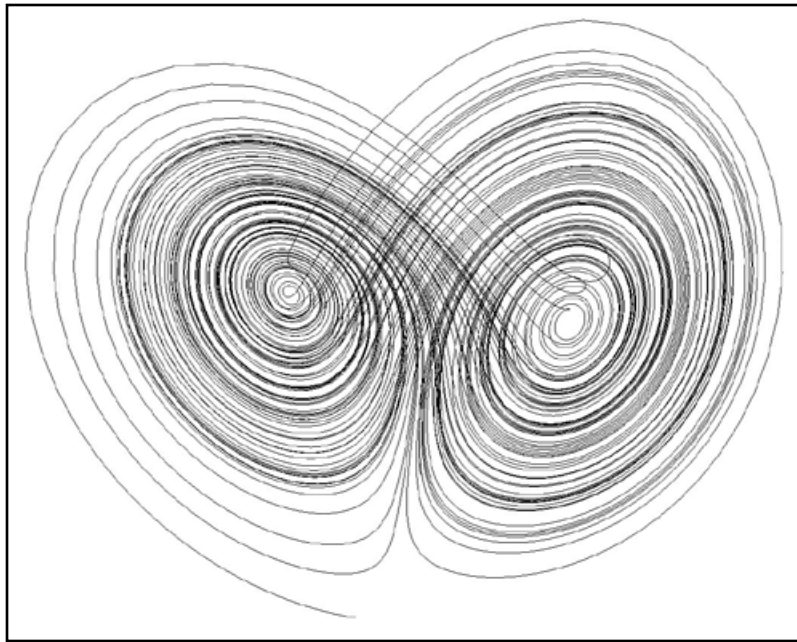


The Lorenz Attractor OTL

—

Creating a Python Operator with Houdini 9

Georg Duemlein



The Industrial Parametrisation of the World

<http://www.preset.de/>

2007

The Lorenz Attractor OTL

Creating a Python Operator with Houdini 9

Georg Duemlein

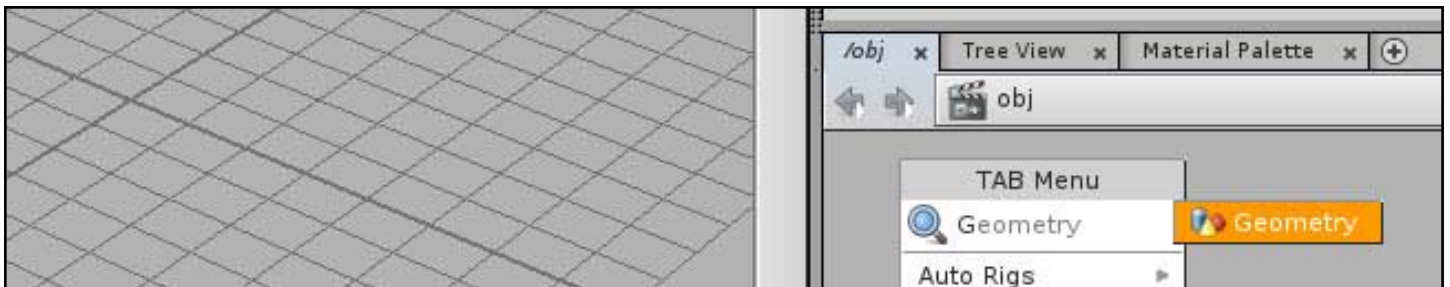
Abstract

This walk through explains the usage of the new python operator in Houdini 9. We will create a new SOP that modifies incoming geometry points to form a Lorenz Attractor [eg. Boruke 1997].

Preparation

After starting Houdini 9 save the file to a directory of your choice.

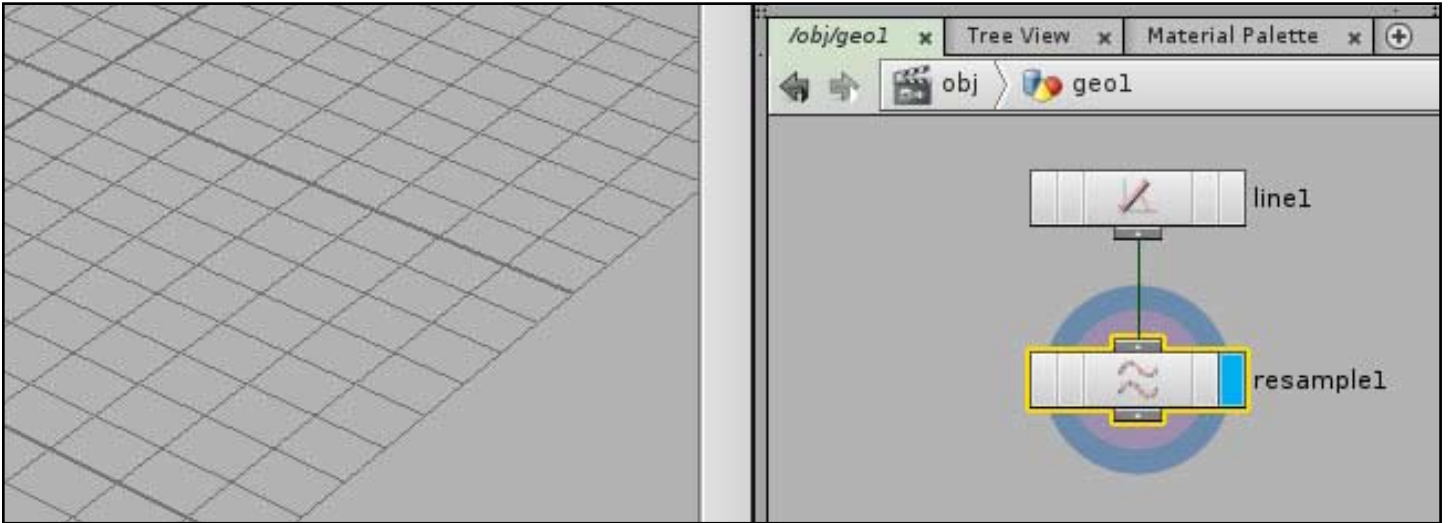
Now create a **Geometry** node with `[TAB]-ge`



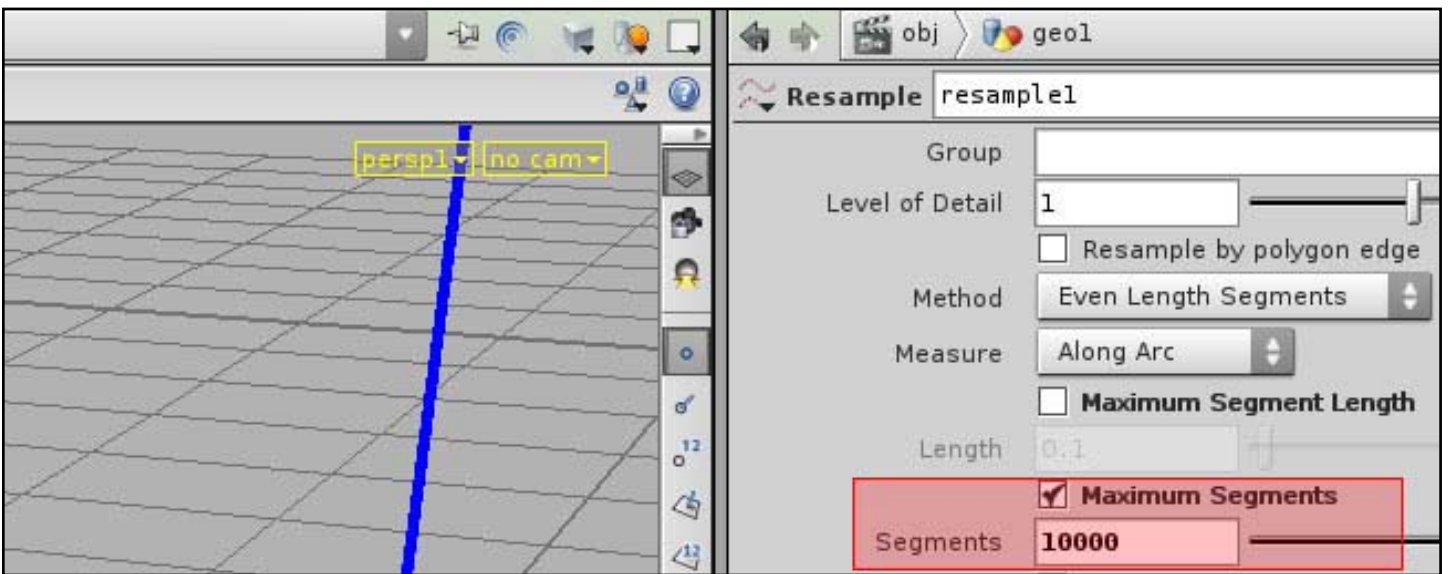
Press *ENTER* three times to select the node, lay it down and enter it.

*** **Note:** maybe the new Interface supports another way to do this, but for now this serves our needs.

Delete the **fileSOP**, create a **lineSOP** and connect it to a **resampleSOP**: [TAB] - line ENTER ENTER lays down the line SOP. *Shift-RMB* on its output and type *resa* ENTER ENTER to add the **resampleSOP**. Your network should now look like this:



Resample the line to about 10000 points: disable the “Maximum Segment Length” option and switch to “Maximum Segments”:

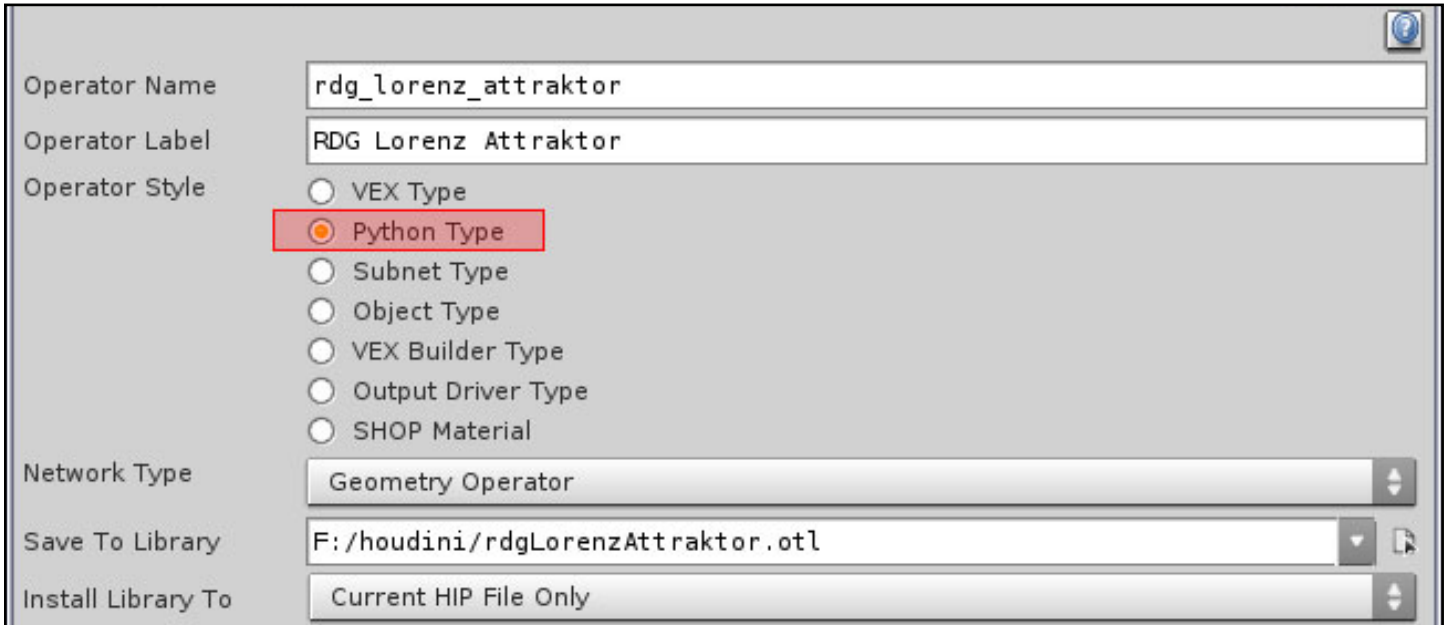


This is the base for our Lorenz Attractor. Save your file.

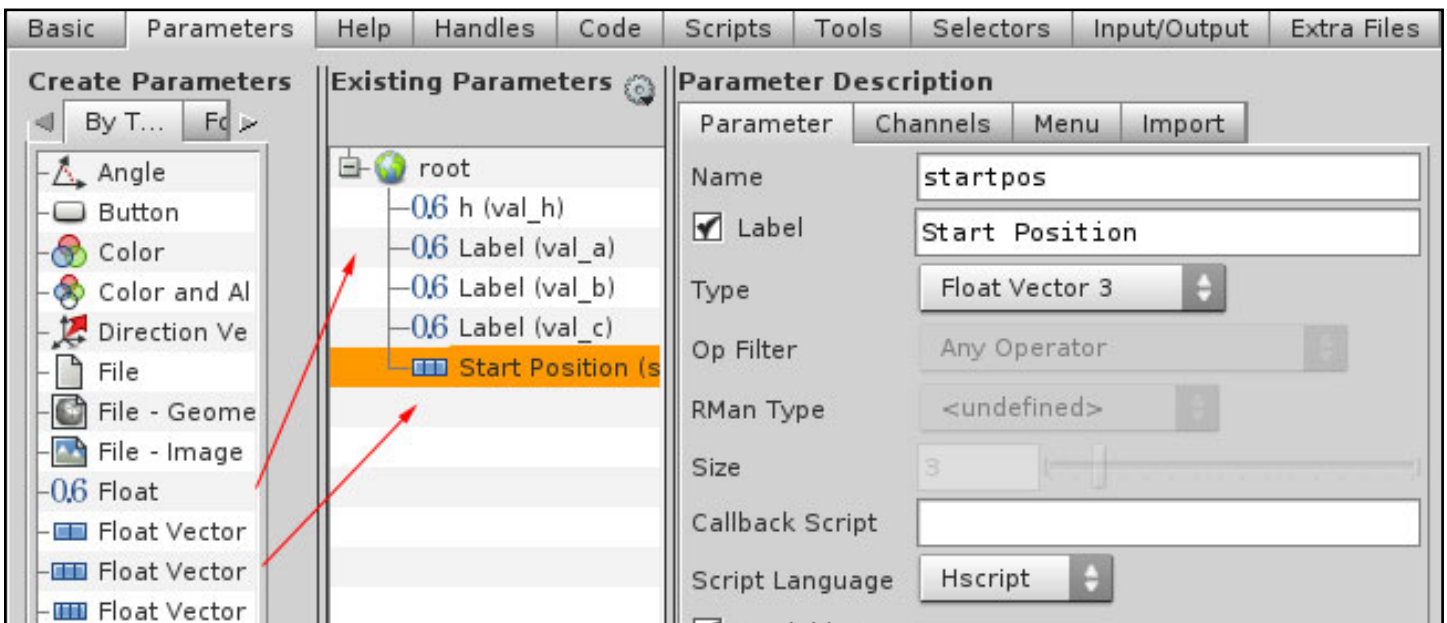
The Python Operator

To start we need to create a new Python Operator: **File -> New Operator Type ...**

Select **Python Operator**. I named my Operator *rdg-lorenz_attraktor*, labeled it as “RDG Lorenz Attraktor” and saved it into a new Operator Type Library (OTL):

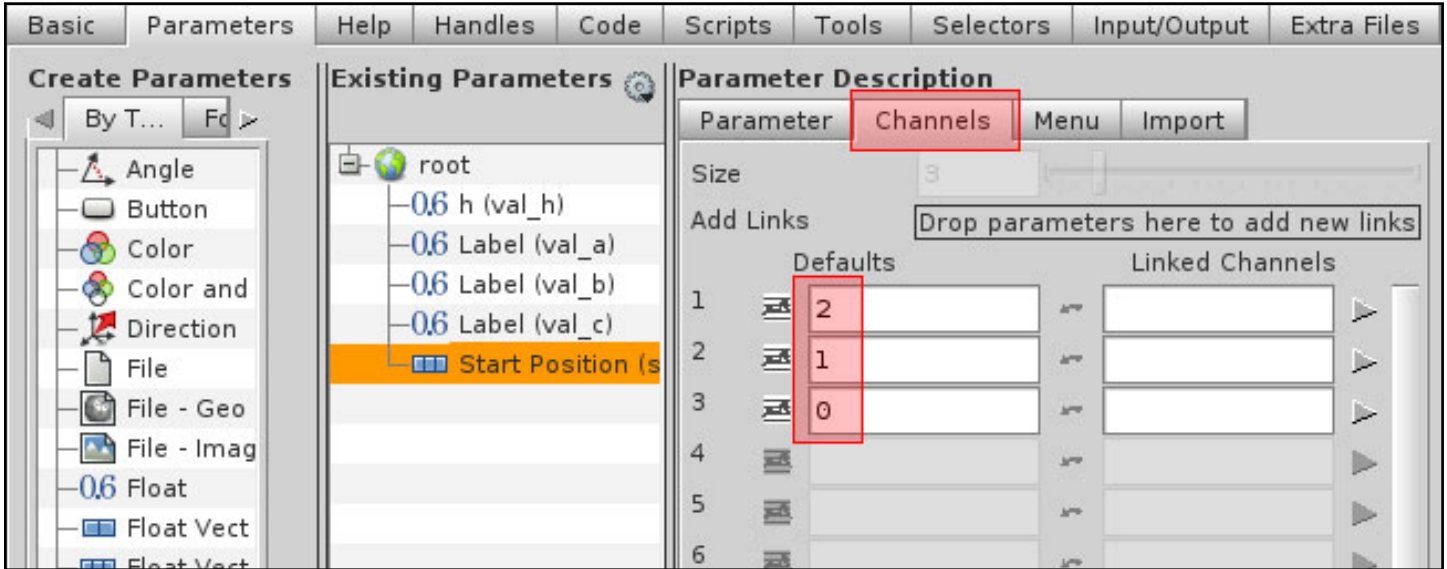


The window to edit the operator type properties appears. Create four float values and one float vector 3. Name them: **val_h**, **val_a**, **val_b**, **val_c** and **startpos**. I labeled them as **h**, **a**, **b**, **c** and **Start Position**.



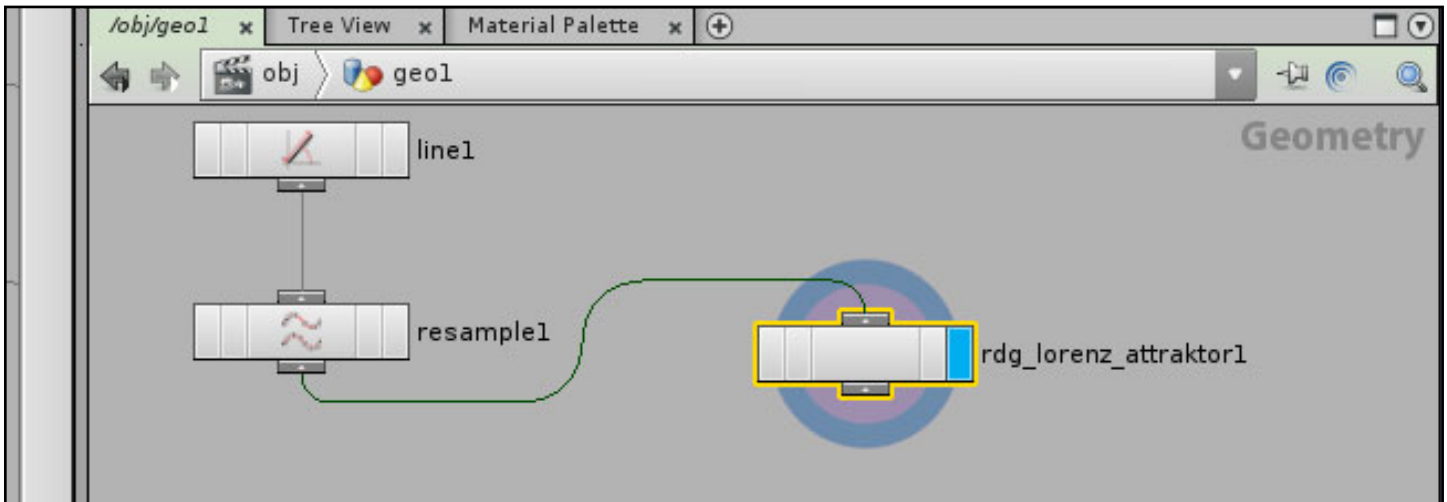
Switch to the **Channels** tab and set the default values for the parameters you just created:

`val_h = 0.01, val_a = 10, val_b = 28, val_c= 2.6667` and `startpos = (2, 1, 0)`



Now click on *Apply* to save your operator. Leave the Edit Operator Type Properties open as we need it in the following steps.

The operator is now installed and you can create an instance: In your network press `[TAB]` `rd` `ENTER` `ENTER` and lay down the new node. Connect the **resampleSOP** to the input of the **RDG Lorenz Attraktor** and set update the display/render flag by clicking on the blue side of the node:



Now it's time to enter the code!

Switch to the **Code** tab and enter this snippet:

```
# This code is called when instances of this SOP cook.
geo = hou.pwd().geometry()
# Add code to modify the contents of geo.
# read the parameters of the node
startpos = hou.Node.evalParmTuple(hou.pwd(), "startpos")
h = hou.Node.evalParm(hou.pwd(), "val_h")
a = hou.Node.evalParm(hou.pwd(), "val_a")
b = hou.Node.evalParm(hou.pwd(), "val_b")
c = hou.Node.evalParm(hou.pwd(), "val_c")
# init some xyz
x1 = 0.
y1 = 0.
z1 = 0.
oldp = startpos
# loop through the point of the SOP
for pt in geo.points():
    # do the lorenz attraktor
    x1 = oldp[0] + h * a * (oldp[1] - oldp[0])
    y1 = oldp[1] + h * (oldp[0] * (b - oldp[2]) - oldp[1])
    z1 = oldp[2] + h * (oldp[0] * oldp[1] - c * oldp[2])
    newpos = (x1, y1, z1)
    # assign the new point position
    pt.setPosition(newpos)
    # save the position for recursive action
    oldp = newpos
```

Python Code:

```
# This code is called when instances of this SOP cook.
geo = hou.pwd().geometry()

# Add code to modify the contents of geo.

# read the parameters of the node
startpos = hou.Node.evalParmTuple(hou.pwd(), "startpos")
h = hou.Node.evalParm(hou.pwd(), "val_h")
a = hou.Node.evalParm(hou.pwd(), "val_a")
b = hou.Node.evalParm(hou.pwd(), "val_b")
c = hou.Node.evalParm(hou.pwd(), "val_c")

# init some xyz
x1 = 0.
y1 = 0.
z1 = 0.
oldp = startpos

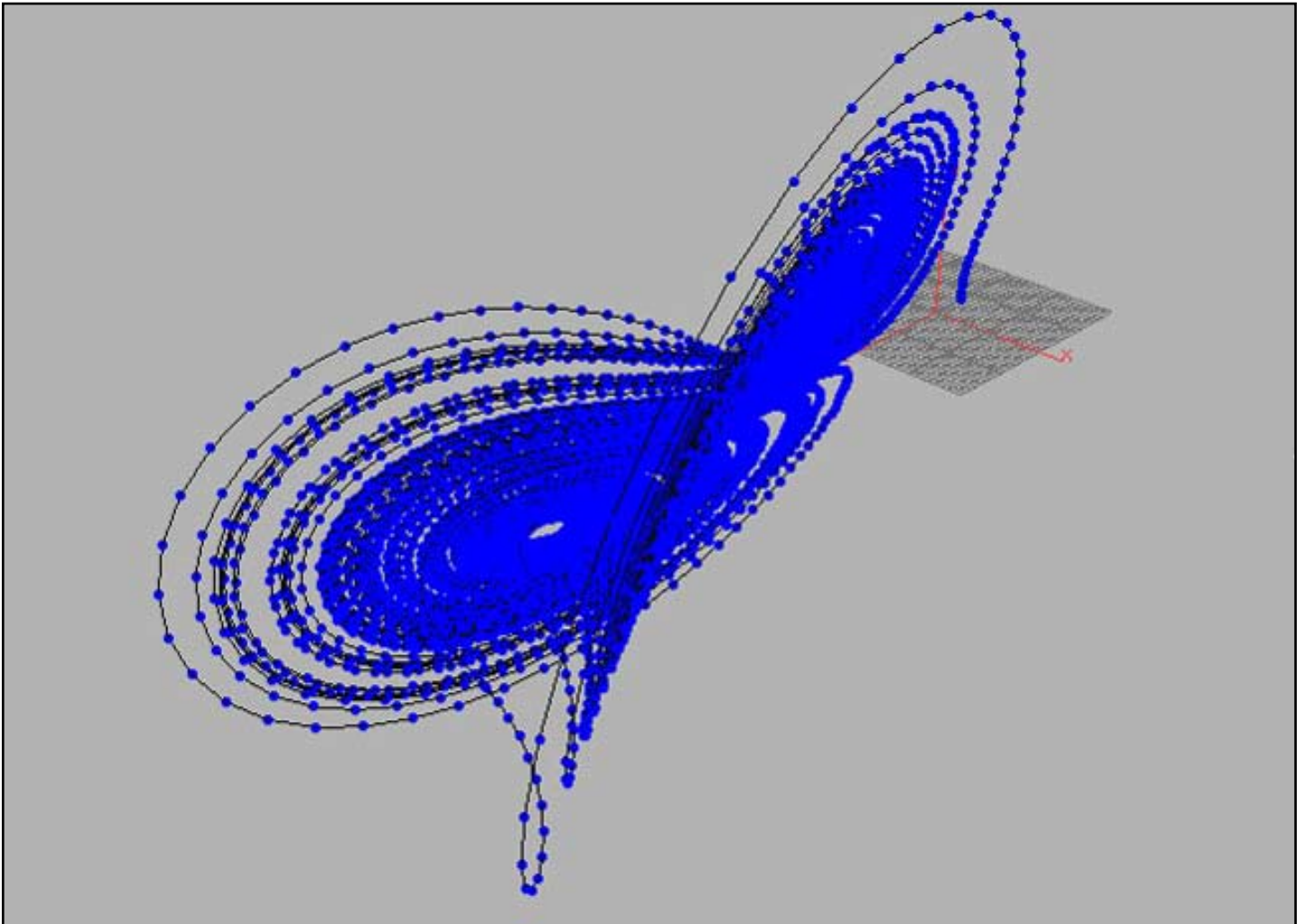
# loop through the point of the SOP
for pt in geo.points():
    # do the lorenz attraktor
    x1 = oldp[0] + h * a * (oldp[1] - oldp[0])
    y1 = oldp[1] + h * (oldp[0] * (b - oldp[2]) - oldp[1])
    z1 = oldp[2] + h * (oldp[0] * oldp[1] - c * oldp[2])
    newpos = (x1, y1, z1)
    # assign the new point position
    pt.setPosition(newpos)
    # save the position for recursive action
    oldp = newpos
```


This might cause some discussion. Jason Iversen kindly posted some different approaches to access parameters with python [Iversen 2007]. But I couldn't follow ... Using some print statements and the incomplete documentation of this first public Houdini 9 beta I kind of reverse engineered the access. But maybe this is just my impression.

hou.pwd() is the python version of **oppwd()** and returns the path to the current operator,
hou.Node.evalParmTuple(node, parm) reads float3 value as tuple,
hou.Node.evalParm(node, parm) reads the float values as floats.

geo.points() returns all the points in the geometry as a list/tuple,
pt.setPosition(newpos) assigns the new position to the point currently processed.

Now it is time to press *Apply* again. Free the view to your Scene Viewer and press *[SPACE]* - *a* to home the view on the attractor. It should look like this:



Play with the parameters to modify the attractor and find the valid value ranges. You can change these in the parameters tab of *Edit Operator Type Properties*.

You can download a finished Operator Type Library for Houdini 9 and sample file at the odforce forum [odforce 2007].

Conclusion

This was just a quick introduction how to modify geometry with Python and the new Python Operator of Houdini 9. You can add icons, help text like in any other digital asset and create some cool new SOPs.

I have chosen the Lorenz Attractor because I really like it [Dümlein 2004] and wasn't able to create one with H8. In the meantime I realized that VEX and a text editor would have been the way.

If you have any suggestions or questions, don't hesitate to contact me: info@preset.de

Muenster, 7/11/2007

Georg Dümlein

References

Bourke, Paul, *The Lorenz Attractor in 3D*, 1997:

<http://local.wasp.uwa.edu.au/~pbourke/fractals/lorenz/index.html>

Iversen, Jason, *Cool new sops*, 2007

http://www.sidefx.com/index.php?option=com_forum&Itemid=172&page=viewtopic&t=8617

Dümlein, Georg, *RDG Pandora*, 2007:

<http://forums.odforce.net/index.php?s=&showtopic=5260&view=findpost&p=37695>

Dümlein, Georg, *lorenz attraktor*, 2004:

<http://www.preset.de/3dsMAX/lorenz/>