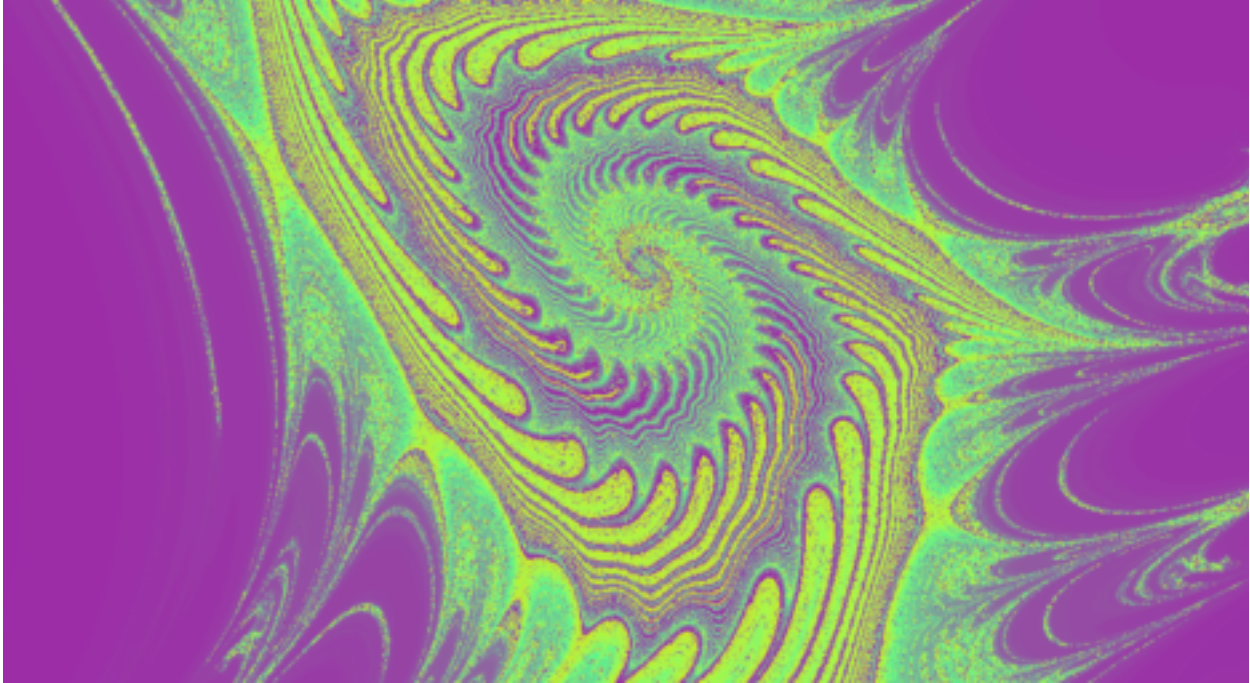


Welcome to Synthimax!



Introduction

Synthimax is an application for synthesizing images. It is based on mathematical functions, but you don't need to be a math geek to use Synthimax to make gorgeous pictures and movies.

The easiest way to learn to use Synthimax is to just play around with it. Choosing generators, typing in new values for parameters, fiddling around with the gradient, will give you an idea of the possibilities of the program. This manual tells you both what you can do (i.e. choose one of many normalizings) and a bit about how this is done. This "how" is mostly of interest to math geeks (like the program's author). If you can't make sense of the mathematical formulas in the manual, it won't prevent you from using Synthimax anyway.

Table of Contents

Installing Synthimax	5
Registering Synthimax	6
Basic Workflow (How to Use Synthimax)	7
The Main Window	8
The Generator Pane	8
Image Generators	11
Fractals	11
Orbit Method Popup	11
Quadratic Basins	13
Halley Maps	15
Complex Feedback	16
Square Root Tracing	18
Gumowski Tracing	19
Textures	20
Mandala	20
Iterated Sines	20
Polynomial Quotient	21
Perlin Noise	21
Dynamical Systems	23
Square Root	24
Sine minus Cosine	24
Sine Powers Mixed	24
Cos xy + Sine Powers	24
Hopalong	24
Gumowski	24
Phase Plane	26
Line Art	27
The View Pane	28
Center	28

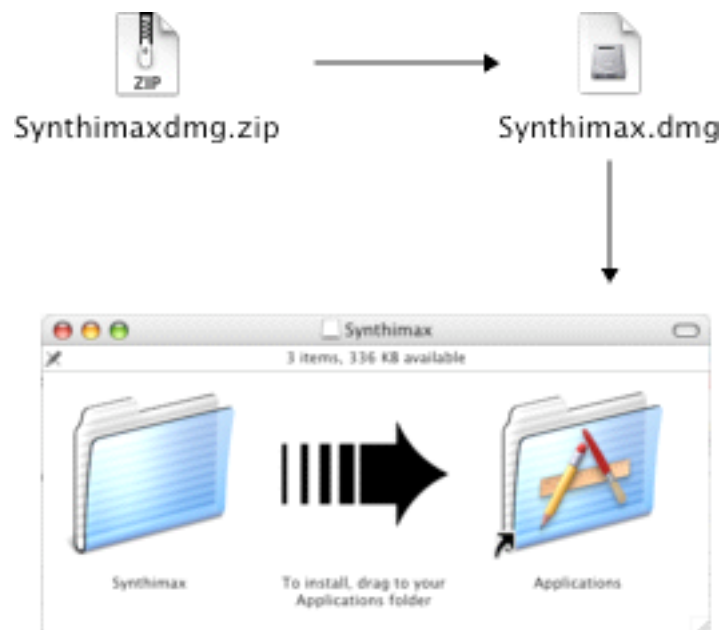
Radius	28
Rot°	28
Flip Buttons	28
Zoom Buttons	28
Reset Button	29
Geometries	30
The Drawing Pane	32
Filter Popup and Filter Parameter Text Box	32
Normalizing Popup and Norm Text Box	35
Adjust Button	38
Auto-Adjust Checkbox	38
Floor and Ceiling	38
Gradients	39
Factor and Offset	39
Edit Gradient	40
Store Gradient	41
Gradients	42
Randomize Gradient	42
Factor	42
Offset	42
The Timeline Pane	43
Adding, Moving and Deleting Keyframes	44
Activate Keyframe	44
Replace Keyframe	44
Saving and Loading Movies as Keyframe Sets	45
Saving Pictures	45
Rendering Movies	45
Rendering a Movie	46
Space Considerations	47
Toolbar Controls	48
Randomize Configuration Button	48

Randomize Gradient Button	49
Previous/Next Configuration Buttons	49
Store Configuration Button	49
Configurations Button	49
Draw Button	49
The Image Window	51
Flip Horizontal Button	51
Flip Vertical Button	52
Zoom In Button	52
Zoom Out Button	52
Clicking to zoom and move the image	53
The Gradients Window	54
The Configurations Window and Configurations Files	56
Credits	59

Installing Synthimax

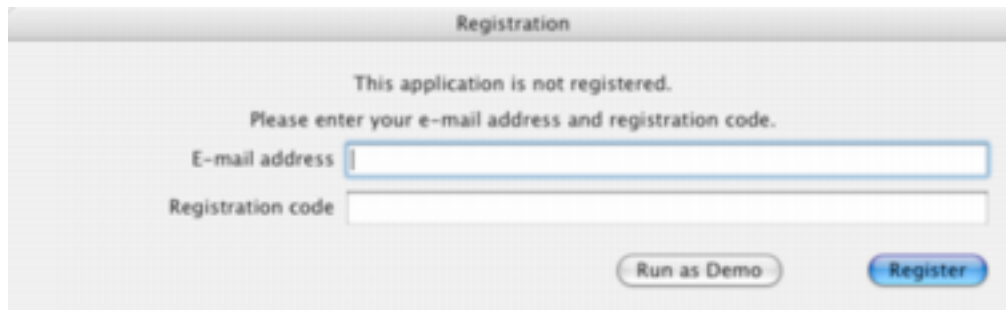
To obtain Synthimax, download the disk image from the official site: www.synthimax.com. You'll get a “zipped” disk image, `synthimaxdmg.zip`. Double-click on the file to expand it into `synthimax.dmg`, then double-click on `synthimax.dmg` to mount the disk image.

Once the disk image is mounted, just drag the “Synthimax” folder to the Applications folder to install it. (This folder is an alias to the Applications folder for your computer.)



Registering Synthimax

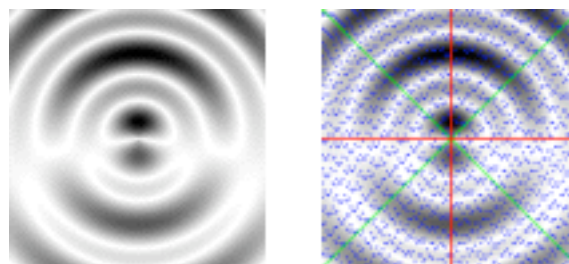
When you start Synthimax, it displays the following registration dialog.



Enter your e-mail address and the registration code you received when you purchased the license and click “Register”. Your application is now registered and you won’t see this dialog again.

If you haven’t yet bought a license, you can run the demo version of the application. This version is almost fully functional, so that you can get an idea of Synthimax before deciding to purchase it. The demo version cripples the program in three ways:

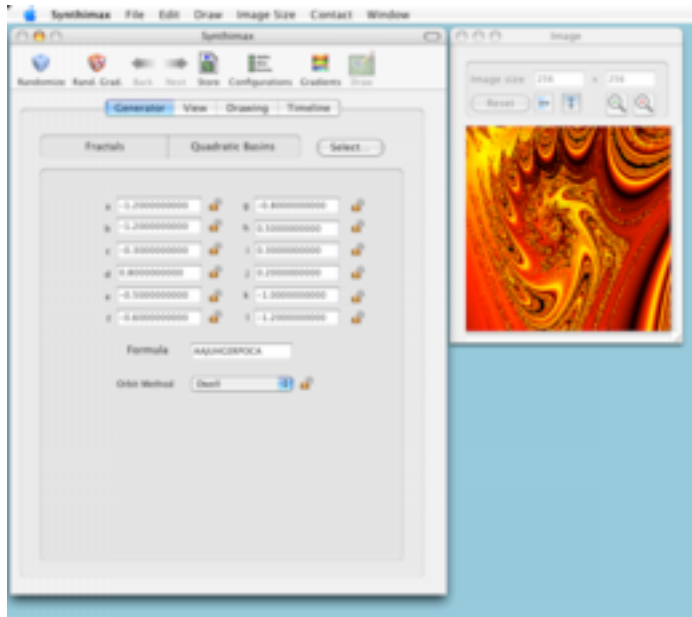
- You can’t save pictures.
- You can’t render movies.
- All images are degraded. You can still see what the image should look like, but it is overlaid with “snow” and colored lines to make it unusable as is.



Until you register Synthimax, the registration dialog will appear each time you start the program.

Basic Workflow (How to Use Synthimax)

Synthimax initially displays two windows: the main window and the image window.



You will start working in the main window, where you can select the kind of image you want and specify its various parameters. Once you have chosen the parameters, you can click the Draw button to draw the image.

You will frequently need to adjust the position or the size of the image you produce. To do this, you turn to the image window, where you can zoom in and out and move the image around by clicking and dragging.

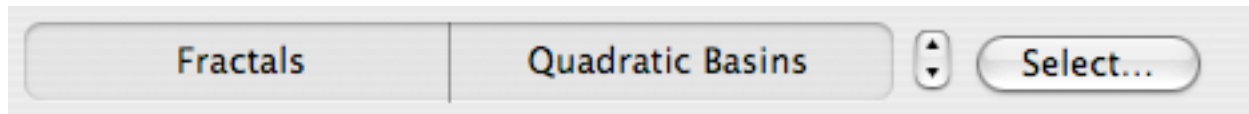
If you like an image, you'll want to save it. The File menu allows you to save the image. However, this only saves the picture itself. To remember how you produced the image, you'll need to save the "recipe" for it, which is called a *configuration*, in a configurations file.

The Main Window

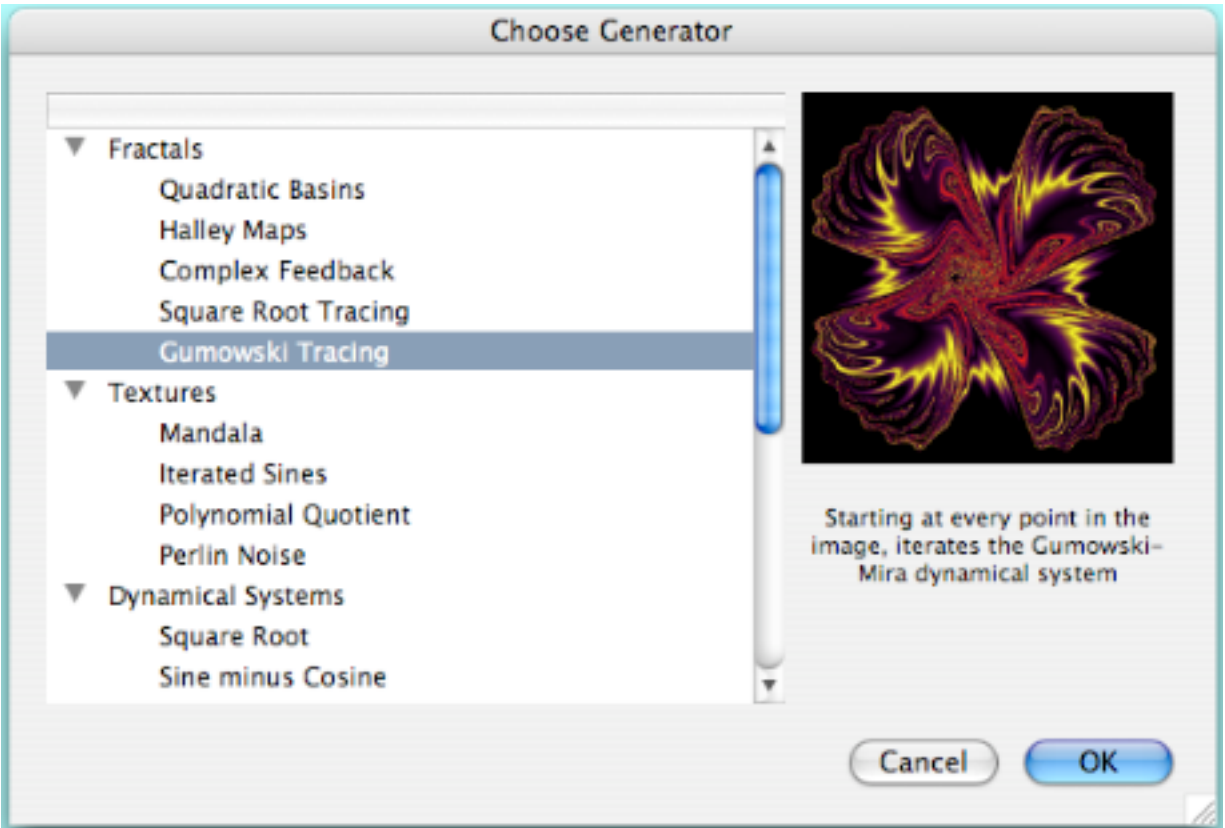
The main window has a toolbar at the top, and beneath the toolbar a group of four tabs. Clicking on one of the tabs reveals the corresponding pane. The panes are: the *Generator* pane, to select an image generator, the *View* pane, to determine the view rectangle and the geometry for the image, the *Drawing* pane, which is where filters, normalizing and gradients are controlled, and the *Timeline* pane, which is used for creating animations.

The Generator Pane

Making images in Synthimax starts with choosing an *image generator*. You do this by using the controls at the top of the pane.



To the left of the pane are two text boxes that display the category for the generator and the generator's name. If you click on the arrow control to their right, you can step through the list of available generators. Since there are over 20 generators, it may be faster to click on the Select... button, which brings up a selection dialog. The dialog displays some information on each generator along with a typical image for it to help you decide on what you want. Once a generator is chosen the box below the controls is updated



Each generator has a set of *parameters*, which are displayed in the box below the selection controls. You can edit these parameters by typing values into the text boxes or choosing values using buttons or menus. The output of an image generator is called the *raw image*. It is a table of numbers. Each number in the table is called an *element*.

Image generators evaluate mathematical functions anywhere over the (x,y) plane. Since that plane is infinite, you need to tell the program which part of it you want to see. This is done in the View pane, where you specify the center and radius of the view. You can also specify a rotation angle to rotate the image around its center point. Finally, two pushbuttons allow you to flip the image top-to-bottom and/or left-to-right.

Image generators actually compute values over the plane, not images as

such. To turn the output of a generator into a picture, it must be processed pixel-by-pixel into an image. The controls in the Drawing pane allow you to specify how this is done.

When entering parameters, you can hit Tab to go from one text box to the next. Shift-Tab takes you to the previous text box. If you hit Enter or Return in a text box, the image will be immediately computed and drawn.

Image Generators

The image generators in Synthimax fall into several different categories: fractals, textures, dynamical systems, phase plane and line art.

Fractals

Fractal image generators iterate a mathematical function over each point of the plane and keep track of its successive values, until a termination condition is reached. The set of successive values is called the *orbit* for that point.

The value of an element of the raw image is derived from the orbit of its corresponding point. You can control how this value is obtained, by making a selection from the orbit method popup.

Orbit Method Popup

This control appears for every fractal image generator. It allows you to choose how the orbit is analyzed to produce a value.

We denote the orbit of a point by the series: $\{z_0, z_1, \dots, z_n\}$ where z_0 is the original point. Each orbit method produces a final value according to this description:

- **Dwell:** Returns n , the size of the orbit.
- **Distance:** Returns the distance between z_0 and z_n .
- **Length:** Returns the length of the orbit = $\text{dist}(z_0, z_1) + \text{dist}(z_1, z_2) + \dots + \text{dist}(z_{n-1}, z_n)$.

- **Center of Gravity:** Returns $v = \left\| \frac{1}{n} \sum z_i \right\|$.
- **Closeness:** Returns $v = \frac{\|z_1 - z_0\|}{\|z_n - z_0\|}$.
- **Angle:** Returns the argument of $z_n - z_0$ — basically its angle with the horizontal axis.
- **Harmonic Center:** Returns $-z_0 + z_1 + 1/2 z_2 + 1/3 z_3 + \dots + 1/n z_n$
- **Real:** Returns the absolute value of the real part of z_n .
- **Imaginary:** Returns the absolute value of the imaginary part of z_n .
- **Potential:** Returns $v = \frac{\|z_n\|}{n}$.
- **Min. Distance:** Returns $v = \min\{\|z_i\|\}$.
- **Cross Trap:** Examines all points after the first. If any of them has a real or imaginary part, α , within 0.01 of 0, the method returns $0.01 - \alpha$. This produces variously distorted intersecting stalks (make sure to adjust the norm, otherwise you won't see a thing!).

Quadratic Basins

This generator computes general quadratic map basins, as described in Clifford Pickover's article "Automatic Generation of General Quadratic Map Basins", in *Computers and Graphics*, Mar/Apr 1995.

For every (x,y) point in the image, the generator iterates these functions:

$$x' = a + bx + cx^2 + dxy + ey + fy^2$$

$$y' = g + hx + ix^2 + jxy + ky + ly^2$$

Where a, b, \dots, l are the parameters. This process is repeated until the current values of x and y are outside a circle of radius 1000 from the origin (0,0) or the iteration limit (1024) has been reached.

The parameters box contains a textbox called "formula". This is mostly a historical legacy, for backward-compatibility with the *EscapeFractals* application I developed for the Classic MacOS in the distant past. That program used a "formula" (as described by Mr. Pickover) to control the parameter values, as follows:

A = -1.2 B = -1.0 C = -0.8 ... X = +1.0 Y = +1.2

If you enter a 12-letter formula in the text box and hit Enter, the parameters will be updated according to the formula. Conversely, when you modify the parameters, the corresponding formula, if there is one, will be displayed.

a	<input type="text" value="-1.2000000000"/>		g	<input type="text" value="-0.8000000000"/>	
b	<input type="text" value="-1.2000000000"/>		h	<input type="text" value="0.5000000000"/>	
c	<input type="text" value="-0.3000000000"/>		i	<input type="text" value="0.3000000000"/>	
d	<input type="text" value="0.8000000000"/>		j	<input type="text" value="0.2000000000"/>	
e	<input type="text" value="-0.5000000000"/>		k	<input type="text" value="-1.0000000000"/>	
f	<input type="text" value="-0.6000000000"/>		l	<input type="text" value="-1.2000000000"/>	

Formula

Orbit Method

Halley Maps

This generator displays the behavior of a root-finding approximation method for a polynomial, known as Halley's method (hence the name). The polynomial is determined by the coefficients input in the parameters box. The parameter λ is a factor that governs the size of the step for each iteration of Halley's method. Setting it larger than 1 will make the picture more chaotic; setting it smaller will "smooth it out" to some extent, but the effects are not necessarily straightforward.

Each pixel in the image corresponds to a complex number. The generator applies the Halley method to obtain a root of the polynomial starting from that number. The iteration stops when a root has been reached or the iteration limit (32) has been attained.

You can choose one of three convergence tests, which affect the determination of when a root has been reached. They have pronounced effects on the overall picture, particularly the "branches" test.

The image shows a control panel for the Halley Maps generator. It features a grid of input fields for polynomial coefficients, a parameter λ , and dropdown menus for convergence test and orbit method. Each input field has a lock icon to its right.

Powers of z	
c	0.0000000000
1	-1.0000000000
2	0.0000000000
3	0.0000000000
4	0.0000000000
5	0.0000000000
6	0.0000000000
7	1.0000000000
8	0.0000000000
9	0.0000000000
10	0.0000000000
11	0.0000000000
12	0.0000000000

λ 1.0000000000

Convergence Test: Standard
Orbit Method: Dwell

Complex Feedback

This generator iterates $z \leftarrow f(z) + c$ where both z and c are complex numbers. The text boxes allow you to determine the function $f(z)$ by specifying the coefficients for powers of z , $\sin z$, e^z and z^z . The function is iterated until either $\|z\| > 100$ or the iteration limit has been reached.

The well-known Mandelbrot set is produced when setting $f(z) = z^2$; $z_0 = (0, 0)$; and varying c over the plane. Conversely, you produce Julia sets by setting c constant and varying z_0 over the plane. But there are more possibilities: since z_0 and c are both complex variables, there are a total of four degrees of freedom. It is possible for example to set the real parts of z and c , and let their imaginary parts vary over the plane.

The two parameters θ and ϕ control the view plane, which is a generalization of the above idea. If both are zero, the generator produces Mandelbrot sets: z_0 is set by the parameters a and b , while c varies over the plane. If $\theta = 1$, $\phi = 0$, it produces Julia sets: c is set by parameters a and b , while z_0 varies over the plane. If $\phi = 1$, $\theta = 0$, the imaginary parts are fixed: a sets $\text{Im}(c)$ and b sets $\text{Im}(z_0)$, while the real parts vary over the plane. If $\phi = 1$, $\theta = 1$, the real parts are fixed: a sets $\text{Re}(c)$ and b sets $\text{Re}(z_0)$, while the imaginary parts vary over the plane. Other values for ϕ and θ morph between those extremes.

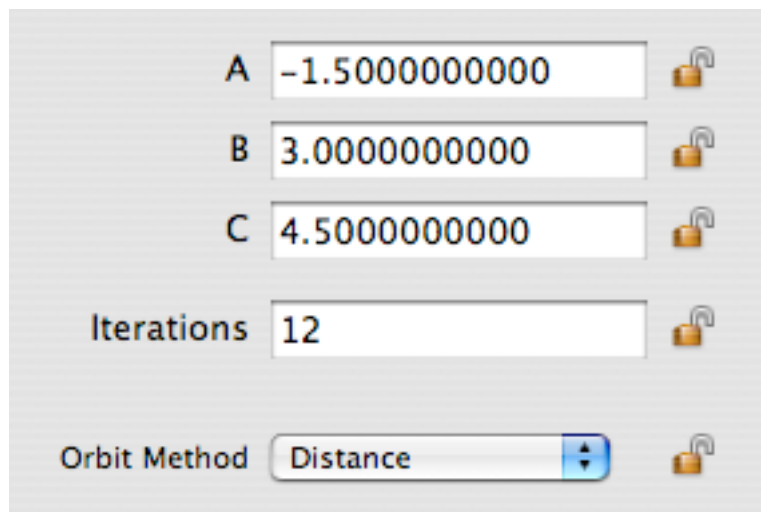
Clicking the Biomorphs checkbox adds elongated spikes to the image. Those are produced by an additional test on the orbit: if at the end, $|\text{Im}(z_n)| < 100$ or $|\text{Re}(z_n)| < 100$, then the orbit is truncated to its first element, z_0 .

z	<input type="text" value="0.0000000000"/>		sin z	<input type="text" value="0.0000000000"/>	
z^2	<input type="text" value="1.0000000000"/>		e^z	<input type="text" value="0.0000000000"/>	
z^3	<input type="text" value="0.0000000000"/>		z^z	<input type="text" value="0.0000000000"/>	
z^4	<input type="text" value="0.0000000000"/>				
z^5	<input type="text" value="0.0000000000"/>		iterations	<input type="text" value="32"/>	
z^6	<input type="text" value="0.0000000000"/>				
View plane			Parameters		
θ	<input type="text" value="1.0000000000"/>		a	<input type="text" value="0.5000000000"/>	
ϕ	<input type="text" value="0.0000000000"/>		b	<input type="text" value="0.0000000000"/>	
<input checked="" type="checkbox"/> Biomorphs					
Orbit method		<input type="text" value="Dwell"/>			

Square Root Tracing

This generator iterates the “Square Root” dynamical system (see page 16) for a fixed number of iterations, starting from every point in the image’s view rectangle. The images produced look like a chaotic overlay on top of a regular structure; this becomes more and more true as you increase the number of iterations.

Note that if you use the “Dwell” orbit method with this generator, you will get a blank image, since every point is iterated the same number of times.

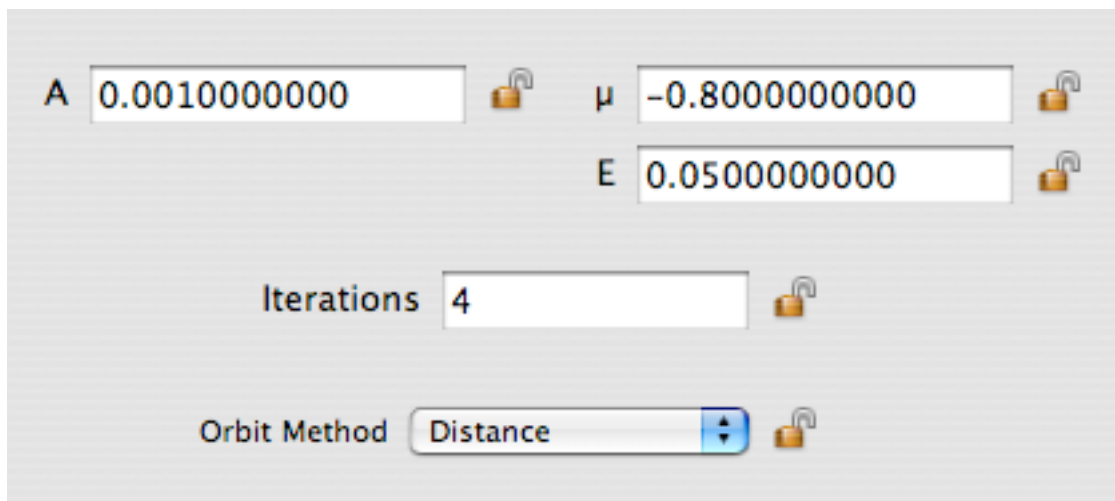


The image shows a control panel with five rows of input fields, each with a lock icon to its right. The first row is labeled 'A' and contains the value '-1.5000000000'. The second row is labeled 'B' and contains '3.0000000000'. The third row is labeled 'C' and contains '4.5000000000'. The fourth row is labeled 'Iterations' and contains '12'. The fifth row is labeled 'Orbit Method' and contains a dropdown menu with 'Distance' selected. The lock icons are currently open, indicating that the settings are not locked.

Gumowski Tracing

This generator iterates the “Gumowski” dynamical system (see page 16) for a fixed number of iterations, starting from every point in the image’s view rectangle. The images produced are often similar in shape to those produced by the dynamical system, but they have a very different quality.

Note that if you use the “Dwell” orbit method with this generator, you will get a blank image, since every point is iterated the same number of times.



The image shows a control panel for the Gumowski Tracing generator. It features five input fields, each with a lock icon to its right:

- A: 0.0010000000
- μ : -0.8000000000
- E: 0.0500000000
- Iterations: 4
- Orbit Method: Distance (selected from a dropdown menu)

Textures

Unlike other generators, textures do not involve deeply iterated functions. Because of this, textures are computed much faster than other kinds of images.

All textures share a special parameter: this is the multiplicand. The final result of the computation is multiplied by this factor. This parameter is relevant when you're using the Square Root Remainder filter and/or the Modulo or Fuzz normalizing. With other filters and normalizers, adjusting the norm by the same factor as the multiplicand will give identical results (i.e. mult = 20.0, norm = 10.0 is the same as mult = 100.0, norm = 50.0).

Mandala

This texture comes from the Mandala program in the book *What To Do After You Hit Return* (People's Computer Company, 1975) and is included here as an homage. The original program drew its picture using \$, @, * and . symbols. Let's just say we've come forward a bit in thirty years.

Mandala computes the surface described by a line sweeping 360° around the origin. The line dips a certain number of times over the course of this orbit, and it's shaken by ripples. You can specify frequency and amplitude for both these parameters, and a combination influence as well. If you choose a non-integer value for the dip frequency, you'll get a discontinuity along a line from (0,0) projecting left: this is normal.

Iterated Sines

At every point of the view rectangle, this generator computes an

iterated sine function of the form $f(t) = \sin(at + \sin(bt) + \phi)$ for three values: x , y , and the product xy . It then adds these three values, each with a weight factor and divides by the sum of all three weights.

Polynomial Quotient

This generator computes two polynomials in x and y and divides one by the other. In order to create repeating patterns, the generator takes minimum and maximum values for x and y . Outside these values, the coordinates are “mirrored”, which produces symmetrical textures.

Perlin Noise

Ken Perlin devised an algorithm to produce a form of noise which could be used to produce all sorts of interesting textures. While the applications of Perlin Noise are vast, here we are dealing with a limited approach. The image generator contains three separate Perlin noise generators, whose output may be combined in different ways.

Each of the noise generators has three parameters: scale, octaves, and persistence. Scale determines the “size” of the noise. The lower the value, the coarser the effects of the noise. Octaves affects the level of detail of the noise. With a single octave, the noise generator produces blobby shapes; increasing the number of octaves results in much more detailed, feathery noise. Persistence also increases detail in the noise. Persistence is a factor applied to every successive octave of noise, damping its effects. A persistence of 1.0 means every octave of noise has as much effect as the first octave. A persistence of 0.1 means the second octave has 10% of the influence of the first, the third 1%, etc. Clearly, persistence is irrelevant unless you have at least 2 octaves.

The output of the three generators is combined in a way indicated by the “combination” pop-up menu. The value of the “scale” parameter is fixed to 1.0 for generator a because, if you use a single generator, doubling the scale gives the same result as doubling the radius of the view rectangle. When using two or three noise generators, their relative scales matter, so generators b and c’s scale parameter is free to vary.

	a	b	c
Scale	1.0	<input type="text" value="0.20000"/>	<input type="text" value="0.50000"/>
Octaves	<input type="text" value="5"/>	<input type="text" value="4"/>	<input type="text" value="5"/>
Persistence	<input type="text" value="0.50000"/>	<input type="text" value="0.70000"/>	<input type="text" value="0.60000"/>
Combination	<input type="text" value="a"/>		
Multiplicand	<input type="text" value="1000.00000"/>		
Conversion	<input type="text" value="linear"/>		

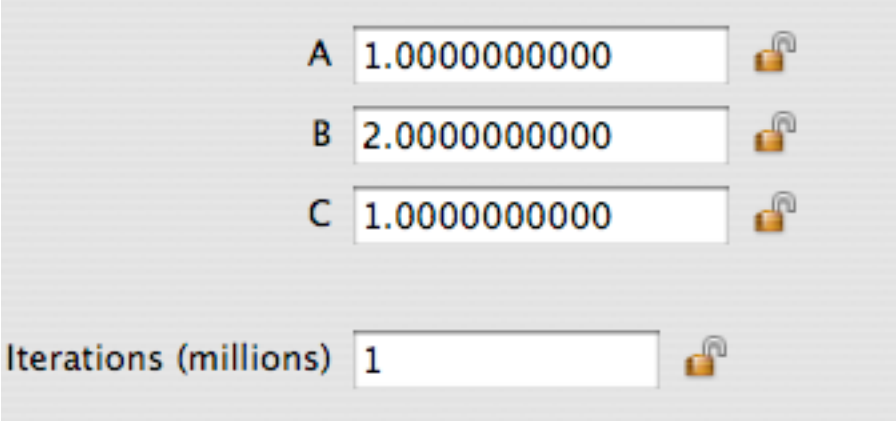
Once the combination of the three noise generators has been computed, it is multiplied by the multiplicand parameter and converted by one of three techniques. “Linear” gives an image where all of the pixels fall around the middle of the gradient. “Absolute value” will produce an image where pixels range all the way along the gradient. “Cutoff” produces the same image except that roughly half of it has been clamped down to 0. (Math geeks will have realized this corresponds to three different ways of converting the range [-1,1] to [0,1].)

Dynamical Systems

Dynamical system image generators work differently from others: what they do is iterate function systems from a fixed starting point (typically $x = 0, y = 0$) and keep track of the successive values taken by the variables x and y . This makes the images they generate behave in a different fashion from fractals or textures.

Namely, unlike what happens with other kinds of generators, if you magnify a dynamical system image, it will get fainter. This is because the greater the magnification, the fewer values of the system “fall into” one of the raw image elements. You will need to decrease the norm or increase the number of iterations to improve the detail when magnifying an image.

Dynamical systems have a number of parameters. The equations being iterated are displayed in the box. The number of iterations can be set to the nearest million (or in one case, thousand).



The image shows a control panel for a dynamical system generator. It features four input fields, each with a label to its left and a lock icon to its right. The first three fields are labeled 'A', 'B', and 'C', and each contains the value '1.0000000000'. The fourth field is labeled 'Iterations (millions)' and contains the value '1'. All fields have a white background and a thin grey border.

A	1.0000000000	🔒
B	2.0000000000	🔒
C	1.0000000000	🔒
Iterations (millions)	1	🔒

The behavior of dynamical systems is sometimes much simpler than at other times: in other words, some combinations of parameters will result in an image consisting of only a handful of dots.

Square Root

This system uses a pair of equations involving square roots and trigonometric functions.

Sine minus Cosine

This dynamical system was taken from Clifford Pickover's article "Million-Point Sculptures" in *Computer Graphics Forum* vol 10, p.333 (1991).

Sine Powers

This dynamical system is a generalization of a system presented in Clifford Pickover's original article.

Sine Powers Mixed / Cos xy + Sine Powers

These dynamical systems are variations on a theme, playing with trigonometric functions.

Hopalong

This dynamical system is very different from the others: it generates intricate images with strong symmetry. It uses equations discovered by Barry Martin and initially presented in A.K. Dewdney's column "Computer Recreations" in *Scientific American*, September 1986.

The parameters for this system include the *coloring method*: you can choose to color a pixel according to the number of points that were sampled there, or according to the iteration number of the last point to be sampled there. In the first case, the image you get represents the "density" of the dynamical system in the neighborhood of those coordinates; in the second, it reflects at what time this neighborhood was last visited.

Gumowski

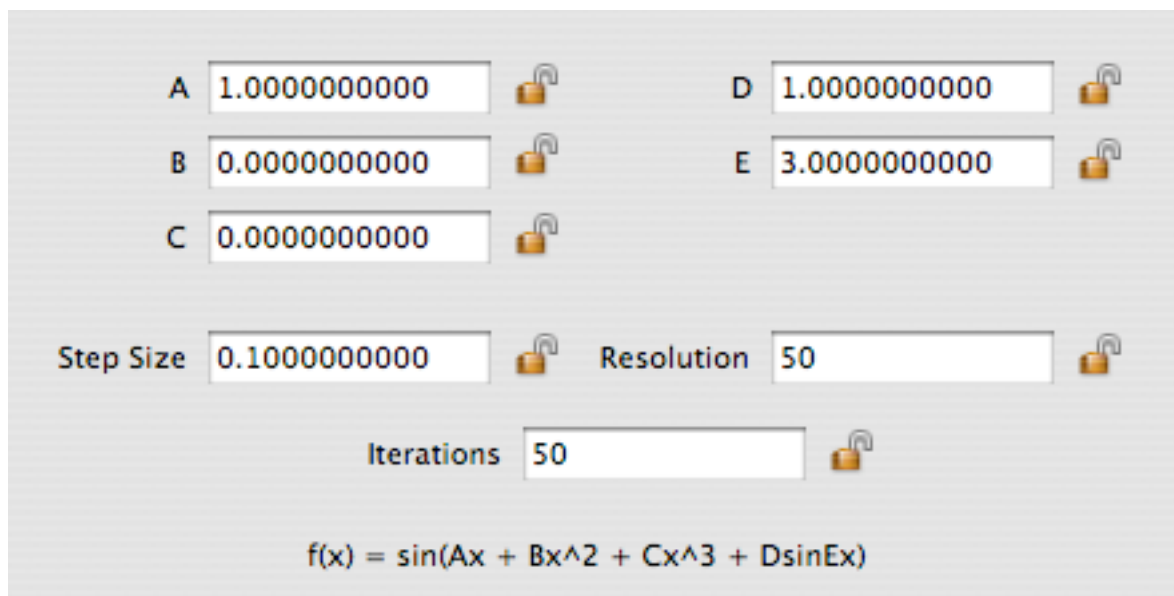
This dynamical system is based on work by I. Gumowski and C. Mira.









The Gumowski attractor is very sensitive to values of A and μ . The E parameter tends to affect details of the final image rather than its overall shape. The attractor is not always well-behaved for artistic purposes and can settle into a boring cycle after tens of thousands of iterations. For this reason, when using the randomize function, you may produce images that appear reduced to a handful of points. In that case, try reducing the number of iterations from 500 thousand to 50 thousand or even 5 thousand: you may well be surprised by the picture that is suddenly revealed. This advice actually applies to other images as well: choosing a smaller number of iterations can produce an image with a very different feel.

Phase Plane

Phase plane generators are similar to Dynamical system generators, except that they follow the paths of many initial points, arranged in a regular grid.

All Phase plane generators use three common parameters: step size, resolution and iterations. Step size affects how far each point travels when the computations are performed. Values between 0.01 and 0.5 are best. Higher values make for a much “airier” image. Resolution determines how many points are in the grid. With a resolution of 50, the grid will be 51 points high by 51 points wide. Iterations sets the number of steps taken for each point. The larger this value, the darker and fuller the image will be. Very small values will result in clearly separated individual trajectories.



A 1.0000000000  D 1.0000000000 
B 0.0000000000  E 3.0000000000 
C 0.0000000000 
Step Size 0.1000000000  Resolution 50 
Iterations 50 
 $f(x) = \sin(Ax + Bx^2 + Cx^3 + D\sin Ex)$

Line Art

Line Art is an ad-hoc category for generators that produce images by drawing lines.

Grid

This generator draws horizontal and vertical lines. You can control the spacing in both directions. The *thickness* parameter specifies the width of the part of the line that is set to the maximum value (black with the traditional gradient). The *edge* is the width of the line that fades to white, and the *sharpness* is how fast this fading occurs. If edge is less than thickness, you will get razor-sharp flat-looking lines.

The menu allows you to choose whether you want lines in one dimension only, lines that cross by fusing into each other (crossing), lines that cross by superposition (blend), or dots. For dot size, only the vertical parameters are used. Set thickness to zero to get fuzzy dots.

Activating the checkerboard option will fill in alternate squares with black.

Curves

This generator plots parametric curves. Up to three trigonometric functions of the form $x = \sin(at + \phi)$ are combined for each coordinate. The larger the value of a , the more complicated the curve. ϕ can be any value between 0 and 1. Setting it to 0.0 or 0.5 will make cleaner curves. The number of points determines how well the curve will be drawn. If “Continuous curve” is unchecked, only individual points are drawn. If you animate this generator, the curves will appear to rotate in three dimensions. Finally, note that geometries do not work with this generator, because it needs to draw line segments in the image directly.

The View Pane

This pane is where you specify how the image is viewed. The top controls determine the view rectangle, while the bottom box controls the geometry.

Center

Specifies the center of the image. The first box specifies the horizontal coordinate, and the second box specifies the vertical coordinate.

Radius

Specifies the distance between the center of the image and the nearest edge. If the image is wider than it is tall, the radius is one-half of the image height. If the image is taller than it is wide, the radius is one-half of the image width.

Rot °

Specifies the rotation in degrees. The image is rotated counterclockwise (if you flip the image, this also reverses the effect of the rotation).

Flip Horizontal Button

Flips the image right to left. This does not recompute the whole image.

Flip Vertical Button

Flips the image top to bottom. This does not recompute the whole image.

Zoom In Button

Reduces the radius by a factor of 2.5, which magnifies the image by a factor of 6.25. This forces a recomputation.

Zoom Out Button

Increases the radius by a factor of 2.5, which reduces the image by a factor of 6.25. This forces a recomputation.

Reset Button

Clicking this button resets the view to the default view for the current generator. This default view is intended to work well for the majority of images produced, but it isn't guaranteed to be good in every case.

Geometries

Geometries are a new feature in Synthimax 1.3. A geometry changes the shape of the plane in which the image is drawn and radically affects the result.

A geometry always has a center and may have a radius. Both of these parameters may be defined as fixed or relative. A geometry with a relative center is like a pair of glasses on your nose: wherever you look, the glasses warp the image you're looking at. A fixed center is like putting the glasses on a surface: you can change the area you're looking at and the distortion from the glasses won't move with your gaze.

If you change the mode of the center or radius parameter, the corresponding values will alter. The relative center coordinates are added to the center of the view and a value of 1.0 corresponds to one-half of the view height or width, so that a center set at (1.0, 1.0) is at the top-right corner of the view rectangle. The relative value of the radius is the radius of the view as set in the text box in the upper part of the pane.

Geometries do not work equally well with all generators. Dynamical systems and phase plane generators work by recording the path of points across the plane; this means that they must use the *inverse* of the geometry to plot the final point. But not all geometries have an inverse! When you use a geometry that the generator can't handle, you will get a warning below the geometry box. However, rather than not using the geometry at all, Synthimax applies the direct geometry rather than its inverse. The results are somewhat hard to predict, but anything that makes weird images can't be all bad...

N-fold symmetry

This geometry basically acts like a kaleidoscope with 3 or more mirrors, equally-spaced. If you uncheck the Reflections checkbox, then a slice of the image will be repeated n times, rotated appropriately. The offset

angle determines where the slice to repeat is located.

Inverse

The inverse geometry takes the complex inverse of every point in the plane. The center of the image is projected outwards in all directions, and the periphery is projected inwards.

Complex powers

Considers every point of the plane as a complex number and raises it to an integer power. This distorts the plane symmetrically.

Bulge

Pushes the outer edges of a circle towards infinity. The bulge can have a flat center or swell to a point. The steepness parameter affects how sharp the bulge appears.

Split disk

A weird geometry that bends straight lines into curves and splits the view along a horizontal line. Note that for arcane mathematical reasons, this geometry affects Dynamical Systems oddly: zooming has no effect on the picture as long as the radius is relative.

Swirl

Bends the plane into a spiral. The spin value sets how strong the rotation is, and the intensity determines how fast the effect tapers off as you move away from the center.

The Drawing Pane

Filter Popup and Filter Parameter Text Box

This selects which filter will be applied to the raw image data. It is important to remember that filters act upon the elements in the raw image, and not the pixels in the final image.

Some filters use an additional parameter, which is displayed in the text box to the right. This box is disabled when the filter doesn't use the parameter.

Version 1.3 adds a second filter, which is applied to the result of the first. Combining filters of two different flavours, like Mirror and Detail, can result in even more interesting images.

- **None:** no filtering is applied.
- **Blur - Light:** Each element becomes a weighted sum of the element and its neighbors in a 3x3 square. The effect is generally to make the image smoother. However, if adjacent elements are very different and the gradient repeat factor is high, a “halo” effect can be produced around individual pixels.
- **Blur - Heavy** is like Blur - Light but it gives less weight to the central element and thus intensifies the blurring effect.
- **Laplacian:** Pixels where element values change rapidly are picked out by this filter. It uses the parameter, which determines the approximate value of the rest of the image. Try setting it to one-half the norm (see below).
- **Detail:** The image color scheme remains similar, but pixels where values change rapidly are emphasized. Can yield spectacular results with dynamical systems.

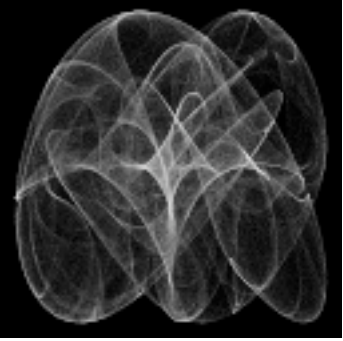

- **Enhance:** Similar to Detail, but the emphasis is more subtle. Works best with images with large regions where values change gradually. For this reason, it gives mediocre results with dynamical systems but is recommended for smooth textures (Mandala, Iterated Sines, Polynomial Quotient) and fractals that use the dwell orbit method.
- **Valley:** The value at a point is divided by the number of neighbors with a higher value. Points that are at a local minimum (a valley, as it were), will be emphasized. Sometimes lines corresponding to the bottom of the valleys will snake across the image.
- **Edger:** Creates a contrasting edge to the right. Can yield spectacular results with dynamical systems.
- **Derivative:** Computes an approximation of the derivative of the image generator function. Emphasizes edges and decreases contrast in the final image.
- **Relief:** Subtracts the element at the upper left from the current element and adds the parameter value to the result. With a good value for the parameter, this yields images that look like smooth surfaces with carved or raised features. Try setting the parameter to one-half the norm (see below) as a first approximation.
- **Minimum:** Each element is replaced by the minimum of all elements over a 3x3 square. This can give the image a blocky appearance; however, for certain configurations, using Minimum smooths out the raw image. This can lead to a cleaner-looking final image.
- **Averaging:** Each element is replaced by its difference from the average of all elements in the image. This looks like the gradient has been altered.
- **Smooth:** This filter uses the parameter. When its value is 0, there is no effect on the image. As the parameter increases towards 1.0, the value of every element is brought towards the minimum of the image.

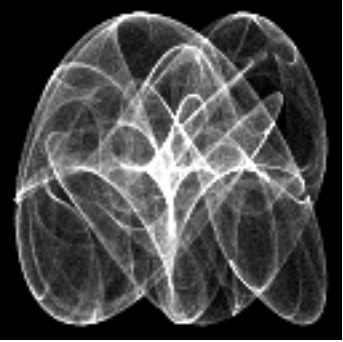
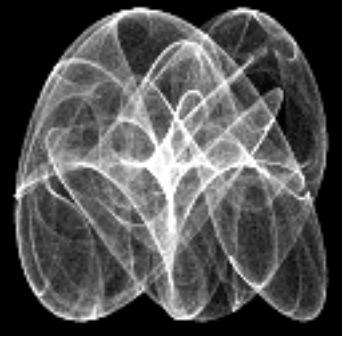
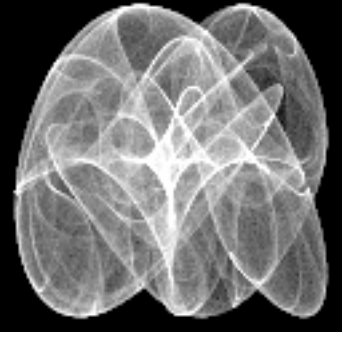
- **Mirror:** Subtracts, from each element, its symmetrical opposite around the center point. This creates a much “busier” image.
- **Kaleidoscope-4:** The top left quarter of the image is repeated three times, flipped as appropriate, to form a symmetrical image.
- **Kaleidoscope-8:** The top right “slice” of the top left quarter of the image is repeated seven times to form a symmetrical image. Note that when the image is not square, pixels in some of the other slices will be “stretched”. Using the “N-fold symmetry” geometry will produce a much nicer effect, so it is the preferred option.
- **Sines:** Pixel values are multiplied by the difference of the sines of the products of their diagonal neighbors. This adds noise at various scales to the image; it is best employed with smooth textures (Mandala, Iterated Sines, Polynomial Quotient). Reducing the multiplicand will reduce the amount of noise: a value of 100 produces much “cleaner” results than 1000.
- **Circular offset:** This multiplies every element by the square root of its distance to the center of the image. It can create a bull’s eye effect in an image.
- **Square root remainder:** A weird filter that multiplies the element by its right-hand neighbor, takes the square root of the result, and multiplies the fractional part by the maximum of all elements. Results tend to be a bit unpredictable.
- **Cutoff:** This filter uses the parameter, which must be between 0 and 1. If the value at a point relative to the minimum and maximum of the image is lower than the parameter, it is forced to the minimum. As you increase the parameter from 0 to 1, larger and larger areas of the image will appear cut out.

Normalizing Popup and Norm Text Box

This selects how the result of the filtering is normalized. The normalization parameter is called the *norm*, and it is set in the text box to the right of the popup. Generally speaking, the norm represents the maximum value allowed for the filter result. Although the intent of normalization is to bring the filtered value to the range [0,1], you can always choose a norm that produces values outside this range.

The first 5 normalizing types are more straightforward than the others, especially if you are using a simple gradient. This table shows their results on an image using a black-to-white gradient.

Arctangent	
Exponential	

Linear	
Logarithm	
Square Root	

Notice that these normalizings do not merely lighten or darken the image, they also change the contrast between its various parts.

- **Arctangent:** Returns $1 - \frac{\left| \tan^{-1} \left(\frac{norm}{filt} \right) \right|}{2\pi}$.

- **Exponential:** Returns $1 - e^{-\frac{filt}{norm}}$.
- **Linear:** Divides the filter result by the norm. This is the most straightforward normalization.
- **Logarithm:** Returns $\log_2\left(1 + \frac{filt}{norm}\right)$.
- **Square root:** Returns $\sqrt{\frac{filt}{norm}}$.
- **Inverse:** Returns $\frac{1}{1 + \frac{filt}{norm}}$. This more or less reverses the gradient.
- **Sine:** Returns $\frac{1}{2}\left(\cos\left(\frac{2\pi \cdot filt}{norm}\right) + 1\right)$. If the filter result is 0, this is 0, and if it is equal to the norm, it's 1.
- **Modulo:** Truncates the filter result, takes its remainder modulo the norm and divides that by the norm minus 1. In other words, it returns $\frac{[filt] \bmod [norm]}{[norm] - 1}$.

For norm = 2, this means every element becomes 0 or 1. For norm = 3, possible values are 0.0, 0.5 and 1.0. For norm = 4, we get 0.0, 0.33, 0.67, and 1.0, and so forth. High values for the norm here give the same result as linear normalizing. Note that many gradients start and end with the same color. Using a mod 2 normalizer will then produce a featureless image... Set the factor to 0.5 or change gradients to get a worthwhile result.

- **Fuzz:** Multiplies the filter result by the norm, takes the square root of

the product, and retains the fractional part. For this normalization, the norm should be small—setting it to 1 is often best.

Adjust Button

Sets the norm to the maximum element in the raw image and forces a recompute.

Auto Checkbox

When this is checked, every time a new image is computed, the norm will be automatically adjusted as per the Adjust button mentioned before. This is not necessary in general. It can be useful if you are testing various orbit methods for a generator, since those methods tend to produce very different values from one another. Also, some Dynamical System generators (particularly Hopalong) can produce images with widely varying values when parameters are changed, so again auto-adjusting the norm may be useful in that case.

The auto-adjust checkbox is turned off when you go to the previous or next configuration, and when you select a configuration from the browser.

Floor, Ceiling

These text boxes allow you to specify the minimum (floor) and maximum (ceiling) value to which the normalized value is clamped. You cannot set a floor less than zero, or a ceiling more than one.

Gradients

The color gradient used has a dominant influence over how the raw image produced by an image generator becomes an actual picture. For each pixel of the final image, a numeric value between 0 and 1 is used to obtain a color from the gradient.

In its most basic form, a gradient is defined by two colors: one positioned at 0 and the other positioned at 1. Any number between 0 and 1 will correspond to a color that is intermediate between the gradient's two extremes. This principle continues to apply if we add more colors to the gradient. Each color has a precise position and any value that falls in the range [0,1] falls between two colors.

Factor and offset

Consider these two gradients:



These are obviously different, but they're also very similar, because the second gradient is just the first one repeated twice. So by giving the first gradient a factor of 2.0 we obtain the second gradient.

Likewise, these two gradients are similar because they contain the same sequence of colors; one is just offset from the other. We obtain the second from the first by adding an offset of 0.33.



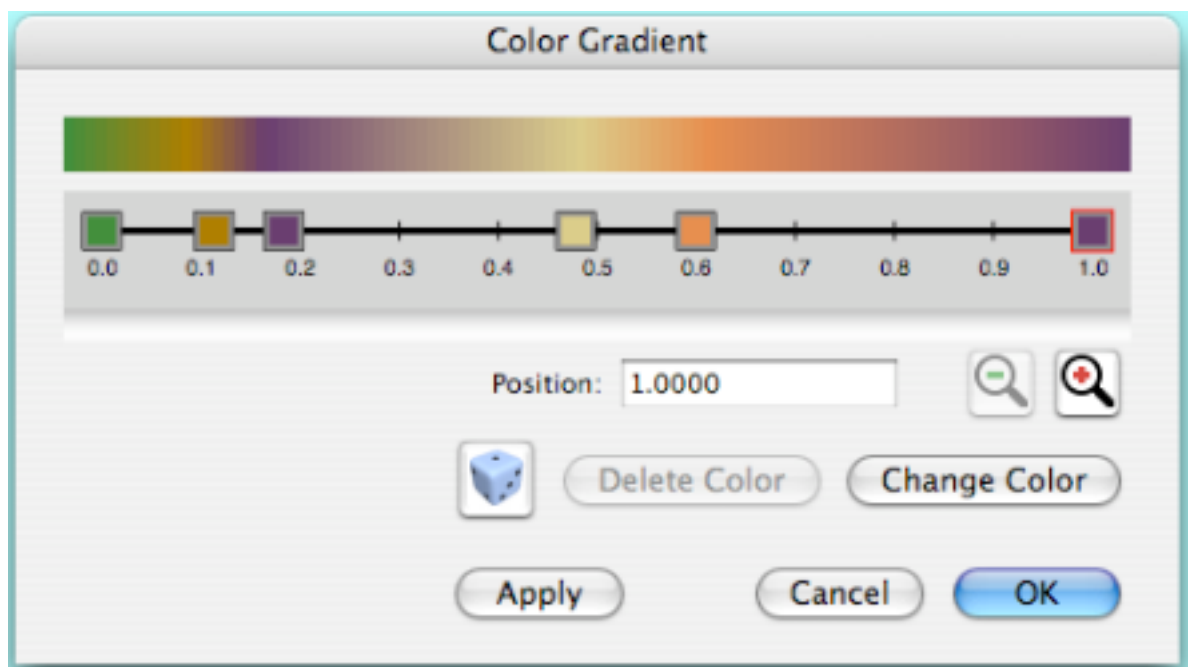
When you specify a gradient in the Drawing pane, you can also specify

a factor and an offset for the gradient. This is much more flexible and concise than having to define a separate gradient for every desired factor and offset.

The offset of a gradient should be between 0.0 and 1.0. An offset of 1.0 is the same as an offset of 0. For the factor, you can use any value you want, except 0. You can even use negative values. Unlike what you might expect, this is *not* the same as using the same gradient with the colors reversed. This is because the color for 0 remains the same regardless of the factor. For some images and gradients this can make a huge difference.

Edit Gradient

Click on this button to edit the current gradient. This brings up the dialog window below:



The Line control at the top shows a line graduated from 0.0 to 1.0 in increments of 0.1. On this line, markers can be added: each marker corresponds to a color.

Click on a marker to select it: its outline turns to a thick red line. Once a marker is selected, its position appears in the Position text box. The marker may be moved by entering a new value in the Position text box or by dragging the marker. A selected marker can also be deleted, except for the markers at 0.0 and 1.0, which must always exist. Click the Change color button to bring up a color-selection dialog for the selected marker.

You may add a marker by option-clicking on the Line control. The cursor changes shape to indicate this. When option-clicking, a marker will be added where you clicked and a color will be added to the gradient at that point.

You can click on the magnifying glass buttons to zoom in or out of the line. This can be useful when you have several markers close to one another.

There is also a Randomize gradient button in the window. Clicking it will generate a random gradient (see below).

At any time, you can click the Apply button, which will apply the current gradient to the image. This is a preview only: if you click Cancel, you will return to the previous gradient.

Store Gradient

Clicking this button will store the current image gradient into the current gradients file. If you haven't opened a gradients file yet, Synthimax will create one automatically. See "The Gradients Window" below.

Gradients

Clicking this button brings up the gradients window, which is described later.

Randomize Gradient

Click on this button to have Synthimax set a random gradient as the current gradient. Random gradients have between 2 and 6 colors, placed at random spots along the $[0,1]$ interval.

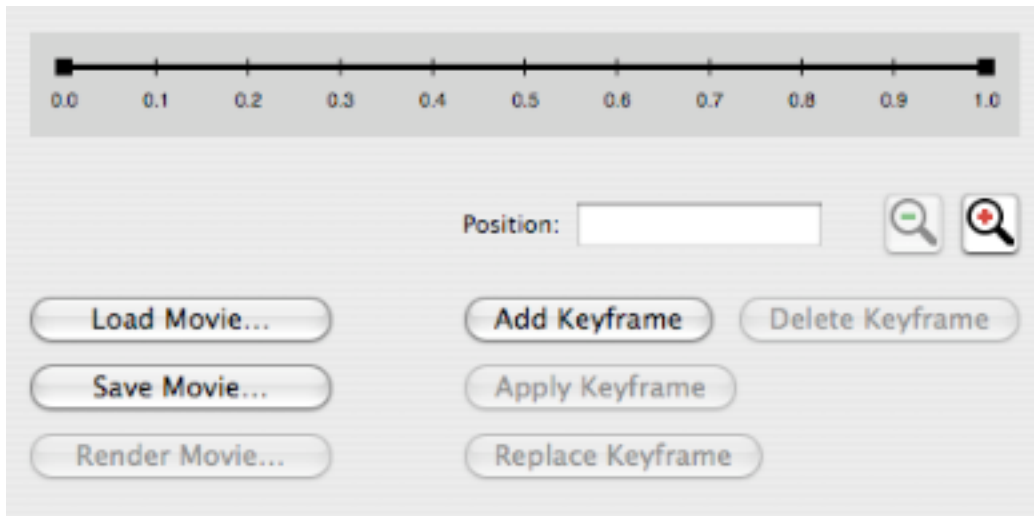
Factor

Makes the gradient repeat n times in the $[0,1]$ interval.

Offset

Offsets the gradient by a fixed value.

The Timeline Pane



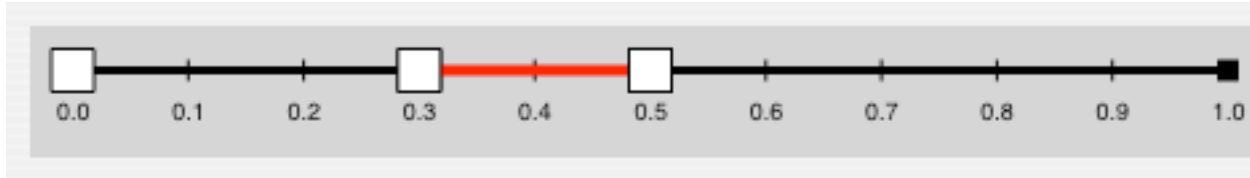
This pane shows a line control at the top. The line control displays a line graduated from 0.0 to 1.0 in increments of 0.1. On this line, markers can be added, corresponding to keyframes.

Click on a marker to select it: its outline turns to a thick red line. Once a marker is selected, its position appears in the Position text box. The marker may be moved by entering a new value in the Position text box or by dragging the marker. A selected marker can also be deleted.

You may add a marker by option-clicking on the Line control. The cursor changes shape to indicate this. When option-clicking, a marker will be added where you clicked, and a keyframe containing the current configuration will be added to the timeline.

You can click on the magnifying glass buttons to zoom in or out of the line. This can be useful when you have several markers close to one another.

Note that Synthimax cannot morph between two configurations with different image generators (see Rendering Movies). If two consecutive keyframes have different generators, the timeline between them will be drawn in red, like this:



This is to remind you that no morphing whatsoever can take place between these two keyframes.

Adding, Moving and Deleting Keyframes

When adding a keyframe, you always add the current configuration to the timeline. You can add a keyframe by clicking the Add Keyframe button. You will be asked to enter the position for this keyframe (which must be between 0.0 and 1.0). You can also add a keyframe by option-clicking in the line control. This adds a keyframe at the cursor position.

When a keyframe marker is selected, you can delete it by clicking the Delete Keyframe button.

Apply Keyframe

It isn't possible to indicate on a keyframe which image it corresponds to. When you select a keyframe on the timeline, you can click the Apply Keyframe button to apply the corresponding image to the controls. Clicking the Draw button will then draw the image.

Replace Keyframe

If you want to tweak a keyframe, start by selecting then applying it. Adjust the parameters as you want, then click the Replace Keyframe button. This will change the keyframe on the timeline to match the current control values.

Saving and Loading Movies as Keyframe Sets

You can save a file containing all the keyframes on the current timeline, by clicking the Save Movie button. This file, which has the extension .msnaps, contains all the configurations and their positions on the timeline. If you click the Load Movie button, you can open a keyframe set, which will replace the keyframes on the current timeline.

Saving Pictures

When you've found an image you like, you may want to save it. Choose the menu option **File>Save Picture As...** Synthimax will save the image as a .png file, which can be opened by any modern graphical application.

Rendering Movies

You can make movies with Synthimax, using *keyframe animation*. A keyframe, for our purposes, is a given configuration. You put keyframes on a timeline and the application will create an animation by smoothly morphing between one keyframe and the next. Morphing is applied to all parameters at the same time. Some parameters cannot be morphed: for instance, the orbit method used for a fractal generator. In this case, the parameter will not be morphed and will change suddenly when the second keyframe is reached. The same effect applies if you try to morph between two configurations that have different generators: there will be no morphing at all between the configurations.

Rendering a Movie

You render a movie by clicking the Render Movie button on the Timeline window. The application will first ask you where to save the movie file. It will then put up a dialog where you specify the following parameters:

- Image size (in pixels).
- Number of frames: how many frames (images) the movie will have in total.
- Frames per second: how fast the movie runs. The higher this number, the smoother the movie will play, and the shorter it will last.
- Codec: this determines how the images are encoded into the movie. Currently, there are two possibilities: Animation, which keeps every image as it was generated and produces the best quality; and H.264, which compresses the images using the latest-and-greatest Quicktime technology, saving huge amounts of space but degrading quality.

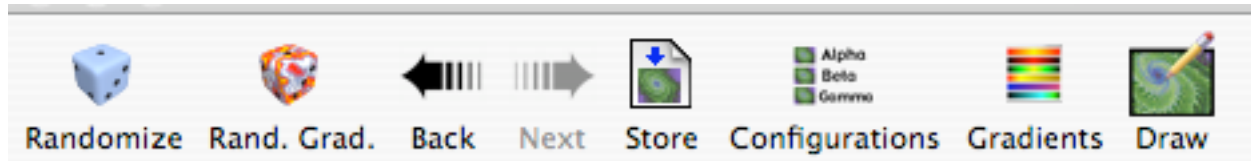
Once you have specified the parameters, Synthimax will start rendering the movie. During this time, it puts up another dialog box that shows the most recently-rendered frame (scaled to fit) and the rendering progress. The application tries to estimate how much time the rendering will take and displays this in the dialog box. This figure is not fully reliable, since some configurations may be much slower to render than others.

If you click the “Stop rendering” button, the rendering will end immediately. The movie will still be saved to disk but will have been cut short.

Space Considerations

Movies can take up *a lot* of space. A typical 120-frame movie (at 256x256) rendered using the Animation codec uses up nearly 26 megabytes! Using compression on images reduces the amount of disk space consumed: the same movie rendered using H.264 only uses up 3.4 megabytes. However, using compression means there will generally be a noticeable degradation in quality. This effect is far more noticeable for abstract animations than for actual live footage, unfortunately.

Toolbar Controls



Randomize Configuration Button

Selects a random configuration of parameters for the current image generator. The view rectangle will be reset to the default view for this generator. The drawing controls will remain unchanged.

Parameters are randomized in an intelligent fashion: insofar as possible, they are verified to ensure that the resulting image will be interesting, i.e. that it will not be limited to a thin scattering of points. Starting with version 1.2, this is true for all Dynamical Systems generators: the configuration is run for a relatively small number of iterations, to verify that the points are widely distributed and don't just cycle through a few values. If the configuration is uninteresting, it is rejected and a new random configuration is chosen.

This means that in practice, when you click the Randomize button, you can be almost certain you won't get an image limited to five or six points on a blank background. Note that the Gumowski system is not always well-behaved, and certain configurations may eventually settle into uninteresting cycles even though they appeared interesting at the start. If you get a randomized image that appears limited to a handful of points, try reducing the number of iterations.

Newly-available in version 1.3 is parameter locking. Every generator parameter has a small lock icon next to it. If you click on the icon, the lock closes, which means that this parameter will not be randomized. This can be very useful if, for example, you don't want to change the view plane or the orbit method in Complex Feedback, just the function being iterated.

Randomize Gradient Button

Creates a random color gradient and applies it to the current image. This is the same as clicking the randomize gradient button in the Drawing tab.

Previous/Next Configuration Buttons

Clicking on the left one of these buttons returns you to the previous configuration; once you have gone back, the right button activates and allows you to move forward again to the last configuration. Synthimax remembers the last 20 configurations you have chosen.

If you go back to a previous configuration and modify it, Synthimax still remembers the later configurations, so the “next configuration” button remains active.

New in version 1.3: Synthimax stores the complete image in memory for the last 5 images that were drawn, so that when returning to an image, you don't need to recalculate it. Note that if you've just changed image size, returning to the earlier image will change the size of the image back.

Store Configuration Button

Clicking this button stores the current configuration in a configurations file. You will be asked to give the configuration a name. You can ask to store a configuration even when you haven't opened a configurations file; in that case, Synthimax will automatically create a new file first.

Configurations Button / Gradients Button

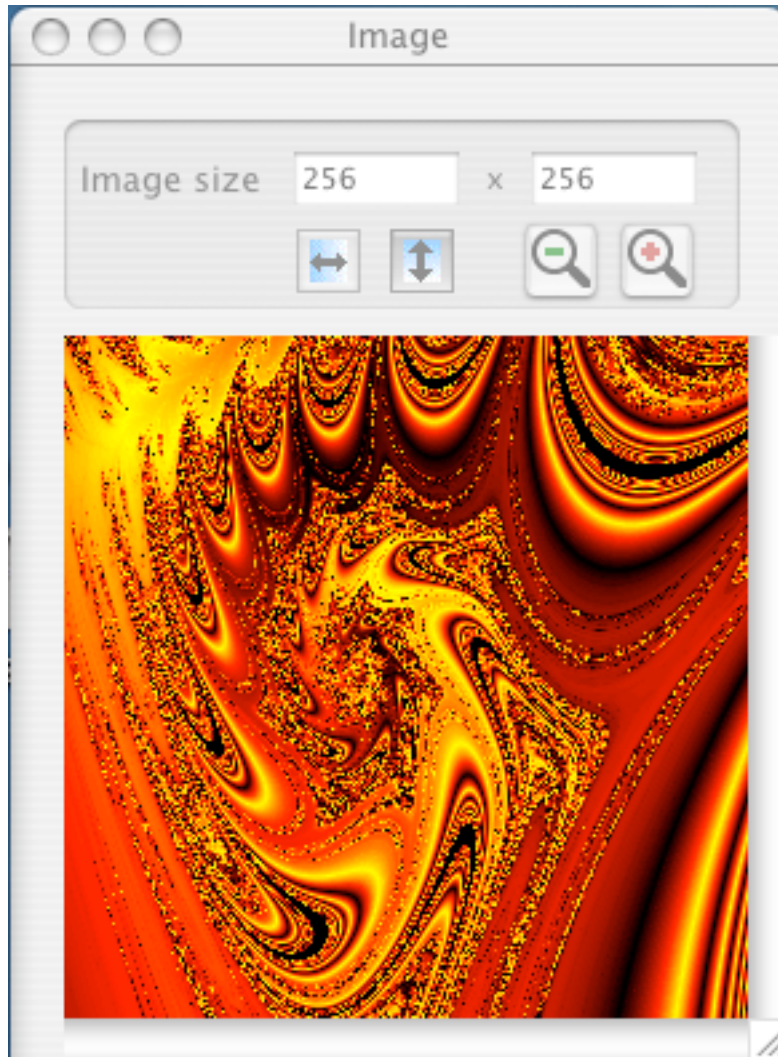
When you click one of these buttons, the Configurations or Gradients window appears. See below for details.

Draw Button

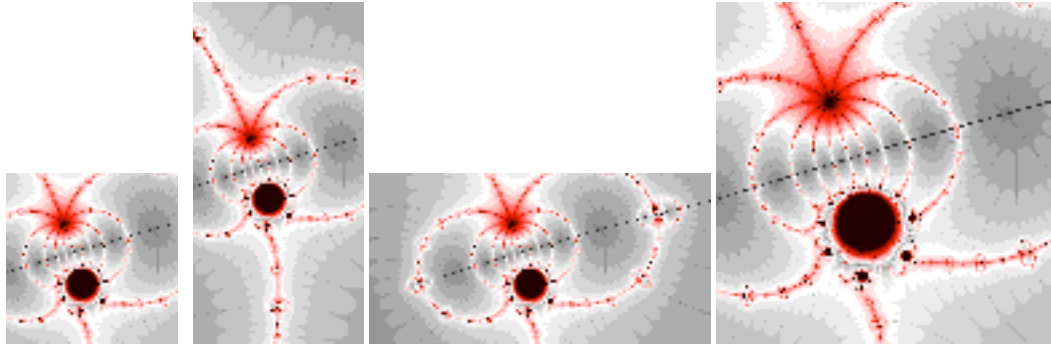
Draws the image according to the current parameter configuration. If

the button is inactive, it's because the current image matches the configuration, so there's no redrawing necessary.

The Image Window



The image window displays the current image. At the top of the window are two text boxes specifying the dimensions of the image, width times height, in pixels. The proportions of the image, however, remain constant: when height and width differ, the image does not get distorted; instead, it shows a larger view. This is best explained visually. The images below have dimensions of 64x64, 64x128, 128x64 and 128x128.



You can resize the image window; if the image is larger than the window itself, scroll bars will activate and allow you to move around the image.

To accept a size you have entered, hit the Return or Enter key, and the image will resize to the dimensions you specified. If you entered a size that isn't part of the standard menu, Synthimax will remember it (until you quit the program) and add it to the end of the Image Size menu.

Flip Horizontal Button

Flips the image right to left. This does not recompute the whole image.

Flip Vertical Button

Flips the image top to bottom. This does not recompute the whole image.

Zoom In Button

Reduces the radius by a factor of 2.5, which magnifies the image by a factor of 6.25. This forces a recomputation.

Zoom Out Button

Increases the radius by a factor of 2.5, which reduces the image by a factor of 6.25. This forces a recomputation.

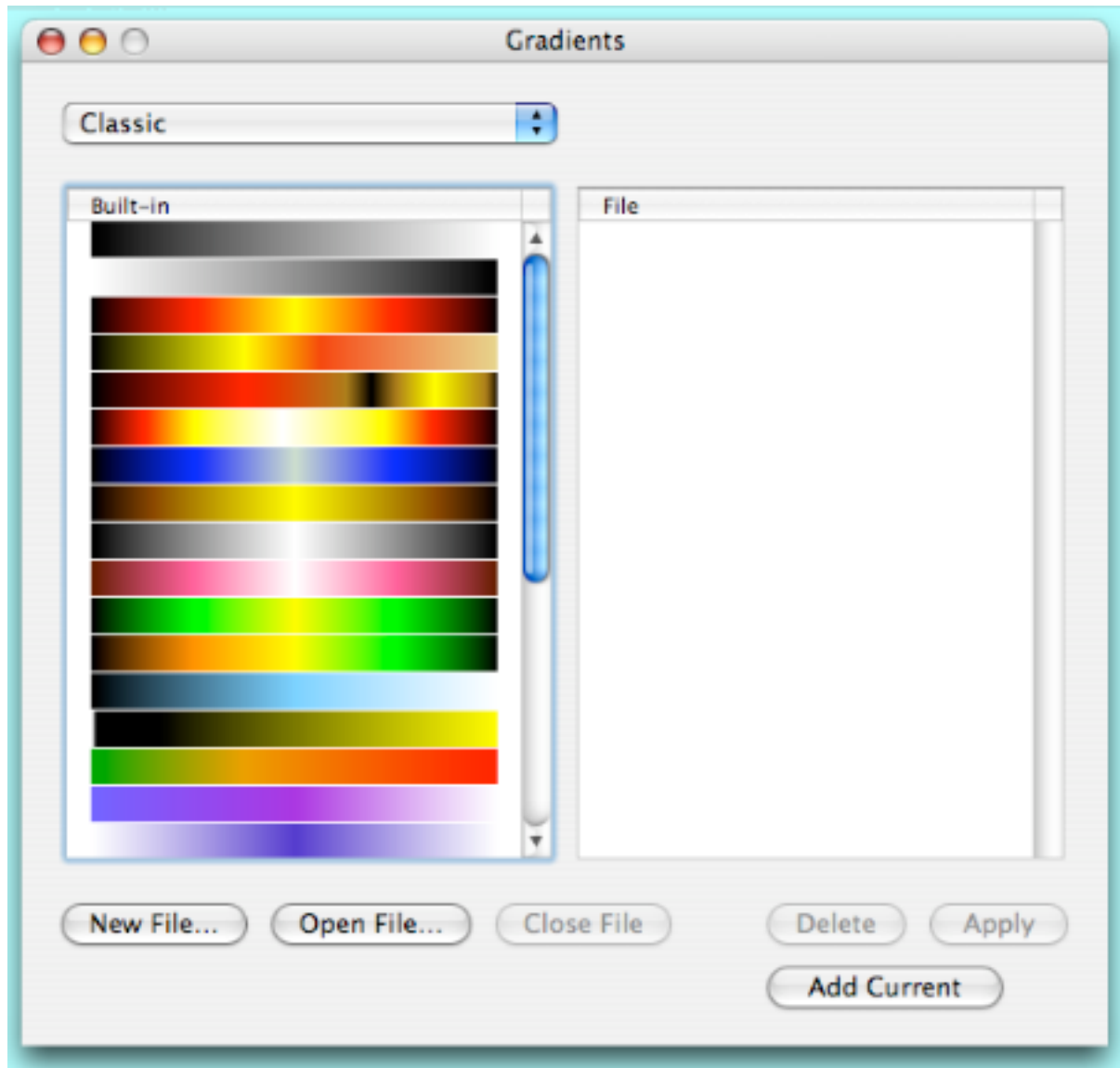
Clicking in the image window to zoom and move the image

If you click and drag in the image window, you will draw a rectangle, centered on the initial clicked point, over the image. When you release the mouse button, the area described by this rectangle will determine the new boundaries of the image. The rectangle has a fixed center because usually what you want to do is to center the image around a feature of interest. If you prefer to draw a more standard rectangle that has a fixed corner, hold down the shift key when clicking.

If you don't like this arrangement, you can change it by choosing the menu option **Synthimax>Preferences...** to select the zoom behavior.

If you hold down the command key over the image window, the cursor becomes a hand. Click and drag to move the image around. It will be recomputed when you release the mouse button.

The Gradients Window



This window displays color gradients. On the left is a browser containing built-in gradients. On the right the other browser shows the gradients contained in the currently-open gradients file. Click on a gradient to select it.

Just above the left-hand browser is a menu that allows you to select a category of gradients. "Classic" contains the same gradients as in

version 1.2 of Synthimax. Other categories group gradients according to their dominant color scheme, or to their structure (symmetric, tricolored, washes).

New File...

Click this button to create a new gradients file.

Open File...

Click this button to open an existing gradients file.

Close File...

Click this button to close the open gradients file.

Apply

When you have selected a gradient, click on this button to apply it to the current image. Please note that the gradient is *copied* into the image: editing the gradient of the current image *never* edits a gradient inside a file.

Delete

Deletes the selected gradient from the current file. This button does not work with built-in gradients, which are always available and can never be deleted.

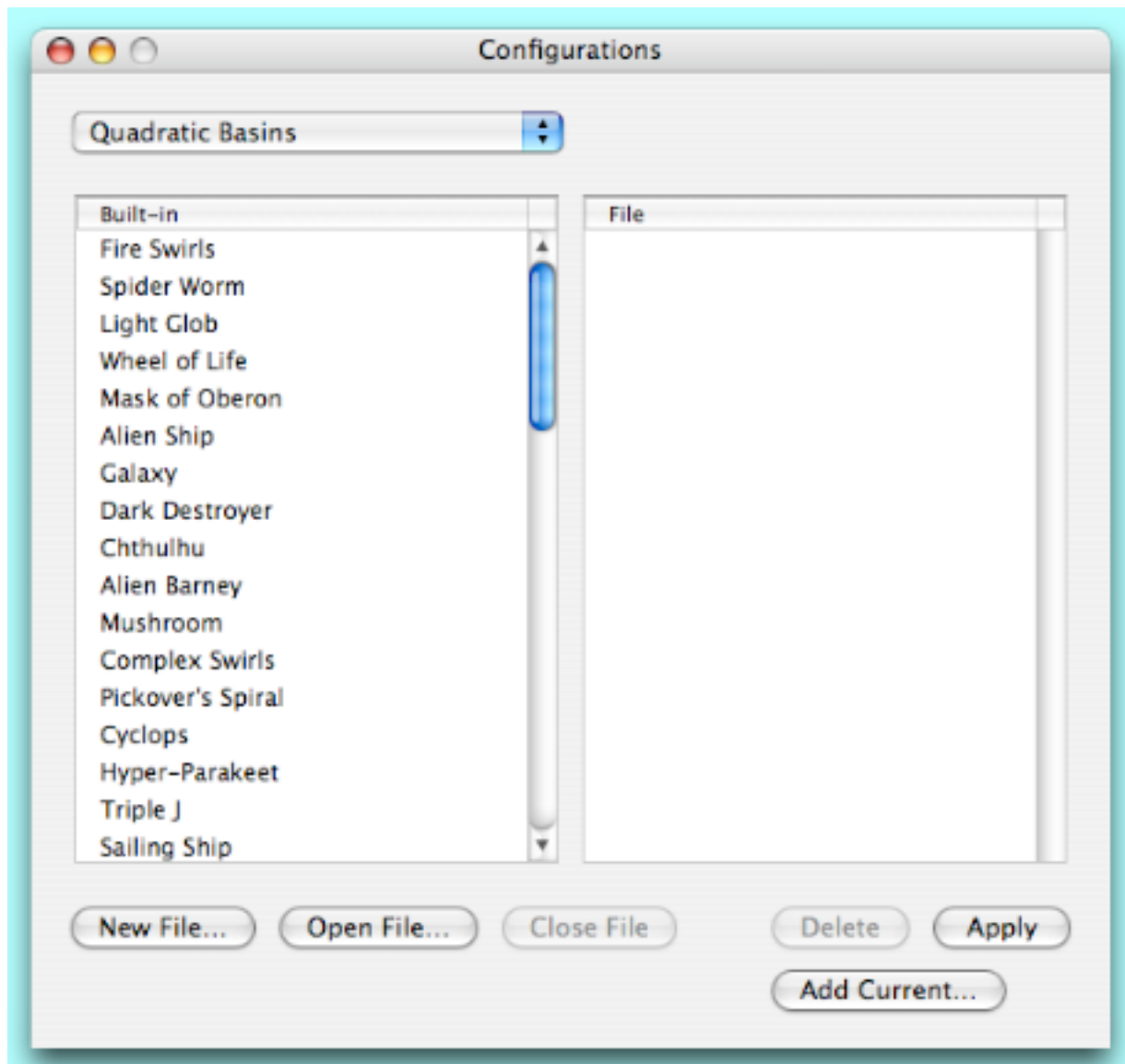
Add Current...

This button adds the current image gradient to the gradients file. It does exactly the same thing as the Store Gradient button in the main window. You can ask to store a gradient even when you haven't opened a gradients file; in that case, Synthimax will automatically create a new file first.

The Configurations Window and Configurations Files

In order to recall the parameters used to produce an image, you can capture a given configuration of parameters in a *configurations file*. A single configurations file contains many different configurations. Note that a configuration contains all the parameters that define an image, except for its size.

To manage configurations files, bring up the configurations window.



This window displays image configurations. On the left is a browser containing built-in configurations. On the right the other browser shows the configurations contained in the currently-open configurations file. Click on a configuration to select it.

Just above the left-hand browser is a menu that allows you to select a category of configurations. These are grouped by generator.

New File...

Click this button to create a new configurations file.

Open File...

Click this button to open an existing configurations file.

Close File...

Click this button to close the open configurations file.

Apply

When you have selected a configuration, click on this button to apply it to the current image. Please note that the configuration is *copied* into the image: editing the current image *does not* edit a configuration inside a file.

Delete

Deletes the selected configuration from the current file. This button does not work with built-in configurations, which are always available and can never be deleted.

Add Current...

This button adds the current active configuration to the configurations file. It does exactly the same thing as the Store Configuration button in the main window's toolbar. You will be asked to give the configuration a name. You can ask to store a configuration even when you haven't

opened a file; in that case, Synthimax will automatically create a new file first.

Credits

This software was developed on an Apple Macintosh using Xcode 2.4.

Developer: Yves Meynard

Chief Beta Tester and Gadfly: Serge Meynard

Quadratic Basins concepts taken from Clifford Pickover's "Automatic Generation of General Quadratic Map Basins", in *Computers and Graphics*, Mar/Apr 1995.

Some concepts for Halley Maps, Complex Feedback, Dynamical Systems and Phase Plane derived from Clifford Pickover's *Computers, Pattern, Chaos, and Beauty*; St Martin's Press, 1990.

Sine minus Cosine dynamical system taken from Clifford Pickover's "Million-Point Sculptures" in *Computer Graphics Forum* vol 10, p.333 (1991).

Hopalong dynamical system discovered by Barry Martin and initially presented in A.K. Dewdney's column "Computer Recreations" in *Scientific American*, September 1986.

Gumowski dynamical system derived from the work of I. Gumowski and C. Mira, as presented in *Creative Computing* Sept-Oct 1978 by John Lansdown, and tweaked along the lines of several adaptations found on the Web.

Mandala texture from *What To Do After You Hit Return*; People's Computer Company, 1975.

Perlin Noise was invented by Ken Perlin. See <http://mrl.nyu.edu/~perlin/> and <http://www.noisemachine.com/talk1/>.

Soundtrack by: M83, Decoded Feedback, Front 242, and David Kristian.