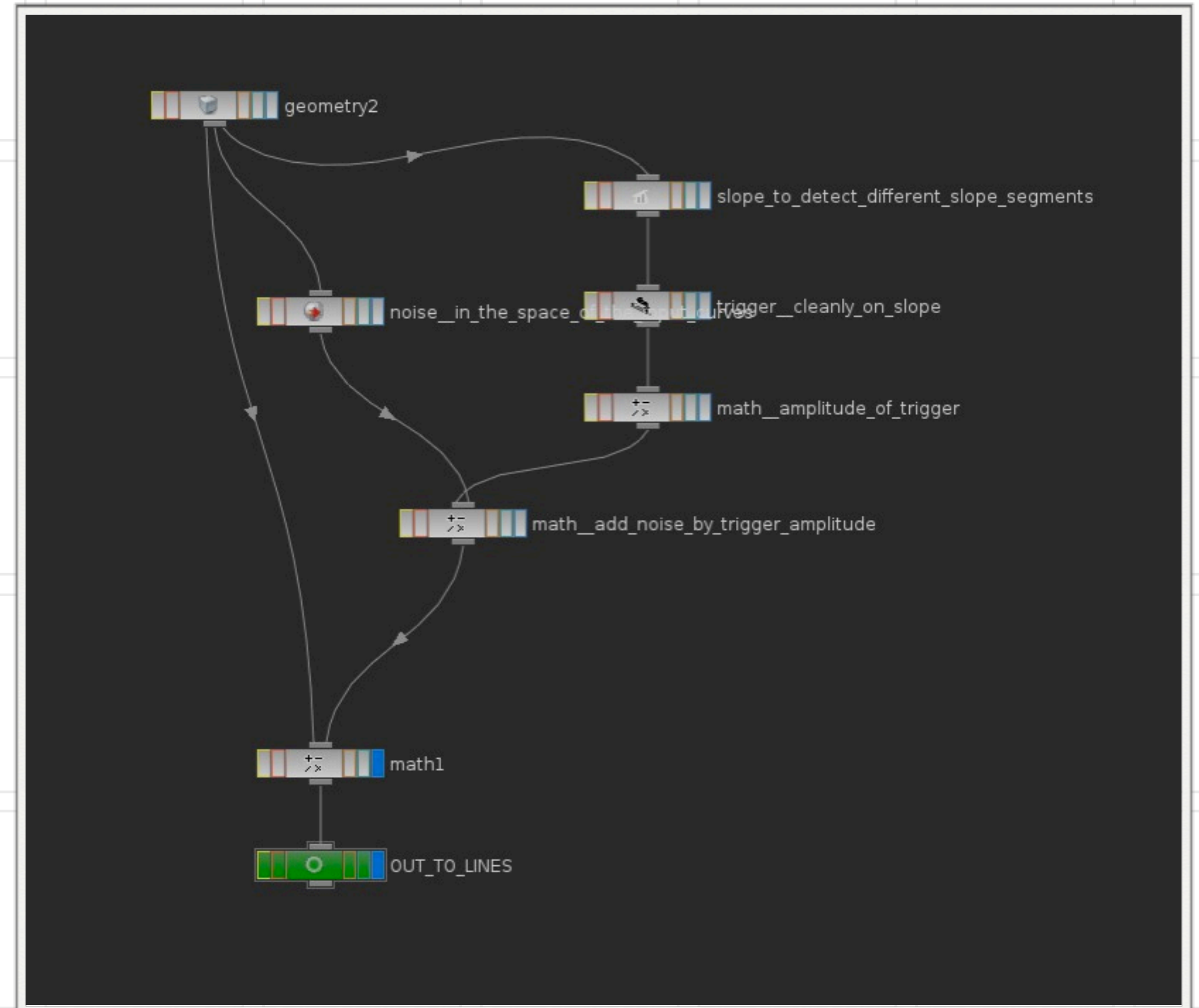
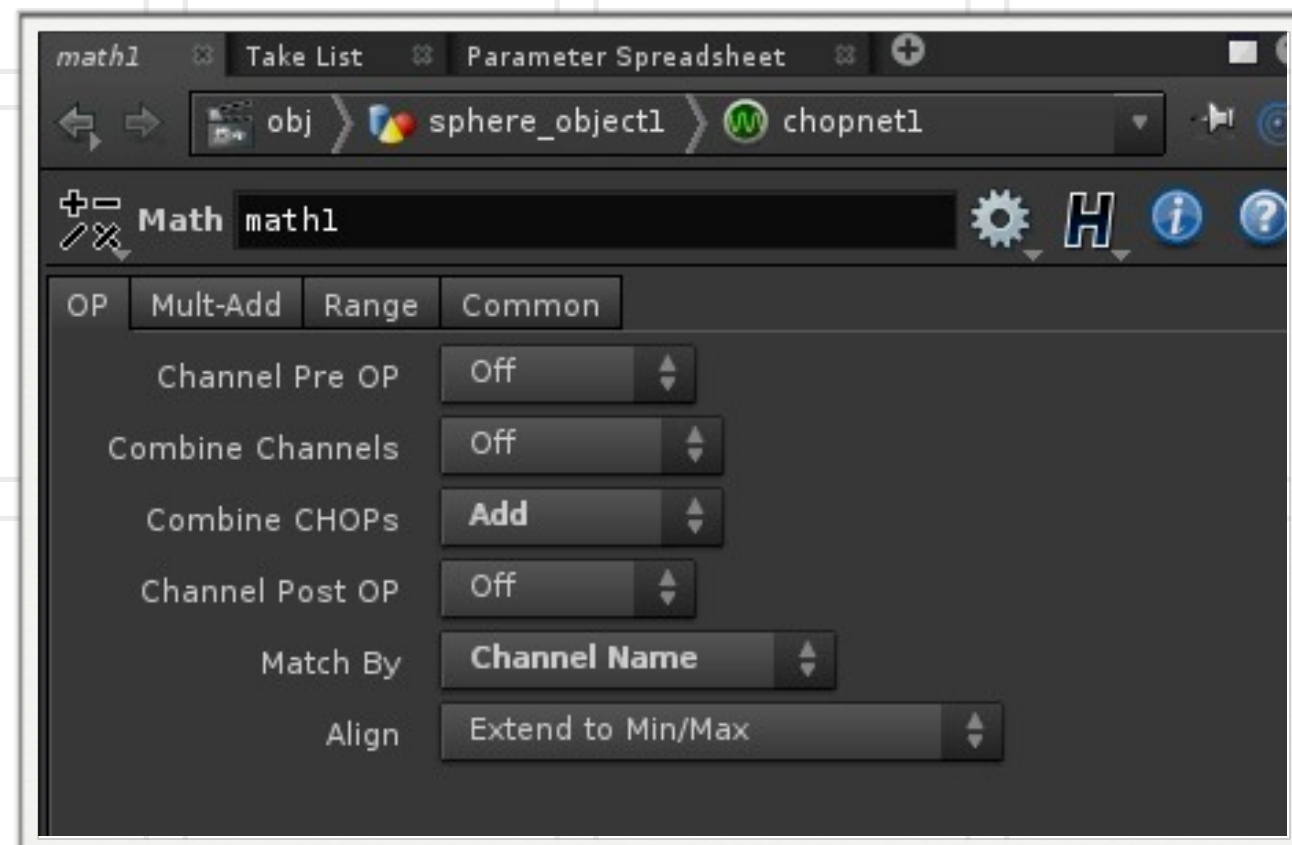


SIDE EFFECTS SOFTWARE

Adding Modulated Noise to Original Wire Position

Complete Network as shown on right

Set Combine CHOPs to Add

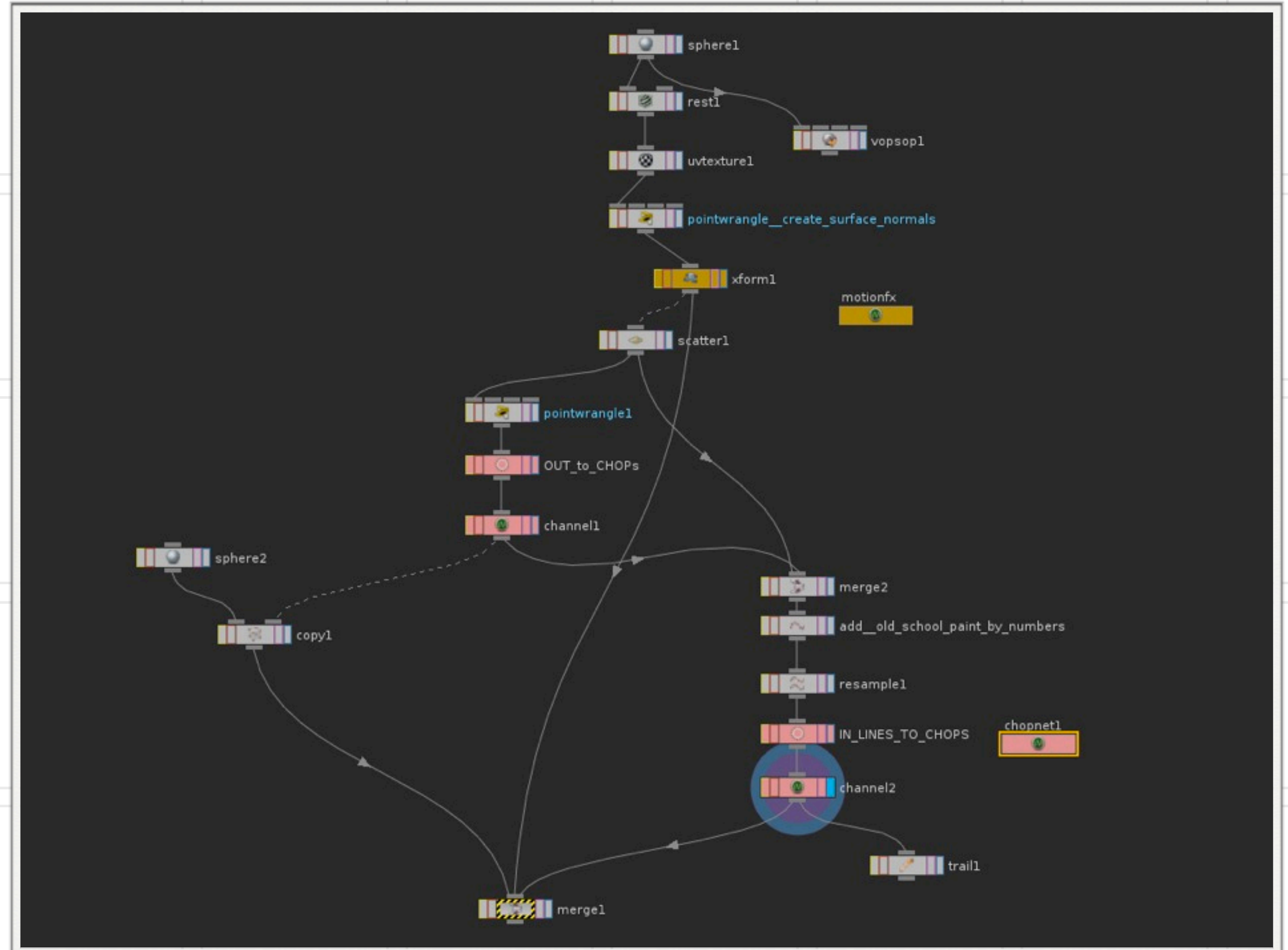
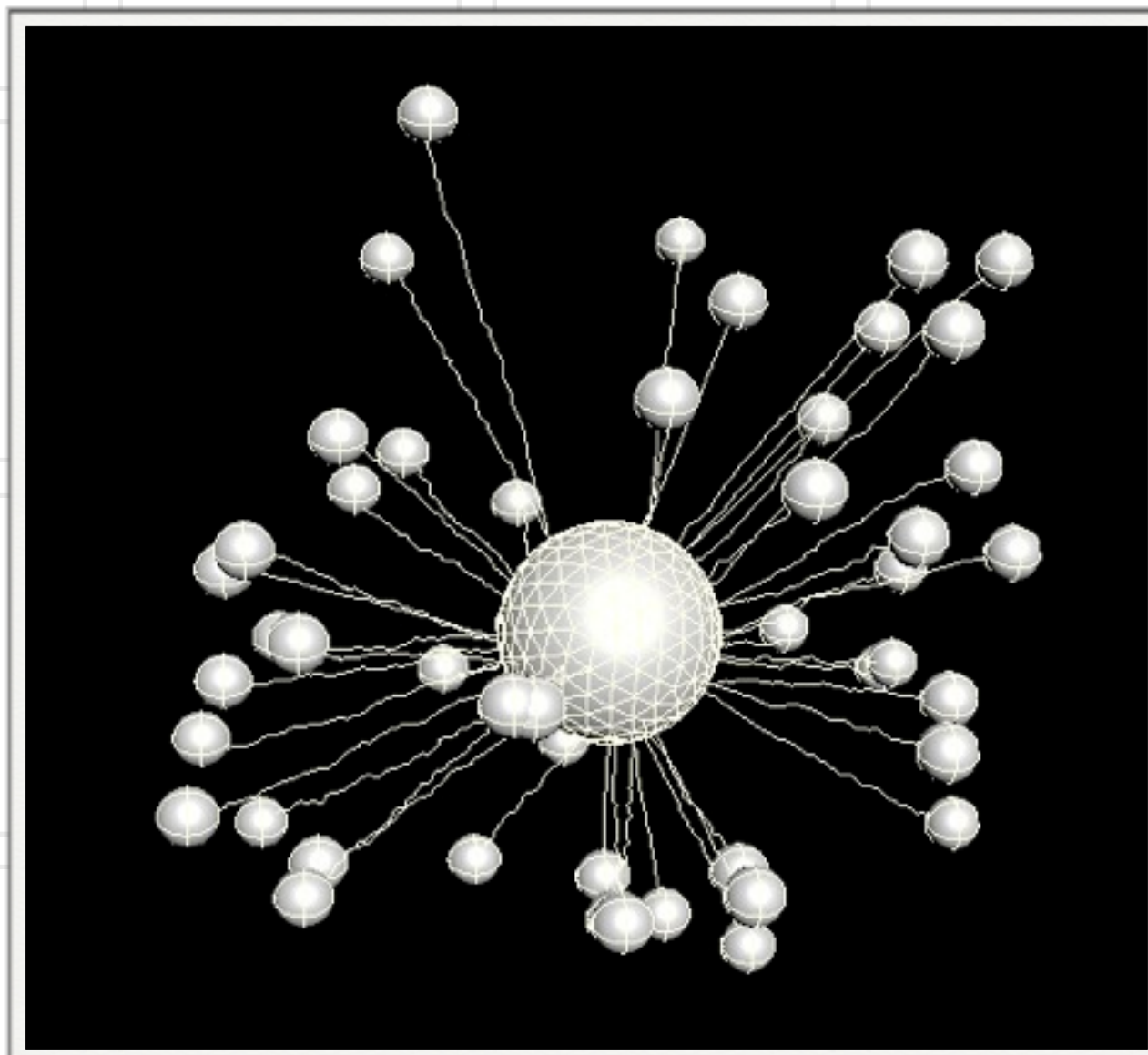


SIDE EFFECTS
SOFTWARE

Finishing the SOP Network

Jump up to the Geometry Level

Use a Merge SOP to Merge the Main Sphere Motion with the Tentacle Spheres



SIDE EFFECTS
SOFTWARE



End of Module 01

Procedural Animation

**SIDE EFFECTS
SOFTWARE**