The Mandelbrot-Julia Set Plotter

Dan Gries
www.dangries.com
info@dangries.com

An application built in Adobe Flash, available at
http://www.dangries.com/Flash/Fractals4D/MandelbrotJulia.html

1 About the application

The **Mandelbrot set** is defined as follows: a complex number $c \in \mathbb{C}$ is in the Mandelbrot set if the infinite sequence $\{z_n\}$ of iterates given by $z_0 = 0$ and $z_{n+1} = z_n^2 + c$ remains bounded (instead of going to infinity in magnitude). The **Julia set** J_c is defined by the same iterating function, but now by fixing a value c and allowing the initial value z_0 to vary. The set J_c is the set of values z_0 which result in bounded sequences.

To generalize, we allow both z_0 and c to vary, and consider the pairs $(z_0, c) \in \mathbb{C}^2$. A point (z_0, c) is contained in the Mandelbrot-Julia set if the iterates of $z_{n+1} = z_n^2 + c$ remain bounded. This is a four-dimensional set so it is difficult to visualize, but we can look at lower-dimensional cross-sections of it. The Mandelbrot-Julia Plot application plots two-dimensional square planar slices of this set.

This is not a great way to visualize the entire Mandelbrot-Julia set; it is somewhat like trying to understand a three dimensional solid by considering one dimensional threads running through it. But the two-dimensional intersections of the Mandelbrot-Julia set are somewhat beautiful and fascinating in their own right.

The application opens up with a default view of the Mandelbrot set. This is produced by parameters which choose a viewing square where the value z_0 is always equal to zero. You can leave the bounds unchanged and explore the Mandelbrot set on its own.

2 The Bounds Tab

The controls in the bounds tab are where you select two-dimensional squares in the four-dimensional space $\mathbb{C}^2$. This is done by selecting three points in $\mathbb{C}^2$. Three points normally determine a plane; if the three points are colinear the application will ask you to select three different points. The selector on the left is for the z_0-coordinates of the three points, the selector on the right is for the c-coordinates. A perfect square which contains the three selected points is where the plot takes place.

The little **M** and **J** buttons at the bottom of the bounds tab set the bounds to plot a view of either the Mandelbrot set or a Julia set (with $c = 0$), respectively. The Julia set for $c = 0$ is a simple filled in circle; hold down the CTRL button and drag the three points in the c selector together to a different location to see more interesting Julia sets.

3 The Color Splines Tab

In reality, we cannot compute this infinite sequence entirely to determine whether it is bounded. Instead, we compute the sequence up to some maximum number of iterations. If a value z_n reaches a magnitude of 2 or more, then we stop the iterations because it can be shown that the sequence will go off to infinity. (We say that the *bailout radius* is 2.) If a magnitude of 2 is never reached after the maximum number of iterations is computed, we assume the sequence is bounded and that the initial point (z_0, c) is in the Mandelbrot-Julia set.

In the plot shown in the application, a point is colored black if it is contained in the Mandelbrot-Julia set. If not, then the point is given a specific color according to the number of iterates that were computed before the bailout radius of 2 was reached. This method produces a discrete (stepped) coloring of the exterior, but as an option the application can employ a more sophisticated method to continuously (smoothly) color the exterior.

The colors are chosen by the gradient shown at the bottom of the Color Splines tab. We would need a very large number of colors to represent all the different possible bailout iteration levels, so instead we cycle through the colors and reuse them. Clicking on the button to the right of the gradient, you can choose whether the colors move back and forth through the gradient or move in only one direction.

The gradient can be edited by clicking and dragging on the red, green, and blue curves (called the *color splines*) shown above the gradient. These curves represent the amount of red, green, and blue light that combine to form the colors at different points along the gradient. Since every color that can be shown on a computer monitor is a combination of red, green, and blue components, almost any desired gradient can be created through this interface. This task is interesting in its own right, but at times can produce some rather garish color combinations. To assist you, some predefined gradients can be selected through an interface that opens up with the *select gradient* button.

The buttons beside the color splines allow you to copy (c) and paste (p) one spline to another. The arrow buttons allow you to scale up or down the values defined by the splines. The *smooth* button will smooth out the curve. If the *auto* button is highlighted, the curve will be smoothed out every time you draw a new portion of it.

4 Iteration and Coloring Parameters

The *color period* parameter determines how fast the colors change - larger numbers make the transition go slower.

The *color phase* parameter shifts the colors without affecting the rate of change of the coloring. This parameter should be set to a value between 0 and 1.

The *max iter* parameter sets the maximum number of iterations to be computed before the sequence is assumed to be bounded. Higher values will be required on certain zoom windows, but expect to wait longer for the plot to render.

The selector which initially reads "log" selects four different coloring methods. The standard coloring methods are the "discrete" method and the "linear" (continuous) method. Since the change in colors becomes more rapid as the boundary of the fractal is approached, the picture can get somewhat messy. Logarithmic coloring slows down the rate of change of colors as the boundary is approached. This slowing of the rate of change can be made even more pronounced with the "log log" setting.

5 Zoom and Window Controls

Clicking and dragging on the fractal display will create a rectangle selection. You can zoom in on this selection with the *zoom selection* button (the new display bounds will be a square containing the rectangular selection). The *zoom out* button zooms out by a factor of three lengthwise. The *redraw* button can be used to redraw the current view if you change the coloring or iteration settings. The *back button* will go back to the previous window settings without changing the current coloring or iteration parameters. The *start over* button will go back to the window defined by the three points in the Bounds tab.

The *transfer bounds* button transfers the current window bounds (the bottom left, top left, and bottom right points) to the point selectors in the Bounds tab.

6 Oversampling

Fractal pictures contain a lot of rapidly changing detail, and thus can become very grainy looking. To improve picture quality, we can *oversample*. If the oversampling checkbox is selected, the application will compute a plot which is twice as big in both the height and width (thus four times the data will be computed), and the picture will be smoothly resized down by a factor of two lengthwise. The result is a much nicer and clearer picture of the fractal. However, since this method requires four times as long to compute, you may wish to use it sparingly.

7 The Gallery

Clicking on the *gallery* button will bring up some saved fractals. At the moment, there is no way to save your work (except to take a screen shot), but that capability is in the works.

8 The Export Buttons

Two buttons at the bottom will allow you to save the current fractal view as an image in either the jpg or png format. Note that the png format is lossless, and thus will provide better results (at the expense of a larger file size). If you use these pictures for any on-line purpose, I would be grateful (but it is not necessary) if you would mention that the fractal was created with this applet, and perhaps include a link.

9 Buttons to Ignore

The button labeled "Fractal XML" and the button on the Color Splines tab labeled "show XML" should be ignored. For now, this XML data is only usable by me.

10 Comments Welcome

This applet is still a work in progress, and your comments are welcome and appreciated. Please direct any correspondence to info@dangries.com.