

## Apophysis: Xaos – Adding an Extra Dimension of Control

The new xaos feature introduced in Apophysis 2.08 beta isn't as mysterious as it might first seem. For those interested, we'll take a brief look at the theory, then move on to see how it relates to the more familiar parameter of transform weight.

### Theory

It's always worth reminding ourselves of the process of flame generation via the Chaos Game. I shan't repeat the information here, rather refer to a graphical demonstration that really opened my eyes as to how it works:

<http://apophysis.wikispaces.com/Flame+Maths+101>

Drawing straight lines obviously emulates the basic linear variation. Introducing other variations into a given transform alters the way *it* transforms the space – the lines are no longer straight. So each transform contributes to the overall distortion of the space, which in turn determines where the point will land. The contribution to the spatial distortion has hitherto been determined by a simple parameter, the transform weight.

With the introduction of xaos, an extra level of fine-tuning is available: instead of the mutual interaction of the transforms resting on the 1-dimensional quantity of transform weight, we have the extra dimension of individual interactions between transforms, hence the number of control parameters increases from  $n$  to  $n^2$ , where  $n$  is the number of transforms.

### The numbers

The following tables show the process of moving from simple transform weights to xaos-based interactions, from a single-column table of weights, via the matrix of xaos coefficients, to a new matrix of weight-coefficients.

Note that From and To are just different views of the same information. Each is an  $n \times n$  matrix, where  $n$  is the number of transforms, and simply a transposition (reflection in the diagonal) of the other. The example given uses entirely arbitrary numbers:

**Weights**

<b>Transform</b>	<b>Weight</b>
<b>1</b>	0.37
<b>2</b>	0.28
<b>3</b>	0.35

**Xaos coefficients**

<b>From</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	1	0	0
<b>2</b>	1	1	0
<b>3</b>	0.1	2	0.2

First, we'll show the From/To relation - each is simply the other transposed, that is reflected about the diagonal.

<b>To</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	1	1	0.1
<b>2</b>	0	1	2
<b>3</b>	0	0	0.2

Now, we'll multiply each column by the weights for the From view.

<b>From</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	0.37	0	0
<b>2</b>	0.28	0.28	0
<b>3</b>	0.035	0.7	0.07

Finally, we'll show the mutually affective weights matrix in the To view.

<b>To</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	0.37	0.28	0.035
<b>2</b>	0	0.28	0.7
<b>3</b>	0	0	0.07

**In practice**

So much for the theory, how to actually use it? One way to illustrate this would be to ask questions:

"How much do I want *this* transform to affect *that* transform?" - use 'To'

"How would I like to see the influence distribution of *these* transforms on *this* transform?" - use 'From'

The 'Add linked xform' is simply a shortcut method of adding a new transform influenced solely by whichever transform is currently selected in the Editor e.g. if xform #2 is selected and a linked xform (#4) generated, it will show, in 'From' mode, xaos = 1 from #2 and 0 from the rest. For further information on linked transforms:

<http://polyfractalisation.blogspot.com/2009/02/linked-transforms-post-transforms-on.html>

Note that the default value of 1 for all xaos coefficients corresponds to the original weights-only scenario. Hence a flame with any non-unit xaos values cannot be reproduced correctly by pre-2.08 beta versions.