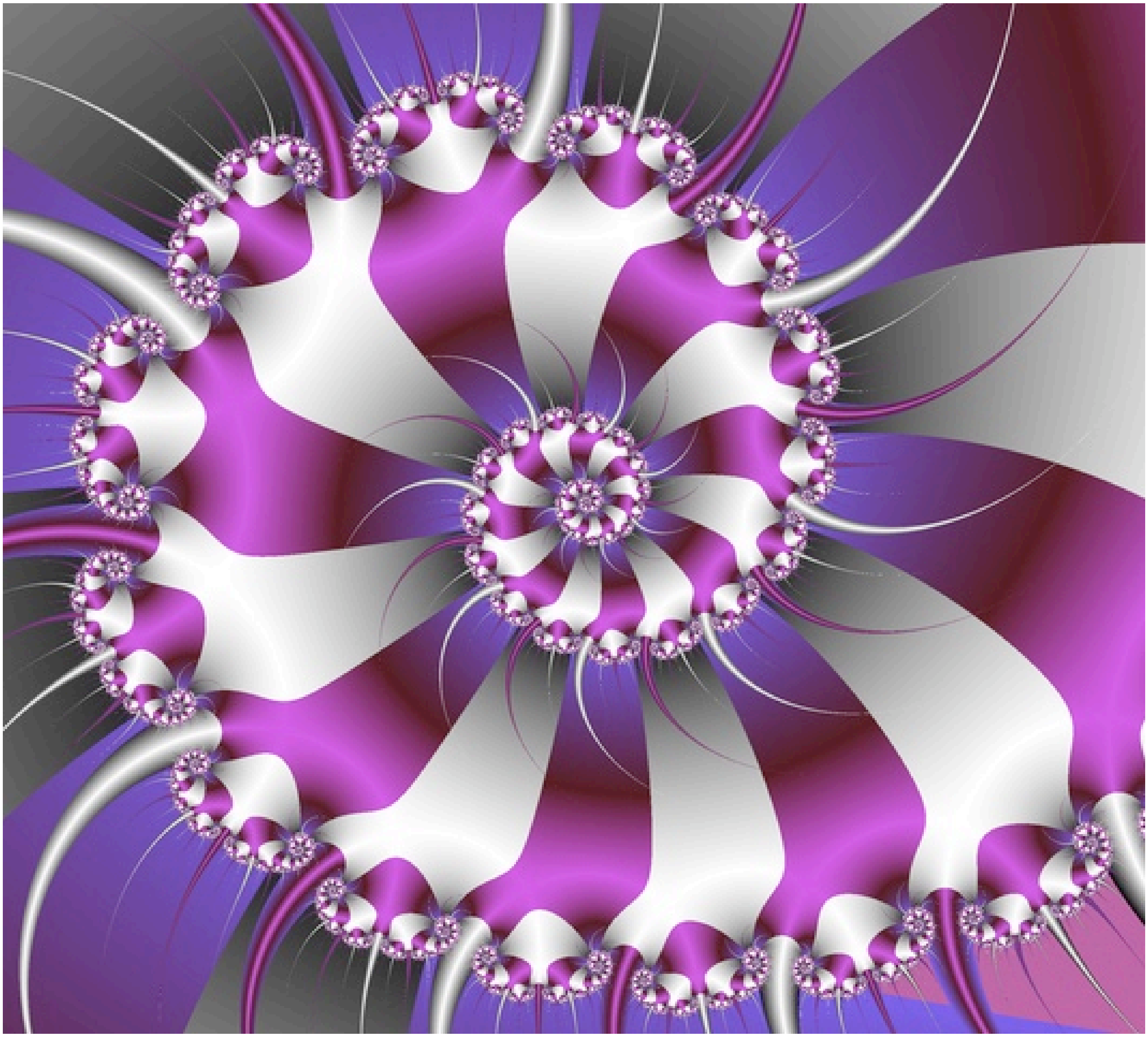


Fractal Domains

User Manual

© 2004, Dennis C. De Mars, All Rights Reserved

1. Introduction

Fractal Domains is a program for generating fractals on the Power Macintosh. It will only run on MacOS-compatible computers equipped with the PowerPC microprocessor. Fractal Domains takes advantage of the floating point capabilities of the PowerPC to allow rapid zooms into fractal images. Fractal Domains v2.0 can generate a virtually infinite variety of fractals, in fact any fractal based on iteration of a rational function. (Even more fractal types will be added to future versions of Fractal Domains.)

Fractal Domains v2.0 is a Carbon application. This means that it will run natively on both Mac OS 9 and OS X. The appearance of Fractal Domains interface elements, of course, differs with the operating system used. This manual contains illustrations using screen shots from both operating systems.

1.1 What is a fractal?

There is a precise mathematical definition of the word “fractal.” I am not going to give that definition here (see the references at the end of the manual). Informally, a fractal is a geometrical object that contains detail at all scales. A conventional geometric object becomes “smooth” as you magnify any particular part of the object. A fractal, however, contains areas that never become smooth at any magnification. In fact, some fractals, such as the famous Mandelbrot set, actually appear *more* complex at smaller scales.

The term “fractal” is also commonly used, very loosely, to refer an image created by assigning colors to points near the boundary of a fractal. These images often contain patterns of astonishing complexity and beauty.

1.2 What can I do with Fractal Domains?

With Fractal Domains, you can explore many different fractal images, including the Mandelbrot set and the associated Julia sets. By simply pointing and clicking, you can magnify a portion of a fractal to examine an interesting detail and also reveal other, smaller details that weren’t visible before. Borrowing a term from photography, magnifying a portion of a fractal in this manner is called **zooming**. By successively zooming into a fractal, new patterns previously unseen are constantly revealed.

In order to make these patterns visible, colors must be assigned to the numbers generated by the mathematical formula that defines the fractal. Fractal Domains lets you create this assignment in any manner you please. You can use a predefined color mapping, generate an endless series of random color mappings, or use the **Color Map Editor** to create a new color map or modify any predefined or randomly generated color map.

Zooming into a fractal is like a voyage of discovery. Choosing the coloration of a fractal allows you to apply your own individual esthetic sense to determine the ultimate appearance of the image. It is the combination of personal control of the visual palette with the impression of getting a glimpse of unknown worlds that makes fractal exploration unique and compelling.

Fractal Domains provides options to produce different kinds of images from a single fractal plot. There is a choice of dwell options, plus the orbit trap option, all of which use the details of the orbit generated by the fractal formula at each point to create a wide variety of patterns.

If you create an image that you really like, Fractal Domains enables you to render a high quality version of the image with the anti-aliasing capability of the **Render Image…** command. If you want to create a large image for use with a large display or a high-resolution printer, Fractal Domains enables you to produce large images without requiring large amounts of RAM. You can simply spool the image to disk using the spooling capability of the **Render Image…** command.

Fractal Domains gives you the ability to create a limitless succession of images that are always intriguing and often beautiful.

2. Basic Concepts

In order to understand Fractal Domains you need to understand dwells, regions, color maps and the interaction between them.

2.1 Dwells

Mathematically, fractals of the type currently displayed by Fractal Domains are defined by an iterative formula defined on the set of complex numbers. The dwell for a particular complex number is defined as the number of times the iterative formula can be recalculated starting at that complex number before the distance of the result from the origin exceeds a predetermined radius. The members of the Mandelbrot set (or of a Julia set) are those complex numbers that have infinite dwell values. In practice, we set a limit on the number of iterations and if we reach that limit without exceeding the radius, we assume the dwell value is infinite.

Interesting images are created generally by assigning a color to each dwell value. In Fractal Domains we call this assignment a color map.

Although technically a fractal consists of the border of those points for which the dwells are infinite, it is usually the non-infinite dwell values near the border of the fractal that give rise to interesting images. In this document, then, we will informally refer to the entire set of points with dwell values assigned to each point as a fractal.

2.2 Regions

Fractal Domains allows a fractal to be split up into regions. Each point in a fractal belongs to exactly one region. The main feature of regions is that each region has its own color map. The criteria by which a fractal may be split into regions vary for each fractal. (Right now there are only two types of fractals supported by Fractal Domains, Mandelbrot and Julia, and they happen to have the same set of criteria for region splits, but in the future there will be other fractal types supported by Fractal Domains with some different ways of splitting into regions).

To give an example: the default Mandelbrot fractal has only one region, the exterior (that is, all points that do not have infinite dwell values). However, using the "Dwells" tab panel of the "Parameters…" dialog box, we can split the exterior into two regions: all points with even dwell values are in one region, all points with odd dwell values are in the other region. We can then apply completely different color maps to each region.

2.3 Color Maps

Every region has a color map. Predefined color maps can be assigned to a region from the Palette menu. A random color map can be assigned by choosing Randomize from the Palette menu, clicking the Randomize button in the Randomizer window, or by pressing Command-R. Any of these color maps may be edited in the “Color Map Editor” window. The structure of a color map is as follows: a color map contains between two and thirty-two color breaks. A color break is a dwell value and an associated color. The color breaks are ordered by increasing dwell value.

Points in a region are colored as follows: if the dwell at a point is in the color break list, the point is assigned the corresponding color for that color break. If not, then the dwell generally lies between two color breaks. In this case a linear gradient of colors is generated between the two color breaks and the point is assigned the color corresponding to the dwell's position along the color gradient.

This color map can be modified by a power law if the “mapping exponent” is different from one. If the mapping exponent is “g” then a mapped dwell is generated according to the equation:

$$d' = D\left[\frac{d}{D}\right]^g$$

where

d' = mapped dwell value
d = original dwell value
D = maximum dwell value in region

The mapped dwell is then looked up in the color map and the resulting color is assigned to the point.

2.4 Orbit Traps

As mentioned before, dwell values for each point are calculated by iterating a formula over and over. If we call the iterative function R and the initial point is z_0 then a sequence of points is generated by applying R over and over again:

$$z_1 = R(z_0),\ z_2 = R(z_1),\ z_3 = R(z_2),\ \ldots,\ z_N = R(z_{N-1})$$

The end of this sequence is reached either when z_N exceeds the escape criterion or when N reaches the iteration limit. The entire set of points

$$\{\ z_0, z_1, \ldots, z_N\ \}$$

is called the **orbit** of z_0. Thus, every point in a fractal plot has a unique set of points in the complex plane associated with that point.

When we assign a number to the point based on the number of iterations before the escape criterion was satisfied, we are extracting one property of the orbit — the total number of points in the orbit. However, there are obviously many properties of the orbit that could be used to choose values (and indirectly, colors) for the point. This is a lively and fertile area of investigation for fractal-generating programs.

One way of deriving effective images from the orbit uses the concept of an **orbit trap**. Some fixed shape is defined in the complex plane, and if at least one point in the orbit lands somewhere in that shape, the original point that started the sequence is considered to be trapped by that shape, and is assigned a color in some special way.

2.5 Fixed Points and Critical Points

2.5.1 Attractive Fixed Points

It will help you to use certain features of Fractal Domains if you understand a little bit about these special points in the complex plane.

A point z is said to be a **fixed point** of a function f if the following relation holds:

$$f(z) = z$$

Furthermore, if points near the fixed point z tend to converge to z when the function f is iterated, z is said to be an **attractive fixed point**.

Infinity can be an attractive fixed point in a sense that can be made mathematically precise (but we won't go into details here).

Fractal Domains figures out the attractive fixed points for many functions and allows you to use convergence to a fixed point as a criterion in applying color maps. For some fractals, such as Newton and Halley Julia sets, this ability is crucial since often infinity is not an attractive fixed point for these fractals (meaning all orbits are bounded).

For those fractals that allow fixed point convergence, a convergence radius can be set. An iteration is terminated if the orbit passes near a fixed point within this radius.

For many fractals (especially Newton and Halley fractals) there will be multiple attractive fixed points. For these fractals, the complex plane can be split into regions depending on which fixed point an initial point converges to. The "Fixed Point" panel in the Rational, Newton and Halley fractal types offer the option to do this, and the color map editor can then be used to color these regions independently.

2.5.2 Critical Points

A point z is said to be a **critical point** of a function f if the following relation holds:

$$f'(z) = 0$$

Critical points are very important in the study of the dynamics of iterated functions. For rational functions, the critical points are generate "Mandelbrot-like" fractals. See section 3.2 for more details on this.

For fractals of the Rational, Newton and Halley types, critical points can be used as the origin of orbit traps. Orbit dynamics become especially complicated near critical points, so marking points with orbits that pass near critical points produces particularly interesting images.

2.6 Fractal Domains Documents

Several types of files can be created by Fractal Domains. Parameter files and image spool files can be re-opened by Fractal Domains. Images can also be saved in two popular graphic file formats (PICT and PNG) for export to other applications.

2.6.1 Parameter Files

Parameter files contain all of the mathematical information needed to calculate a fractal and all of the color mapping information needed to reproduce the image. Optionally, a parameter file can hold the dwell values of a fractal, obviating the need to regenerate these values, if **Save Dwells** is checked in the **Preferences** window. If the dwells are not saved, parameter files are a very compact way to store all the information about your fractal.

If you want to explore or modify your fractal further, you must save it as a parameter file. The other types of files generated by Fractal Domains (spool files and exported image files) do not allow you to further modify the fractal.

2.6.2 Image Spool Files

2.6.2.1 Spooling Large Images to Disk

Image spool files are files that store the result of fractal image generation to a disk file. They can be used to generate images that are too large to fit in memory at once, and they can also be used to generate images that take such a long time to generate that it is inconvenient to generate them in one session. When an image is save to a spool file, generation of the image can be interrupted and resumed at a later time. Often, both of these goals apply; when an image is too large to fit in available memory, it generally takes a long time to generate.

The size of the image in the spool file is not limited by the amount of memory allocated to Fractal Domains. The spool file window can be used to view the image, but the entire image is never in memory at one time.

The spool file itself is not an image file, that is it is not in any standard graphics file format. However, it contains all of the image information and can be used by Fractal Domains to export the image to one of several standard graphics file formats..

If spooling to file is interrupted by closing the spool window or quitting Fractal Domains, generation of the image can be resumed at any time by opening the file with Fractal Domains. All of the information required to generate the fractal is stored in the spool file along with the image data. Thus, you can generate very large images without having to insure that the computer is continuously up and running for long periods of time. In fact, if a crash occurs during generation of a spool file there is a good chance that the image generation can still be resumed.

When a spool file is first created, Fractal Domains does not check to see if there is enough free space on the volume for the complete spool file. This is actually a good thing, as you can start a spool file and clear the space for the whole file later. If there is not enough space when you start, make sure you clear more space before the disk is full, or stop the image generation before the disk is full, move the spool file to another volume, and resume image generation.

The space required by the complete spool file can be simply calculated. If the height and width of the image are h and v, then the spool file will require (approximately) 3*h*v bytes of storage.

The elapsed time is stored in the spool file; thus, if you resume image generation at a later time, the estimated time to completion can still be calculated.

Disk accesses usually happen whenever the spool file window needs to be updated. If this becomes inconvenient, the spool file window can be collapsed by clicking in the collapse box in the upper right hand corner of the window if you are using MacOS 8, or by triple-clicking the title bar if you are using System 7.x and have WindowShade enabled. Collapsing the window does not interfere with the image generation process.

2.6.2.2 Automatic Image File Export with Spool Files

Once a spool file is finished generating an image, you generally want to export the image in a standard graphics file format, as the image contained in a spool file can only be displayed by Fractal Domains. It would not be exaggerating to say that the only use for a spool file, once the image is completely generated, is to export a graphics file from it.

Normally, a graphics file is exported by manually choosing one of the **Save as [format]…** menu commands. However, since spool files are often used in batch mode (that is, you start them off and then go do something else until they are finished) and since the spool files themselves are generally much larger than the exported graphics files, it is sometime convenient to use an option that allows you to have the export done automatically when the image is completely generated and delete the spool file after the export is completed.

As you can see in the figure in section 2.6.2.4 The Spool Bar , an “Auto Export” menu is available in the header bar of each spool window. The default is “Don’t Export” but you can select any of the four formats Fractal Domains supports (PICT, PNG, TIFF, JPEG). When one of these is selected, the fractal image will automatically be exported to that format when the image has been completely generated. The image file will be given the appropriate extension and will be created in the same directory as the spool file.

If an auto export format is selected, the “Delete Spool File after Export” checkbox is enabled. If you check this box, the spool file will be deleted after the image file is created. You probably will want to check this most of the time when auto image export is used.

2.6.2.3 Serial Generation with Spool Files

Fractals generated in the usual fractal editing windows, along with fractals generated to spool files, are normally generated in parallel. That is, the computer's processing time is shared among all open windows that contain actively generated fractals.

Parallel generation may not be ideal in all situations. Spool files are generally used for images that take a long time to generate, and many spool files may be open at once in order to generate several images in "batch" mode, since spool file generation is not memory intensive. In this case, parallel generation would result in all images finishing at about the same time (if they took about the same time to generate).

In a case like this, you might want the spool file images to be generated one at a time rather than in parallel, so that you could see the first complete image much more quickly. At the same time, you might want to have several spool files open so that a new image could start generating when the previous one finishes, without having to attend the computer to start each image. In other words, you might want the images to be generated sequentially, or serially, rather than in parallel.

You can arrange sequential generation of spool files by checking the box labeled "Generate Serially" in the Spool Bar (see next section) of each spool file to be generated serially.

All spool files with the "Generate Serially" option chosen are considered to be members of a single group. Only one of the files in this group will be given generation time at any particular instant. Of the files in this group, only the file belonging to the front most window that has not yet completed will be given time to generate its image. After it is completed (or closed) the next most front most uncompleted window will start generating.

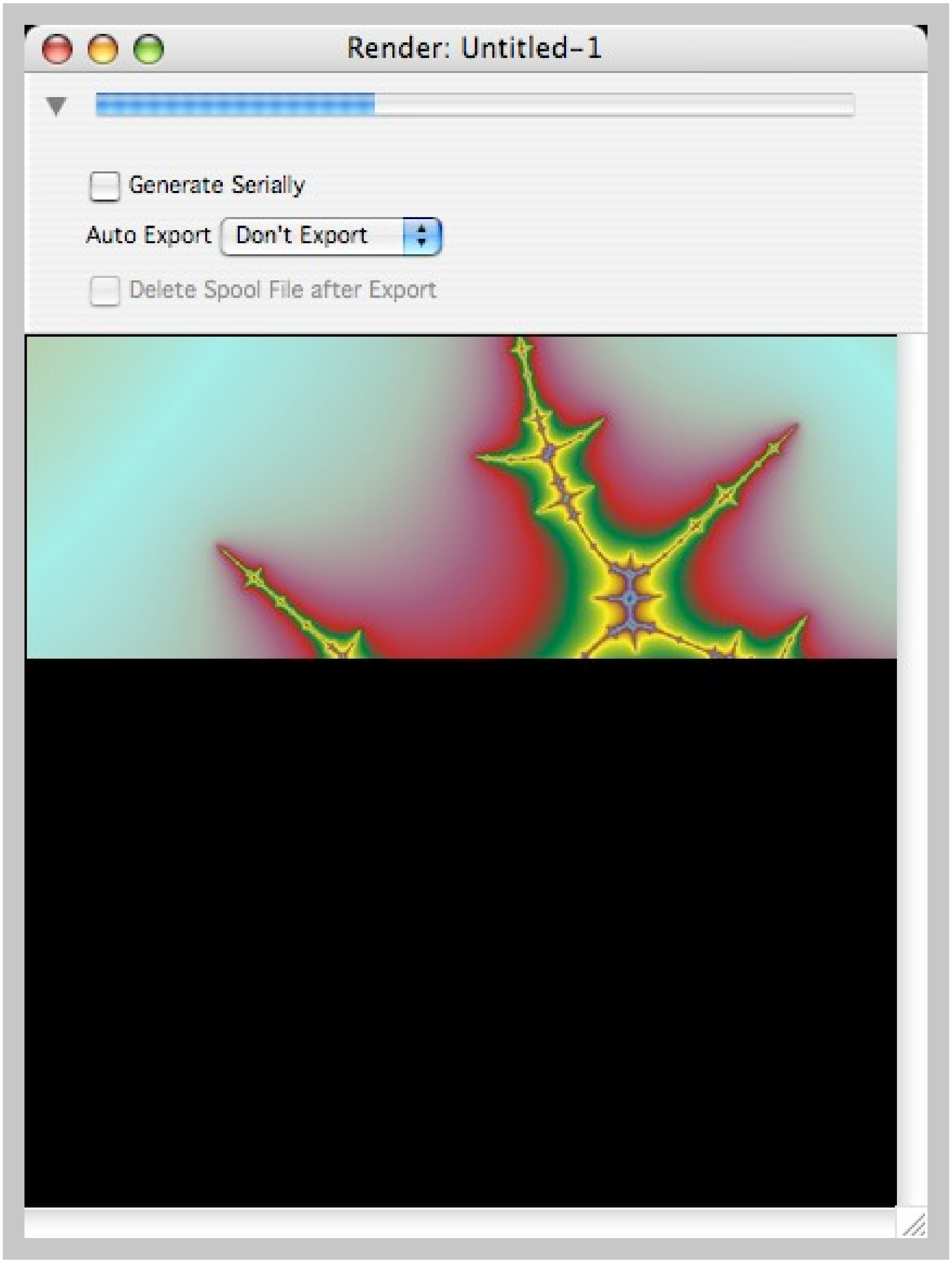
Any open fractal edit windows that have not completed, along with any spool files that don't have the "Generated Serially" option selected, will still be generated in parallel along with the one file in the "Generate Serially" group that is currently generating.

This option is especially useful in conjunction with the "Auto Image Export" option. If a group of spool files have both options checked, with the "Delete Spool File after Export" option also chosen, then one spool file at a time will generate an image, export the image file, and then delete itself. In this case the disk used to store the images only needs enough room for one spool file plus all of the exported image files at any one time. Since spool files are generally much bigger than the exported image file, this can be quite useful for very large images.

2.6.2.4 The Spool Bar

Every spool window contains a progress bar in the header of the window that indicates how much of the image has been generated. We refer to this header as the "spool bar." To the left of the progress bar is a disclosure triangle. Clicking this triangle reveals some additional controls that display options set when the spool file was created with the **Render Image…** command. (See figure below).

These options can be changed on the fly by setting the desired options in the controls. It must be remembered that any changes are only in effect during the current session. If the spool file is closed or if Fractal Domains is quit before the image is finished generating, then the next time you open the spool file, these options will revert tot he value they had when the spool file was created.



2.6.3 Exported Graphics Files

Fractal images in Fractal Domains can be exported to graphics applications. Images in fractal windows or "Render Image" windows can be exported. The resulting file cannot be opened by Fractal Domains, but can be opened by other graphics-oriented applications.

Currently, Fractal Domains supports export of image files in four formats:

PICT The native graphics format of the Macintosh. Practically every program that uses graphics in some manner can read PICT files. For many purposes this is the only graphics format you may need. However, it has three inherent drawbacks:

1. PICT files have an intrinsic limitation on the width of the image; images larger than 4095 pixels in width should not be saved as PICT files. Any attempt to save a larger image as a PICT may lead to a crash. 4095 pixels is very large, but some applications calling for high-resolution images might need larger widths. PICT format cannot be used for these applications.
2. PICT format applies only very modest compression to images. Other formats can do much better.
3. PICT is not a good format if you need to transport your images to different platforms such as computers using the Windows or Unix operating systems. There may be utilities on your target platform to convert PICT files to a native format, but other formats, designed for interchange between different platforms, will be more convenient.

You can export a fractal image as a PICT file by using the **Save As PICT…** command in the **File** menu.

PICT images will be saved with the same pixel depth (8, 16, 24) that is in effect for the fractal window. Rendered image (i.e. spool files) always have 24-bit pixel depth.

PNG PNG (Portable Networks Graphics) is a file format that was created relatively recently. PNG is a lossless image format (lossless means all image information is stored and the original image can be perfectly recreated). PNG was added in order to allow larger image widths than PICT allows; it also provides much better compression than the PICT format.

You can export a fractal image as a PNG file by using the **Save As PNG…** command in the **File** menu.

Although the PNG format can support different pixel depths, currently Fractal Domains always exports images to PNG with 24-bit pixel depth.

TIFF TIFF is a lossless image format that can be interpreted on a wide variety of platforms by a wide variety of applications. TIFF was added to Fractal Domains for compatibility purposes; TIFF has been around a long time and is a standard file interchange format. I had always intended to add this format at some point, but I was prompted to include it now because I have received reports that some popular graphics applications may have problems with large images in PNG format (even though Fractal Domains is generating these files correctly). TIFF provides an additional option for users that cannot use the other formats.

Unfortunately, only the uncompressed TIFF format is provided. Currently, the only widely understood compressed TIFF format uses the LZW algorithm, which is patented. (GIF files use the same algorithm, which is one reason why Fractal Domains does not export to that format either.) For this reason, PNG is the preferred format when it can be used (its compression algorithm is superior even to compressed TIFF, so you get smaller files using that format anyway). Uncompressed TIFF is available as an option in cases where no other format can be used.

You can export a fractal image as a TIFF file by using the **Save As TIFF…** command in the **File** menu.

JPEG Unlike the other three formats, JPEG uses "lossy" compression, which means that some of the original image information is lost. This can cause visible artifacts in certain kinds of images, including some fractal images, so this format should be used with caution. However, for many images JPEG can provide astounding compression ratios with little or no visible effect on the image.

JPEG is a standard format used in web design; all web browsers that can display graphics at all can display JPEG. The high compression ratios with 24 bit color images makes JPEG very popular for displaying fractal images on the web.

I personally have found that often the final step creating a fractal image with Fractal Domains has been to translate the image to JPEG format. Now, this step can be eliminated as the image can be stored directly as JPEG from Fractal Domains.

Since JPEG is lossy, the user may want to use one of the other formats for generating fractals that will be archived or manipulated in some other way. Among the other three formats, PNG provides the best compression and versatility in a lossless format.

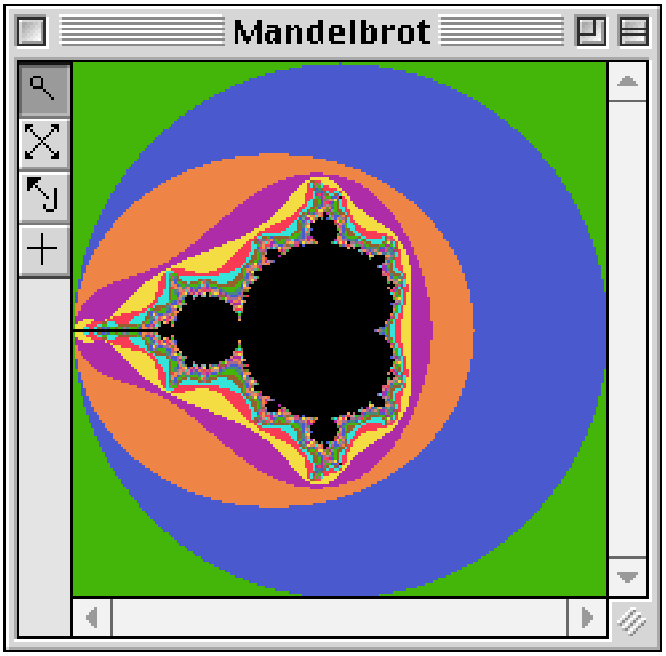
There is a "quality" parameter that affects the image quality of JPEG images. This parameter can vary from 1 to 100. Lowering the value of this parameter increases the compression achieved at the cost of image quality. This parameter can be set in the "Preferences" dialog. The default is 95 which is about optimum for high quality images (increasing to 100 makes the file bigger but doesn't substantially increase image quality.) A value of 75 is about as low as you can go without noticeable image degradation in most images.

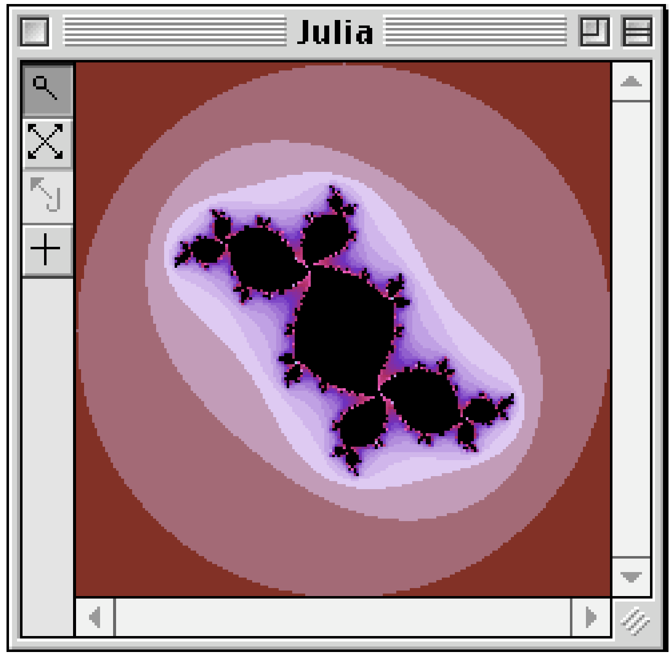
You can export a fractal image as a JPEG file by using the **Save As JPEG…** command in the **File** menu.

3. Fractal Types

Fractal Domains allows you to explore many different kinds of fractals; in fact, a virtually infinite number of expressions can be specified, each one generating a unique fractal. The fractals generated by Fractal Domains 1.3 are divided into four categories, which are described in the following sections.

3.1 Mandelbrot





This is the famous Mandelbrot fractal, studied intensively by mathematicians and explored extensively by fractal enthusiasts all over the world. The Mandelbrot set (show on the left above) is generated by iterating the formula

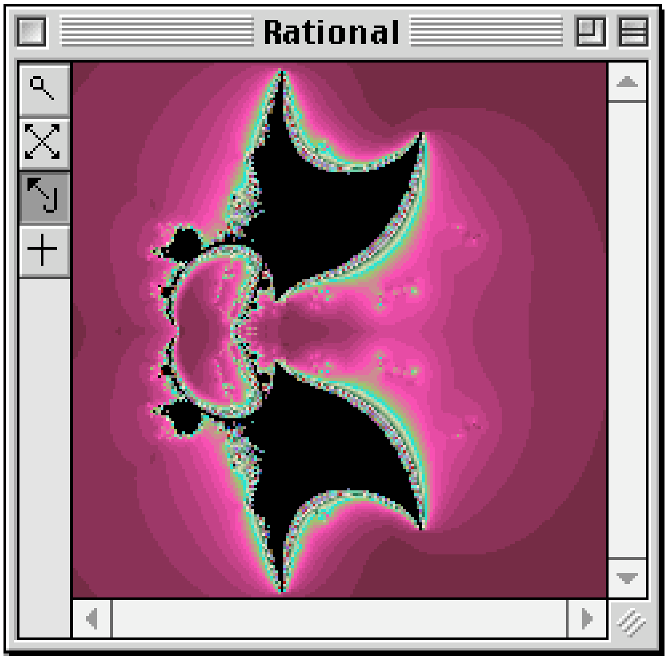
$$z_{n+1} \leftarrow z_n^2 + c$$

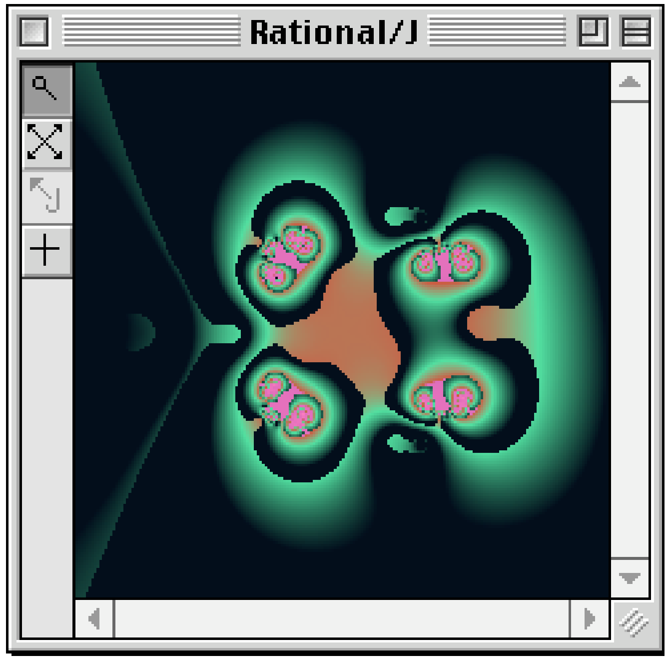
for each point c in the complex plane. The point is considered to be in the Mandelbrot set if the sequence of points generated in this manner is bounded.

The Julia set corresponding to a point c is found by iterating the above equation, keeping c constant and letting z range over the complex plane. Again, if the orbit is bounded then the point z belongs to the Julia set. Every point in the Mandelbrot set corresponds a Julia set that is connected. This can be seen by looking at the Preview window while moving the cursor over the Mandelbrot set. The Preview window shows the Julia set corresponding to the point at the cursor.

If the Julia tool is selected, clicking on any point in the Mandelbrot window will produce a new window with the corresponding Julia set.

3.2 Rational





Rational fractals generalize the method used to generate the Mandelbrot set and its corresponding Julia sets. Rather than iterating a simple quadratic function in the complex plane, an arbitrary rational function may be specified for the iterative function.

A *rational function* is defined as any function that can be written as the ratio of two polynomials. Thus, the general form for a rational function is:

$$R(x) = \frac{a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n}{b_0 + b_1 x + b_2 x^2 + \cdots + b_m x^m}$$

Every choice of coefficients $\{\mathbf{a}_n, \mathbf{b}_m\}$ defines a rational function.

Fractal Domains can calculate what is known as a “connectedness locus” for any rational function. This is a generalization of the Mandelbrot set. The Mandelbrot set is calculated by iterating the equation

$$z_{n+1} \leftarrow z_n^2 + c$$

initializing z to zero and letting c range over the complex plane. When the sequence generated this way is bounded, the point c is in the Mandelbrot set. The corresponding Julia set is then known to be connected (an informal definition of connected is “all in one piece”).

For an arbitrary rational function, the same thing can be done, but instead of initializing z to zero, every finite critical point of the function must be tried (in the case of the Mandelbrot set, the only finite critical point is zero). The resulting set is a map of connected Julia sets. You can see this by dragging the cursor over such an image and watching the Preview window, which shows the Julia set for that point. You only see large areas of black in the Preview window when the cursor is over a black portion of the original fractal image.

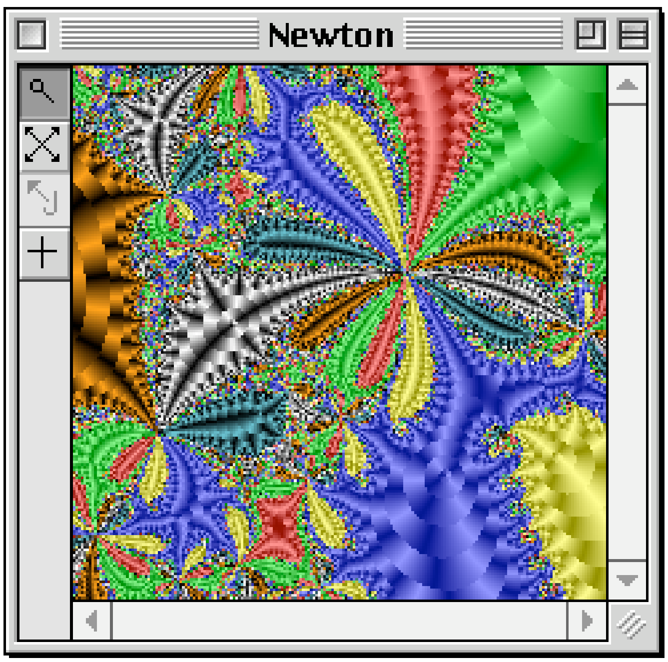
For a rational function R(x), the corresponding fractal is generated by iterating the expression

$$z_{n+1} \leftarrow R(z_n) + c$$

The sequence is initialized with each of R(z)'s critical points in turn. If all sequences are bounded, the point c is considered to be part of the fractal set.

Clicking on any point in such a fractal with the Julia tool will produce the corresponding Julia set. In this fractal, the expression above is iterated with constant c and with z ranging over the complex plane.

3.3 Newton



An interesting type of fractal is based on formulas derived from applying Newton's method. Newton's method is a technique for finding the root of a function *f* if its derivative *f'* is also known. Given an initial guess, Newton's method produces successive approximations to the root by iterating the formula:

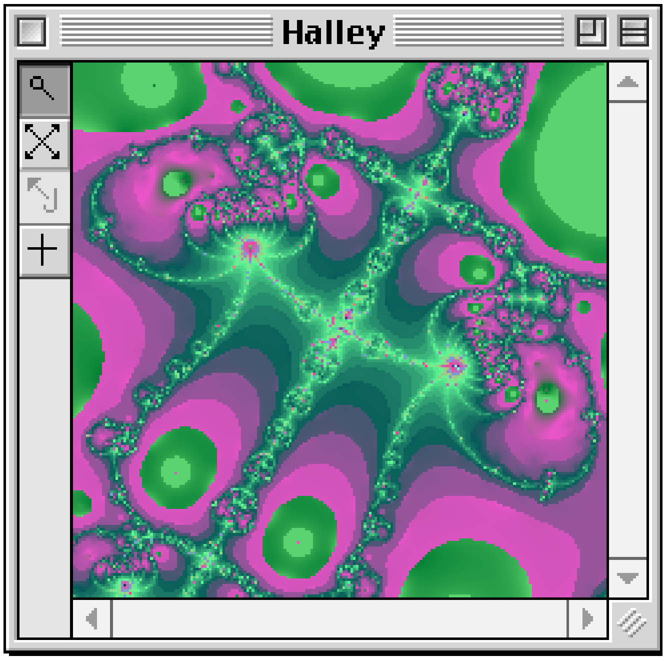
$$x_{n+1} \leftarrow x_n - \frac{f(x_n)}{f'(x_n)}$$

When the initial guess is close to a root, this method converges rapidly. For an arbitrary point, it can bounce around a bit before it converges, and it is difficult to predict which root it will eventually converge to (if any). All initial guesses that eventually converge to a particular root a

In the special case where *f* is a rational function, the expression above is also a rational function (since the sum, difference, product and quotient of any two rational functions is a rational function, and the derivative of a rational function is also a rational function).

Thus, the Rational fractal type is already capable of generating a large number of fractals of this type. However, to generate a Newton fractal for an arbitrary rational function f would require calculations that are quite laborious for all but the simplest function. So, Fractal Domains provides a distinct Newton fractal type. You enter an arbitrary rational function, just as you do for the Rational fractal type, and Fractal Domains substitutes it for f in the expression above and uses the rational function thus derived to generate the fractal.

3.4 Halley



Halley's method is an approximation technique that is similar to Newton's method. Halley's method uses knowledge of both the first and second derivatives. The method works by iterating the following expression:

$$x_{n+1} \leftarrow x_n - \frac{f(x_n)}{f'(x_n) + \frac{f(x_n)f''(x_n)}{2f'(x_n)}}$$

Just as in the case of the Newton fractal, if a rational function is substituted for f in the expression above, the resulting expression is still a rational function. Even relatively simple functions can yield complex expression with Halley's method, but Fractal Domains allows you to specify any rational function for f and does the hard work of deriving the iterative expression for you.

4. Techniques and Tips

The number of options available in Fractal Domains for modifying images can be daunting, and you may at first find it difficult to produce image similar to those you can find on the web at my gallery or the other excellent fractal galleries. Here you can find some guidance to get you started. Those of you who have had some experience with other fractal programs may want to skip the first two sections.

4.1 Starting Out — Zooming Into The Mandelbrot Set

All of the interesting patterns are at the boundary between the black part of the set (the points that actually belong to the set) and the exterior. Zooming into different spots along the boundary reveals different patterns. Appearance of a pattern can be changed dramatically simply by changing the color map. When you find an interesting area, cycle through some random color maps and see how different color maps often bring out different aspects of the fractal.

When you bring up the initial Mandelbrot image, the dwell limit is set at 100. As you zoom in, you will find that you need to increased this value, as some of the black points you see are not really in the set. Generally, you have to increase the dwell limit as you zoom deeper, which in turn means the deeper zooms will take longer to calculate.

If the random maps you are getting for a particular area seem too “busy” you can reduce the maximum number of colors in a random color map via the “Image” dialog (see section 5.4.2.1).

4.2 Julia Sets

The Mandelbrot set is actually a kind of map of Julia sets. Choose the Julia tool and click anywhere in the black part of the Mandelbrot set and a window will open on a Julia set associated with that point. Every point is associated with a unique Julia set.

Points in the Mandelbrot set produce Julia sets that are connected. Points outside the Mandelbrot set produce Julia sets that are not connected. Point near the boundary of the Mandelbrot set produce the most complex and interesting Julia sets. Try click a few different points about the Mandelbrot set boundary and note the variety of Julia sets that are produced.

Technical note: when you look at a connected Julia set in Fractal Domains, the Julia set is actually the points on the boundary of the solid black figure you see. The figure including the interior is sometimes referred to as the filled-in Julia set.

Unlike the Mandelbrot sets, Julia sets possess an exact self-similarity; that is, a small segment of a Julia set (that is, the boundary) contains exact replicas of itself, scaled down and rotated. For this reason, deep zooms of Julia sets are not as interesting as deep zooms of the Mandelbrot set, because after a certain point you start seeing the same thing over and over.

4.3 Dwell Options

The **Dwells** panel of the **Parameter…** dialog box provides some important tools for producing special effects in your fractal image. Refer to section 5.4.1.2 for a complete definition of the options in this panel.

A common and very effective way to show the shape of the dwell regions is to alternately color areas of odd dwell values and even dwell values with different colors. This produces the distinctive “zebra” patterns that you may have seen in some fractal illustrations. This can be produced in Fractal Domains by checking the **By Iterations (Even/Odd)** in the **Dwells** panel. This splits the fractal into two disjoint regions that can have separate color maps. The result is usually a “banding” effect.

Often the dwell bands are too narrow to produce a good effect. They can by made wider by setting the **Skip By** parameter to a large value. Try several values, clicking Apply each time, to see the effect.

Changing the dwell method can produce interesting variations. **Continuous Potential** produces dwell values that vary continuously over the plane, as opposed to the conventional **Escape Time** method, which produces discrete bands. This method works best if the **Escape Radius** is set to something larger than the default value — 256 is a good choice.

Angular Decomposition, or the **Argument** method, is another continuous method. Color maps with a lot of color breaks usually don't work well with this method. Try some of the built-in gradients first. If you want to try some color maps, set the maximum number of breaks in the Map panel (section 5.4.2.1) to a small number — from two to five works well.

4.4 Using Orbit Traps

Orbit traps are fun to play with. Sometimes when you turn them on you just get a mess, but often they can liven up an otherwise uninteresting image. See section 5.4.1.3 for a complete description of the options available in the orbit trap panel.

The dwell options discussed in the preceding section are not in effect when orbit traps are on — the escape time method is used for points that are not trapped. I suggest simply setting the exterior region to all black when using orbit traps.

The most basic parameter to set in the orbit trap panel is the **Trap** item that chooses the shape of the orbit trap. The **Cross** produces the classic "stalk" shapes, and the real & imaginary stalks are subsets of the cross trap. Circles and squares are the other choices.

The outer trap extent is the next most important parameter. You will almost always want to change this parameter from its default value. The default value of 0.001 gives reasonable results for the stalk traps, but will almost always be too small for the circle or square traps. Even for stalks, the best value can easily be an order of magnitude more or less.

The best way to adjust this is to drag the **Parameters…** dialog so that your fractal window is visible, and then try different values for the outer trap extent, hitting the **Apply** key each time. If you are going to try circle or square traps, you might want to nudge the outer trap extent up a couple of orders of magnitude before the first try.

The inner trap extent is usually left at zero. For circles and squares, making this value non-zero produces and interesting "doughnut" effect.

The most useful settings for the **Region Split** are "none" and "By orbit (even/odd)." After setting this, you will need to set a different color map for each region to see a difference.

Choice of color map influences the effect produced. Orbit traps can produce a striking 3D effect. The most straightforward way to achieve this is to use a two-color gradient for the trap color map. If you are applying random color maps, set the maximum colors to "two" to begin with. You can try upping the maximum to three or four after you've tried a few two-color gradients.

As in the case of the argument dwell method, color maps with a lot of colors generally don't work well with orbit trap regions.

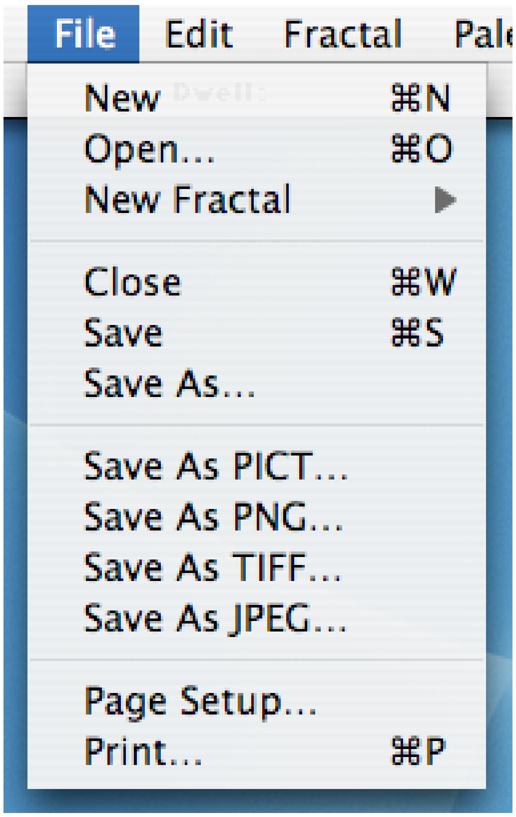
If you are applying random color maps and you have specified a regions split, you may see a map come up for one region that you like a lot, but you may not care for the maps generated for the other regions. In this case, you can randomize the rest of the regions individually by turning off **Randomize All Regions** in the **Map** panel (see section 5.4.2.1) and then choosing the regions to colorize one by one in the **Color Map Editor**.

5. Fractal Domains Reference

5.1 About Dialog

Choosing “**About Fractal Domains...**” in the Apple menu will bring up the “About…” box. This shows the Fractal Domains version number and also allows you to enter the registration number for the program. For a complete description, see section 6.2 on entering your registration.

5.2 File Menu



New	Creates a new fractal window. The default fractal type is Mandelbrot. If you want to create a Julia set image, start by creating a Mandelbrot set image with **New** and then create a Julia set using the Julia tool (see section 5.3 for details). If you want to create any other type of fractal, use the **New Fractal** menu item. A new fractal has dimensions of 200 x 200, the default color map, and all special options such as orbit traps turned off.
Open…	Open a fractal parameter file or a fractal image spool file.
New Fractal	Leads to a hierarchical menu, permitting you to create a new fractal window using any fractal type supported by Fractal Domains. See the next section for details.
Close	Close frontmost fractal window.
Save	Save changes to frontmost fractal window to file.

Save As... Save frontmost fractal window to new file.

Save As PICT... Save image in frontmost fractal or image window as a PICT file (a standard Apple graphics format). See section 2.6.3 Exported Graphics Files for details.

Save As PNG... Save image in frontmost fractal or image window as a PNG file. PNG (pronounced "ping") stands for Portable Network Graphics format. See section 2.6.3 Exported Graphics Files for details.

Save As TIFF... Save image in frontmost fractal or image window as a TIFF file. See section 2.6.3 Exported Graphics Files for details.

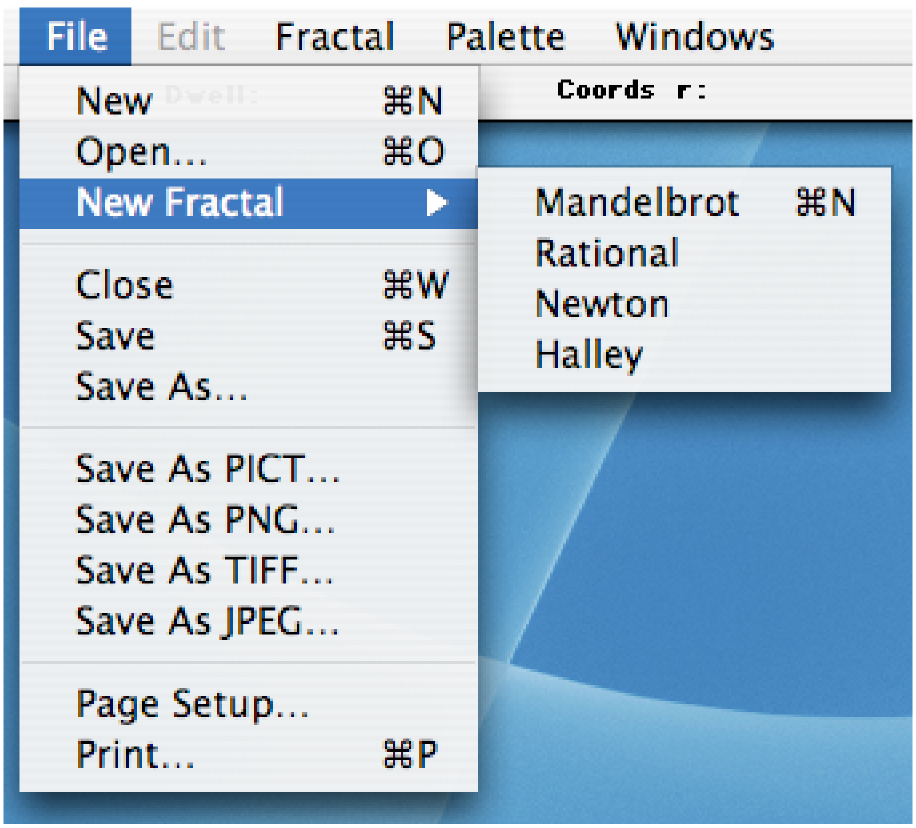
Save As JPEG... Save image in frontmost fractal or image window as a JPEG file. See section 2.6.3 Exported Graphics Files for details.

Page Setup... Bring up Page Setup dialog box to change printer settings.

Print... Print image show in frontmost fractal or image window.

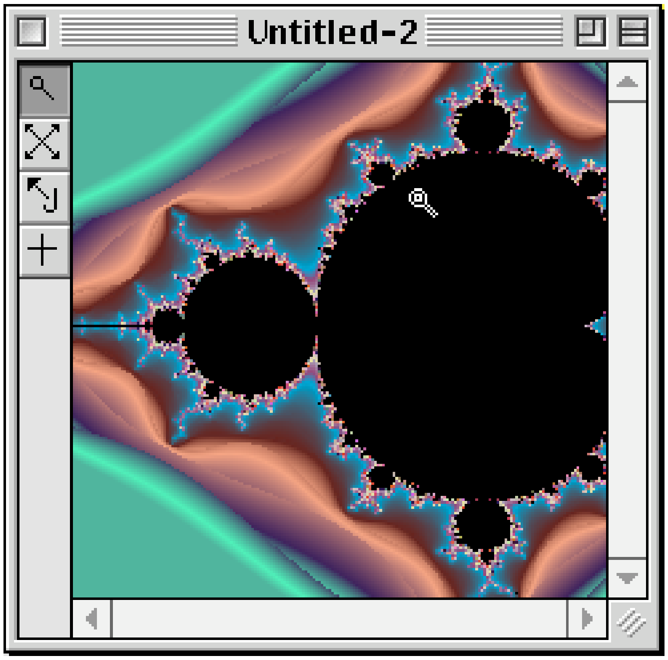
Quit Quit.

5.2.1 New Fractal



A new window with any of the fractal types supported by Fractal Domains may be created with this menu. See section 3 for a complete description of each type of fractal. The fractal is created in its own window with a set of default parameters. You can then modify the parameters using the **Parameters…** dialog.

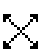
5.3 Fractal Window



Upon opening an existing fractal parameter file or by choosing New from the File menu to create a new fractal, the user will see the fractal window, containing the fractal image and a palette of some commonly used tools

The fractal window can be resized; scroll bars will appear if the window is smaller than the total image size.

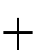
Zoom-in tool
Used to zoom deeper into the fractal. The linear scale is magnified by two when this tool is clicked anywhere in the fractal, with the point clicked being used as the center of the new view. If the Command key is held down while clicking the zoom-in tool, the scale will be magnified by four rather than by two.



Zoom-out tool
Reverse of the zoom-in tool. The linear scale is expanded by two when this tool is clicked anywhere in the fractal, with the point clicked being used as the center of the new view. If the Command key is held down while clicking the zoom-in tool, the scale will be expanded by four rather than by two.

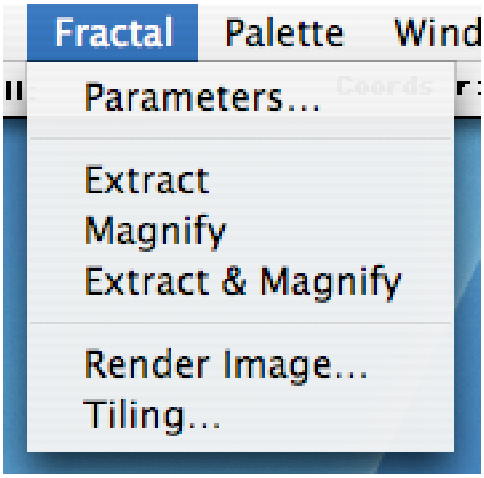
Julia tool
Used to create a new fractal, a Julia set based on the point clicked with Julia tool. Clicking on a black portion of the set (i.e. part of the Mandelbrot set itself) will create a “solid” Julia set.

The Julia tool is only enabled when viewing part of the Mandelbrot set; it is disabled when viewing a Julia set.



Selection tool
Used to select a portion of the fractal. When a selection is made with this tool, the three items Extract, Magnify and Extract & Magnify in the Fractal menu become enabled. See section 5.4 for a description of these commands.

5.4 Fractal Menu



Parameters… Modify parameters of the fractal displayed in the frontmost window. Brings up a modeless dialog box enabling the user to modify practically every aspect of the fractal. A complete description of the Parameters dialog box is given in section 5.4.1.

Image… Modify details of the way color is generated and displayed in the frontmost window. Brings up a modeless dialog box which is documented in section 5.4.2.

Preferences… Modify global preferences. Brings up a modeless dialog box which is documented in section 5.4.4.

Render Image… Create an image based on the fractal displayed in the frontmost window. Enables the user to generate a fractal image with additional options not available in the regular fractal window. Brings up a modeless dialog box which is documented in section 5.4.3.

The following three menu items are only enabled when there is an active selection in the frontmost fractal window. Each of the menu items apply an operation to the displayed selection.

Extract… Creates a new window containing the selection without changing the scale. The new fractal will have the dimensions of the selection. The original fractal window is left unchanged.

Magnify… Zooms the current window to contain the selection without changing the image area (the aspect ratio of the image will change to match the aspect ratio of the selection, however).

Extract & Magnify… Has the same effect as Magnify but opens a new window for the magnified selection. The original fractal window is left unchanged.

5.4.1 Parameters Dialog

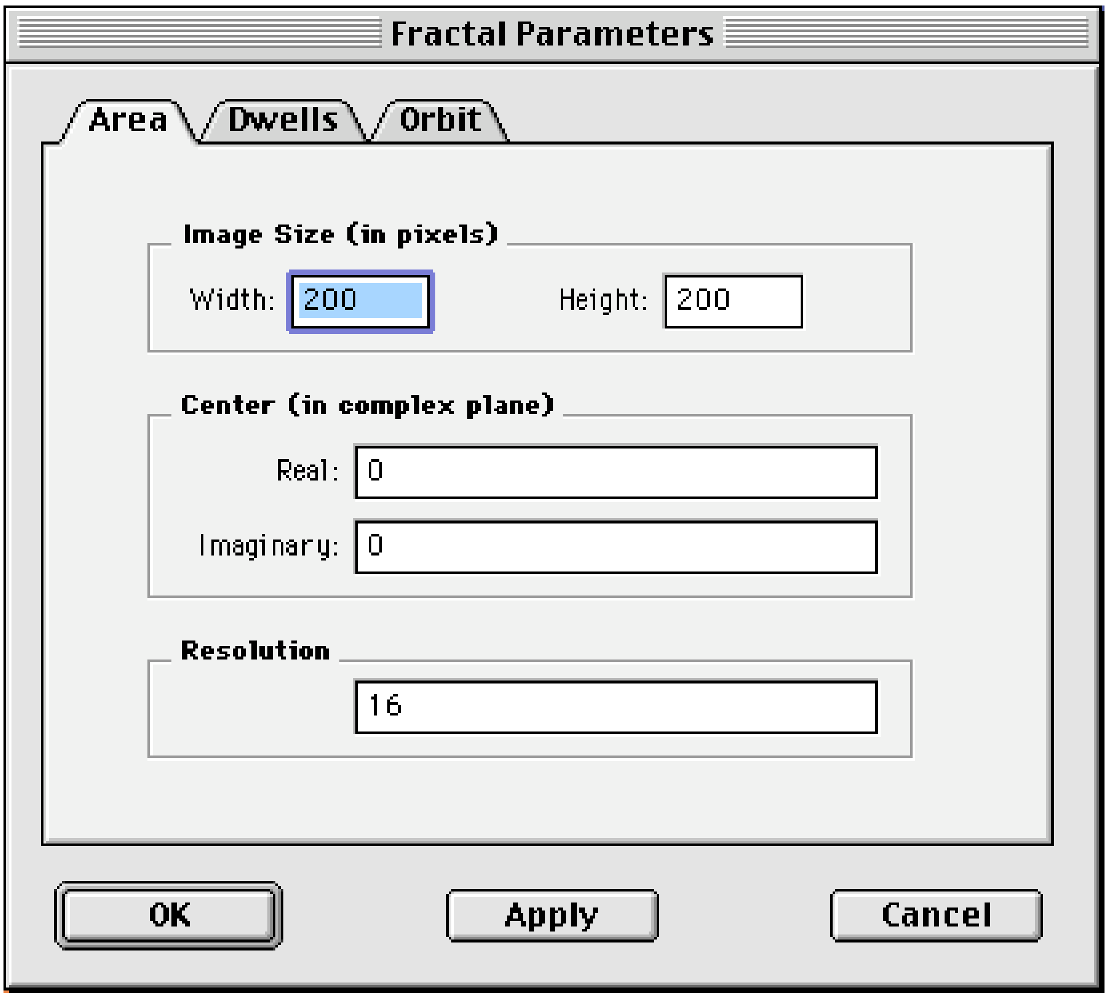
The **Parameters…** dialog box can be used to change the parameters that define the image being viewed. The Fractal Parameters dialog box contains three tab panels for a Mandelbrot set image: the **Area**, **Dwells**, and **Orbit** panels. A Julia set image has these three panels and an additional one titled **Julia**.

At the bottom of the **Parameters…** dialog box are three buttons labeled **OK**, **Apply** and **Cancel**. The **OK** and **Cancel** buttons will both dismiss the dialog box. If **OK** is clicked, any parameters you changed will be applied to the frontmost fractal window; if **Cancel** is clicked the fractal will be unchanged.

If **Apply** is clicked, any parameters that were changed will be applied to the fractal, but the dialog box will not be dismissed. This allows you to see the effect of your changes and modify them without having to continually reopen the dialog box. This is a moveable modal dialog box, so if the box is obscuring the fractal you want to modify you can drag the dialog box to another part of the screen.

If you have clicked **Apply** and are satisfied with your changes, and you have not modified any parameters since the last time you clicked Apply, you may simply click **Cancel** to dismiss the dialog box. This prevents the fractal from being redrawn unnecessarily.

5.4.1.1 Area Panel

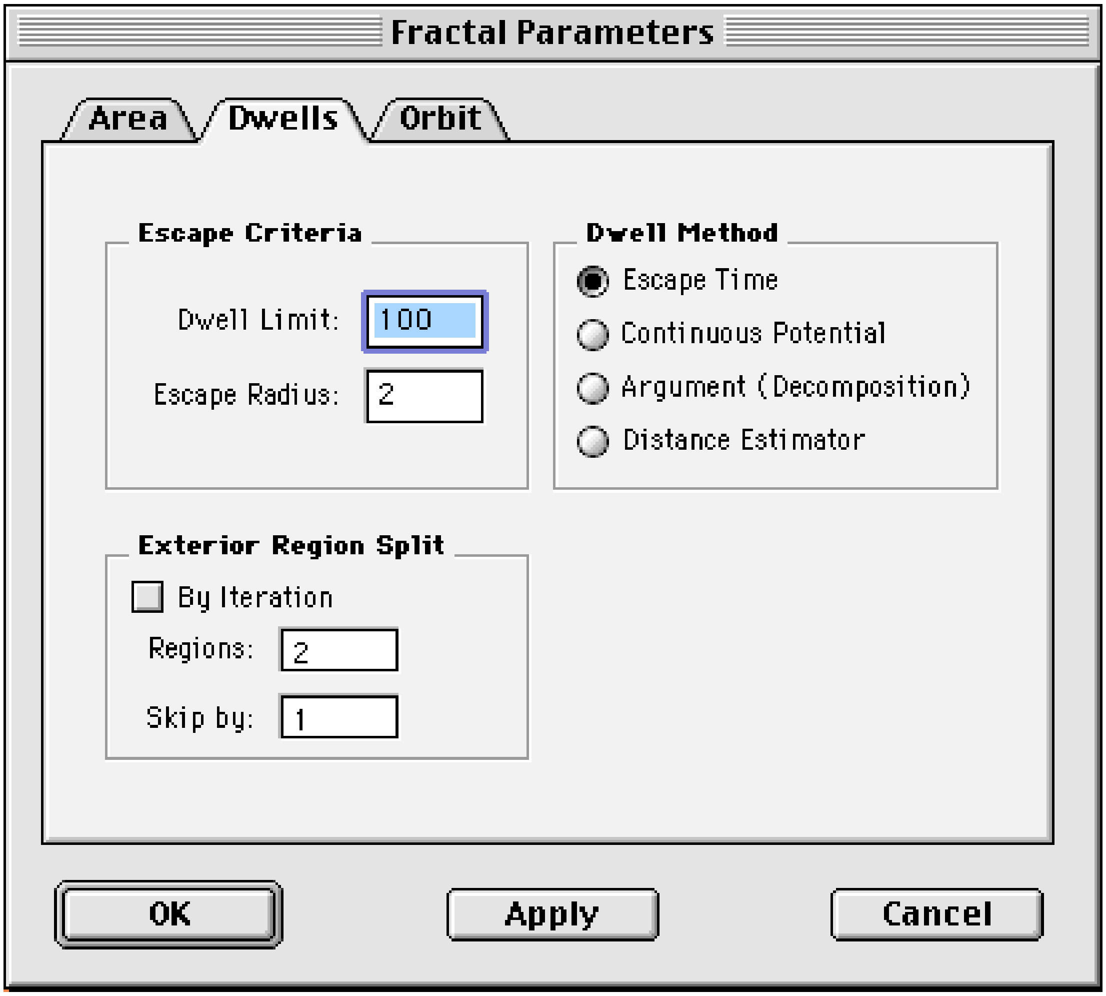


Pixels The size of the image in pixels is determined by the **h** and **v** fields. If either or both is changed, the image is resized if there is memory available for the new size.

Center Determines the center of the image in the complex plane.

Resolution Determines the total area covered by the image in the complex plane; the smaller this number is, the deeper the zoom.

5.4.1.2 Dwells Panel



Escape Criteria

Dwell Limit The total number of iterations allowed for each point in the fractal. If this limit is reached without exceeding the escape radius, the point is considered to be in the Mandelbrot set.

Escape Radius In the process of iterating the fractal formula, if the absolute value of the iterated value exceeds this number, iteration is halted and the number of iterations is recorded as the "dwell value" for the starting point of the iteration.

Exterior Region Split

By Iteration Checking this splits the Exterior region into several regions, based on the dwell value. This can produce a characteristic "zebra" effect. The regions alternate and each region can have its own color map.

Regions Determines how many regions the exterior is split into.

Skip by Only effective if "Split by Iteration" is checked. This causes the bands to be wider than one dwell. It is sometimes useful for zooms where the dwell values change so rapidly that simply splitting according to even or add dwells does not produce a coherent pattern. The “Skip by” value determines the width of the bands.

Dwell Method

The numerical value assigned to each point in the fractal image is called the "dwell" but in fact Fractal Domains provides several ways of computing the value at each point.

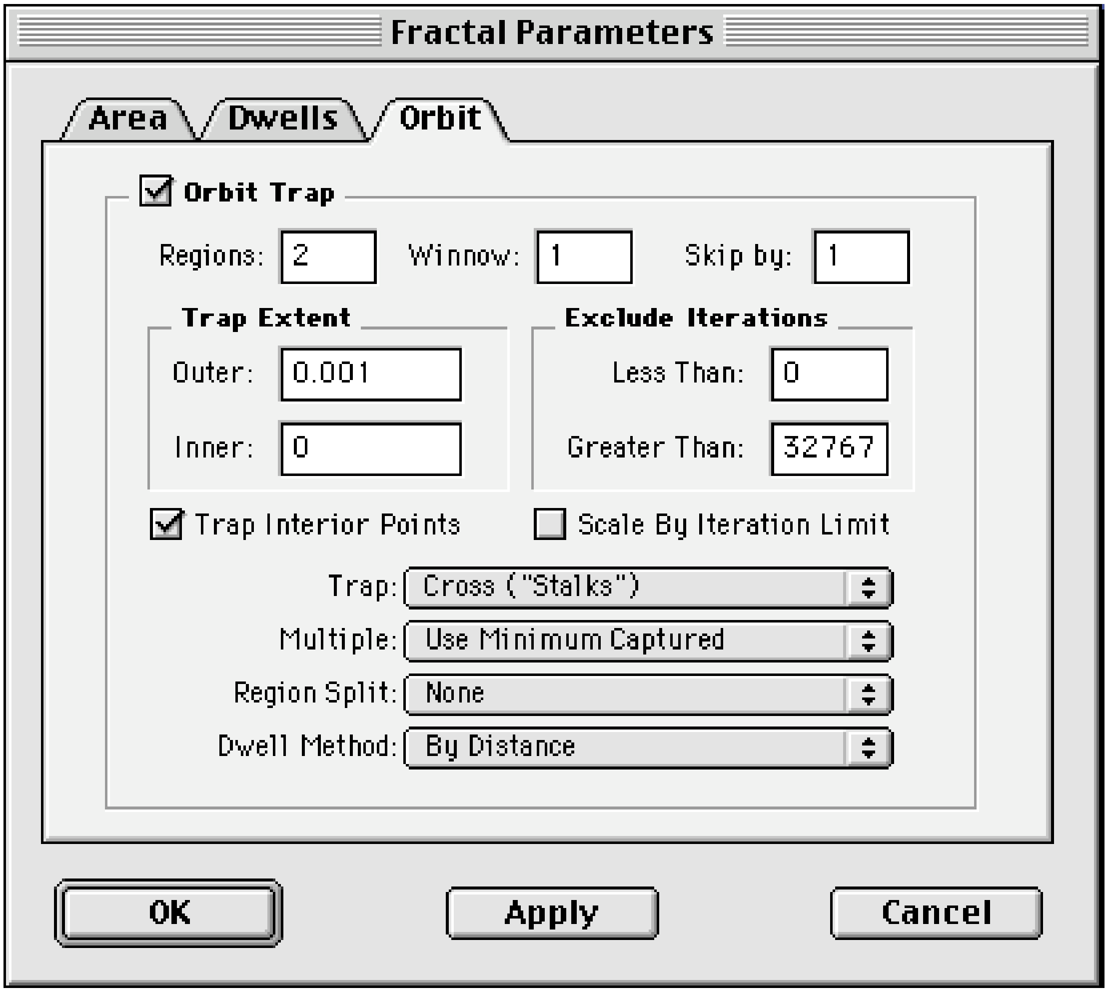
Escape Time This is the conventional way of assigning a value. The number of iterations required before the orbit exceeds the escape radius is called the Escape Time.

Continuous Potential This is a number given by a formula containing the number of iterations required to move past the escape radius and the actual position in the plane at that point. It is closely related to the escape time when the escape radius is very large (256 is a good value for the escape radius when using this method) but it is a continuous function of position and therefore yields smooth gradients rather than bands as the escape time does.

Argument (Decomposition) A number is assigned based on the angle the orbit point makes with the real axis (called the "argument" of that point) when it moves past the escape radius. Again, the results are sensitive to the exact value of the escape radius.

Distance Estimator This method uses a formula to estimate the distance of each point from the Mandelbrot set (the method can also be used for Julia sets). The closer the distance, the smaller the dwell value. With an appropriate color map, this method can be used to reveal the fine structure of the fractal which may not be apparent when using the escape time method.

5.4.1.3 Orbit Panel (Mandelbrot, Julia)



The Orbit panel is rather complicated, but it can be used to create a wide variety of effects. The contents of this window will possibly be different for fractal types supported by future versions of Fractal Domains. For the current version this is the window you will see for Mandelbrot and Julia fractals.

Orbit Trap Turn this on to use the orbit trap options. All other controls in this panel will be disabled unless this checkbox is on. Causes the Exterior region to be split into Exterior and Trap regions.

A point is considered to be part of a trap region if, during the iterative process, one of the intermediate points generated in the iteration process (the set of all these points is called the orbit) passes within a trap area centered at the origin of the complex plane. The shape and size of the trap area is defined by the controls in this panel.

Furthermore, the dwell value assigned to a point in the stalk region is not its actual dwell value, but a value based on the distance of its orbit from real or imaginary axis, in the case of the cross (stalks) or square traps, or the distance from the origin in the case of the circle trap.

Regions Only applies if **Split by Orbit** or **Split by Dwell** is selected. Determines the number of regions that the trap region is split into.

Winnow If this is set to N, only points whose orbits are trapped on an iteration divisible by N is counted as trapped. If set to one, all trapped orbits are counted. This allows fractals very dense in trap images to be reduced in complexity.

Skip by Only applies if **Split by Orbit** or **Split by Dwell** is selected. Increases the width of the bands generated by these option in the same manner as the **Skip by** parameter for the Exterior region.

Trap Extent

Outer and Inner Determines the inner and outer dimensions of the orbit trap. You may have to change this by many orders of magnitude to get a good picture depending on the orbit trap type and the fractal area itself. Circle and square traps call for much larger values than the cross and axis traps. Try entering different values and hitting **Apply** until you see results of the size you desire.

Setting **Inner** to a non-zero value creates a ring-shaped trap.

Exclude Iterations

Less Than and Greater Than The orbit of a point is tested for intersection with a trap only for points in the orbit with iteration values within these bounds. Points that intersect the trap at low iteration values tend to contribute to large features in the final fractal image, so if you want to remove the larger features, set the **Less Than** value to a higher number. Similarly, if you want to exclude very small features, set the **Greater Than** value to a smaller number.

Following are the rest of the Orbit trap panel items:

Trap Interior Points Subjects points interior to the Mandelbrot or Julia set to the same trap criterion, so that points in the set can be mapped to trap regions also. The default is for this to be on. It is here for compatibility with FracPPC, which always excluded set interior points from the trap region.

Scale by Iteration Limit If this is on, values in the trap region are scaled so that the maximum value is equal to the iteration limit. If this is off, the maximum value is close to 2^15. This option is off by default, it is available solely for compatibility with FracPPC, which always scaled values by the iteration limit.

Region Split Further splits the Trap region. This menu offers the following choices:

None Does not further split the Stalk region.

Split by Axis (Real/Imag) Splits trap region into separate regions based on whether the point where the orbit passed through the trap was closer to the real or the imaginary axis.

Split by Orbit Splits trap region into the number of iterations where the orbit passed through the

Split by Dwell Splits trap region into the actual dwell value is even or odd.

Trap Choose the shape of the orbit trap. Choices are:

Cross	Trap is a cross-shaped aperture. Orbit points are trapped if they land in an interval about the real or imaginary axis. Choosing this option creates the famous “stalks” discovered by Clifford Pickover.
Real Stalks	Like Cross, but only traps orbit points near the real axis.
Imaginary Stalks	Like Cross, but only traps orbit points near the imaginary axis.
Circle	Trap is a circular aperture. Orbit points are trapped if they land within a circle about the origin. This option creates circular or oval artifacts in the fractal plot.
Square	Trap is a square aperture. Orbit points are trapped if they land within a square centered at the origin. This option creates artifacts that look like distorted squares in the fractal plot.

Multiple

The orbit of a point in a fractal plot contains many points. It is possible that more than one point in an orbit may fall within an orbit trap. Often, many points in the orbit fall within the trap. The current version of Fractal Domains allows only one of these multiple trapped points to control the dwell value for that orbit, but the “multiple” popup menu enables the user to choose which of the multiple trapped points will be used according to three different criteria:

Use Minimum Captured	Out of all of the orbit points captured in the trap, use the one with the minimum distance from the origin to determine the dwell value.
Use First Captured	Out of all of the orbit points captured in the trap, use the one generated first during the iteration to determine the dwell value.
Use Last Captured	Out of all of the orbit points captured in the trap, use the one generated last during the iteration to determine the dwell value.

The **Multiple** choice usually has a dramatic impact on the appearance of the fractal image. Orbit trap fractals generally appear as a collection of distorted copies of the trap shape. The points captured early in the iteration process usually end up as members of the larger copies. Picking **Use First Captured** will produce a 3D effect in which the larger shapes are in front. **Use Last Captured** will place the larger copies in back. Use Minimum captured will run the shapes together.

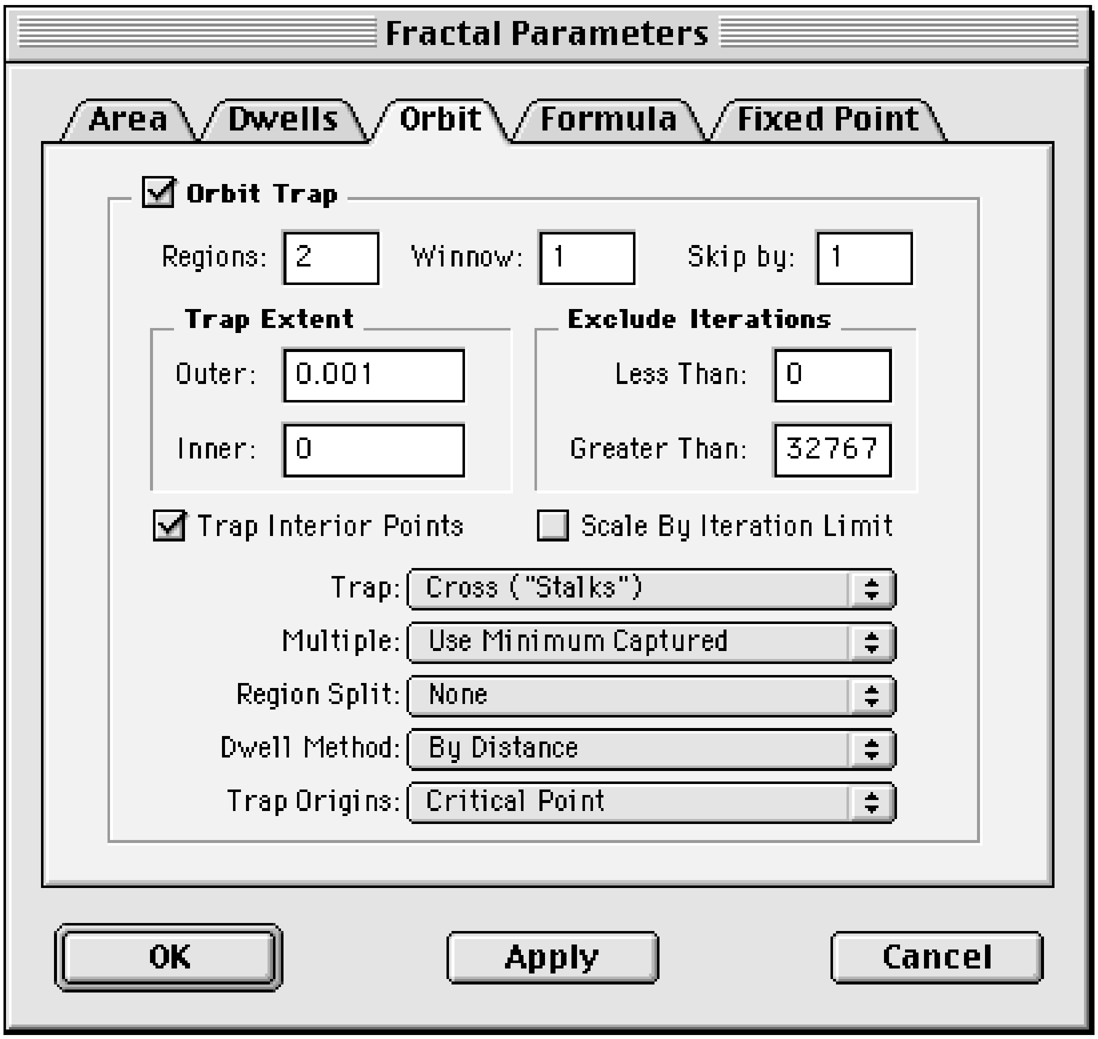
Dwell Method

Choose the method used to generate dwell values for points when the orbit falls within an orbit trap.

By Distance	The dwell value is proportional to the distance of the trapped point from the origin. The maximum dwell will be achieved when the trapped point is at the edge of the outer limit.
By Ratio	The dwell value is proportional to the ratio of the real and imaginary components of the trapped point. If you draw a line segment from the origin to the trapped point, the line will make a particular angle with the real axis. This angle is called the *argument* of the complex number. All points along that

line will have the same ratio of real to imaginary components. Thus, the ratio is essentially a function of the angle about the origin.

5.4.1.4 Orbit Panel (Rational, Newton, Halley)



The orbit panel for Rational, Newton and Halley fractals is nearly the same as the orbit panel for Mandelbrot and Julia fractals. Read the previous section (5.4.1.3) for documentation of most of the panel controls. This section only covers the aspects unique to the Rational-like fractals.

Region Split Further splits the Trap region. In addition to the menu choices listed in the previous section, this panel offers the following choice:

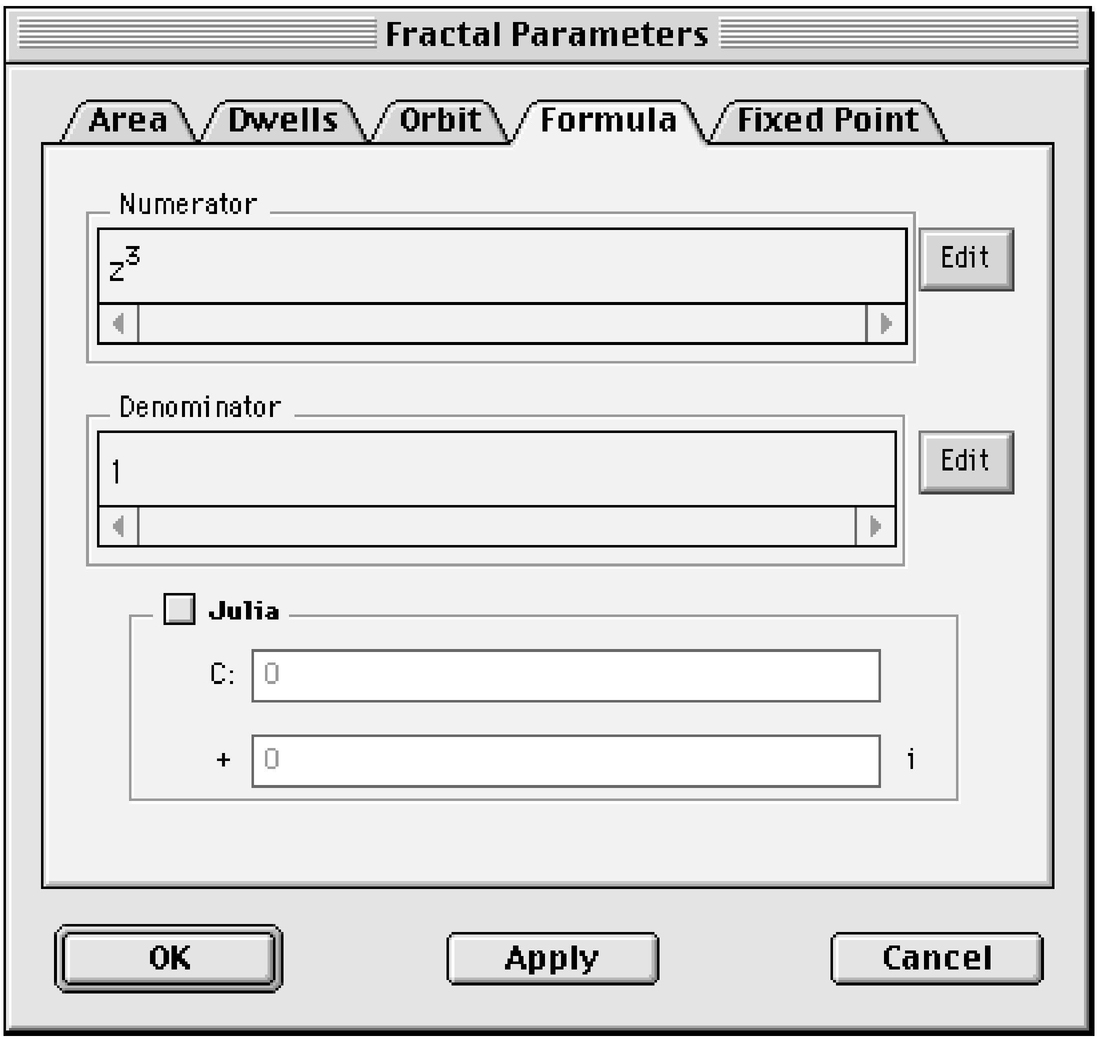
Split by Origin Splits trap region into separate regions depending on which critical point was the origin of the trap that intercepted ley fract.als is nearly the same as the orbit panel for Mandelbrot and Julia fractals.

Trap Origins Choose orbit trap origins. Choices are:

Zero	Places a single trap at the complex plane origin (0,0).
All Critical Points	Places an orbit trap at every critical point of the current function. Critical points are automatically computed by Fractal Domains. This option must be chosen if the user wishes to split the trap region by origin.
Critical Point #n	One of these entries appears for each critical point of the function. Choosing this places a single orbit trap centered at critical point #n.

If only one critical point exists, all critical point menu entries are replaced by a single entry called "Critical Point."

5.4.1.5 Formula Panel (Rational, Newton, Halley)



The formula panel is used to specify rational functions for use in the Rational, Newton and Halley fractal types. See section 3.2 for a discussion of rational functions.

Both Mandelbrot-like and Julia fractals can be generated. The general form of the iterative function specified by this panel is

$$R(z) + c \text{ where } R(z) = \frac{P(z)}{Q(z)} \quad P, Q \text{ polynomials}$$

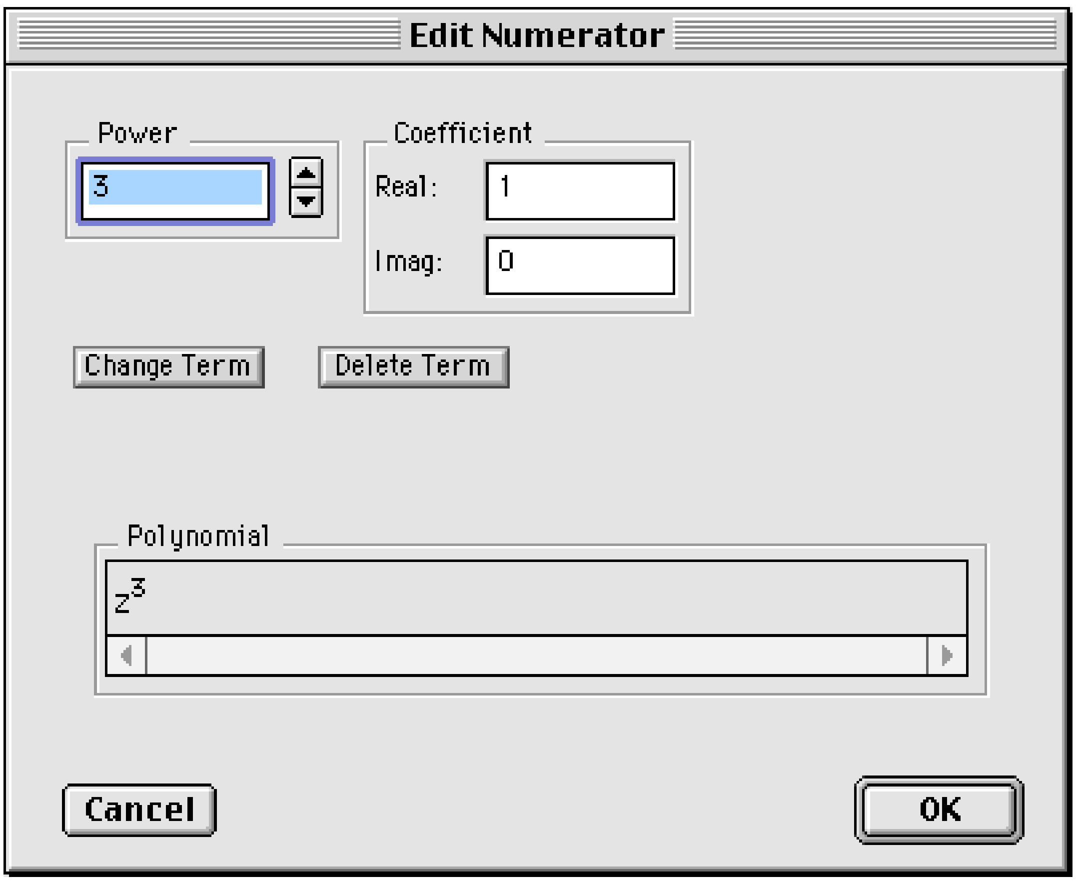
The polynomials P and Q are specified by editing the numerator and denominator of R. The Edit button next to each brings up a polynomial editor dialog box that can be used to create or modify each polynomial. After editing a polynomial, the new polynomial will be displayed in the **Formula** panel, but the fractal's rational function will not actually be changed until the user clicks the **OK** or **Apply** button.

The role of the c term is determined by the Julia check box.

If the Julia check box is checked, then the complex number specified in the "C" fields is used in the formula above; z ranges over the complex plane.

If the Julia box is unchecked, then the parameter c ranges over the complex plane. For each point, the formula is iterated with each of the critical points of R as a starting point. A point is considered to be "in the set" if the sequence is bounded for each critical point tried.

5.4.1.5.1 Polynomial Editor



This dialog box is used to create and edit polynomials. The current polynomial is displayed, and additions, changes and deletions are made term by term.

The term to be edited is chosen with the “Power” field which specifies the exponent of the variable in the term to be edited. The “Power” may be entered manually, or powers can be stepped through successively by clicking on the spin arrows to the right of the text field.

5.4.1.5.1.1 Adding Terms

If the number in the “Power” field does not correspond to the existing term, one button will display the label “Add Term” and the “Delete Term” button will be disabled.

Enter the desired coefficient in the coefficient real/imag fields and then click the “Add Term” button. The displayed polynomial will change to display the added term

5.4.1.5.1.2 Modifying Terms

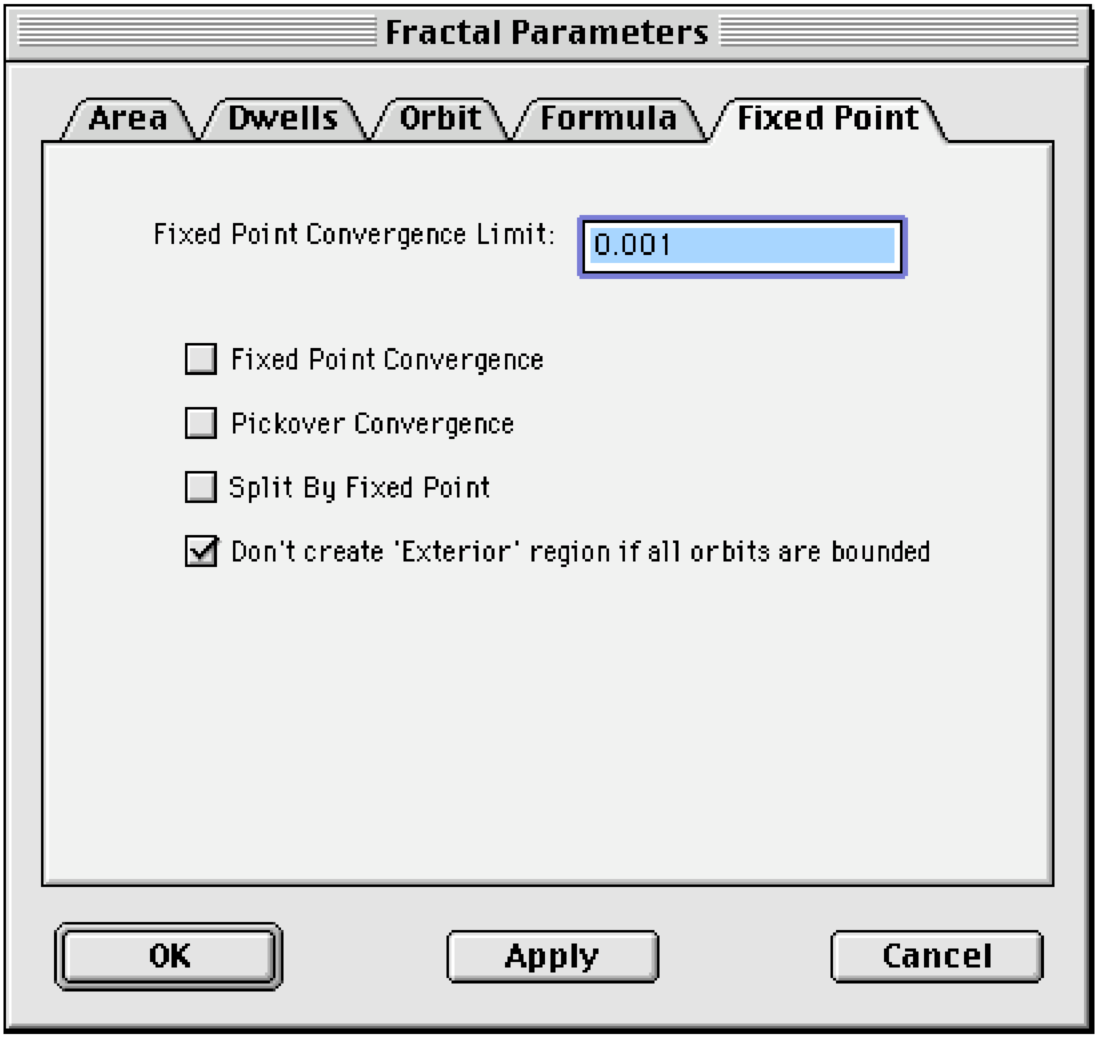
Enter the power of an existing term or scroll to the value with the spin arrows. When the power field displays an existing term’s exponent, the two buttons are both enabled and the labels are “Change Term” and “Delete Term.” The coefficient real/imag fields display the current coefficient of this term.

Enter the value of the new coefficient for this term and then click the “Change Term” button. The displayed polynomial immediately changes to reflect the changed term.

5.4.1.5.1.3 Deleting Terms

Bring up the power of an existing term as in the previous section. Click “Delete Term” and the term will immediately disappear from the displayed polynomial. Entering zero for the coefficient and clicking “Change Term” will produce the same result.

5.4.1.6 Fixed Point Panel



See section 2.5 for a discussion of attractive fixed points. This panel allows the user to control various aspects of the treatment of fixed points.

Fixed Point Convergence Limit — Determines how close together two successive iterations have to be to each other to be considered "converged" to a fixed point.

Fixed Point Convergence — If any attractive fixed points exist, checking this tells Fractal Domains to treat points that converge to a fixed point similarly to points that converge to infinity.

Pickover Convergence — Rather than determining convergence by measuring the distance between two complex numbers, the distance between their absolute values is measured. That is, rather than using this convergence criterion:

$$\left| z_{n+1} - z_n \right| < r$$

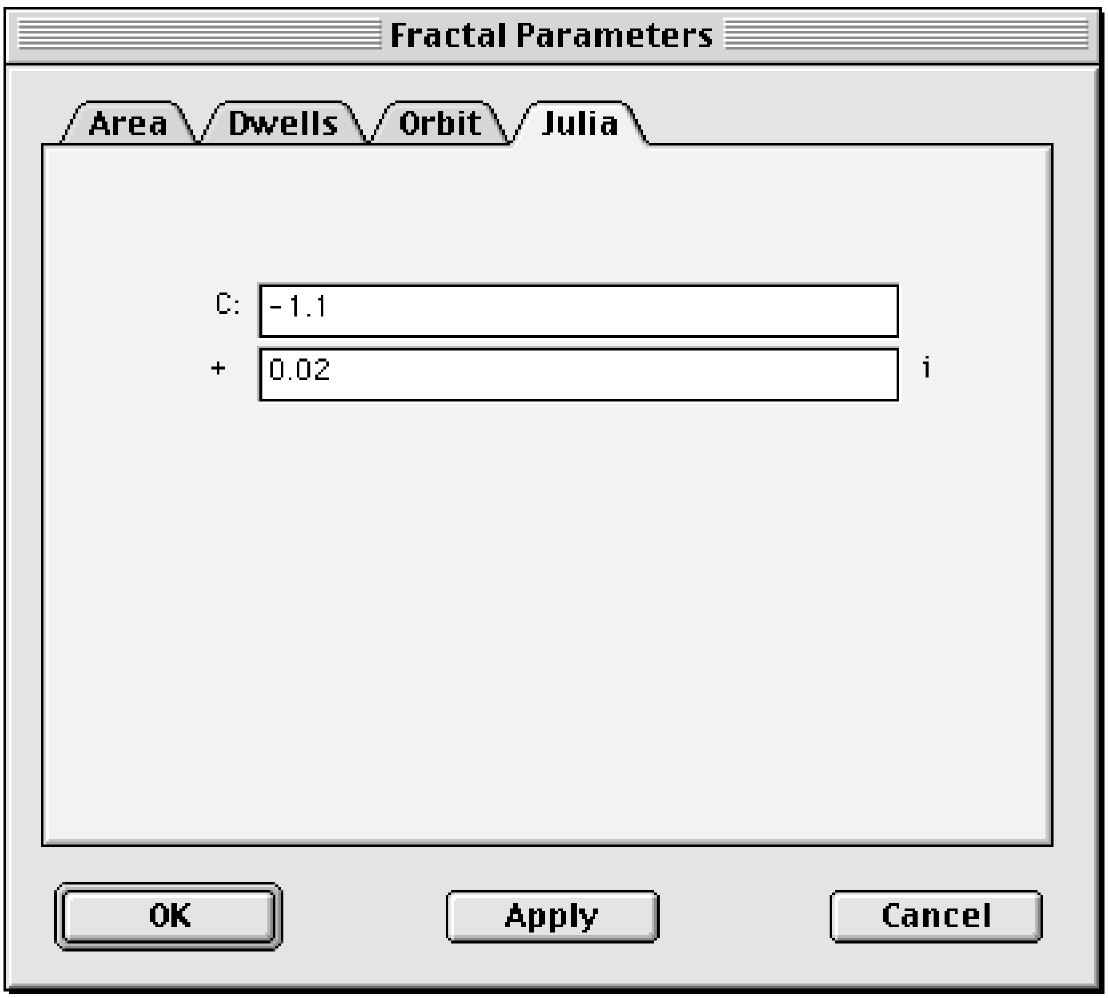
the following criterion is used

$$\left|z_{n+1}\right| - \left|z_n\right| < r$$

Split By Fixed Point Splits fixed point convergence region. The basin of convergence of every attractive fixed point gets its own region.

Don’t create ‘Exterior’ Region if all orbits are bounded If infinity is not an attractive fixed point, in principle all orbits are bounded. However, an exterior region is usually created anyway. If an orbit goes beyond the escape radius it is marked as belonging to the “exterior” even if the orbit may have fallen back below the escape radius. Thus, the existence of points belonging to the exterior is an artifact of a low escape radius. Checking this box suppresses the creation of an exterior region in this case. Turn it off if you like this artifact.

5.4.1.7 Julia Panel

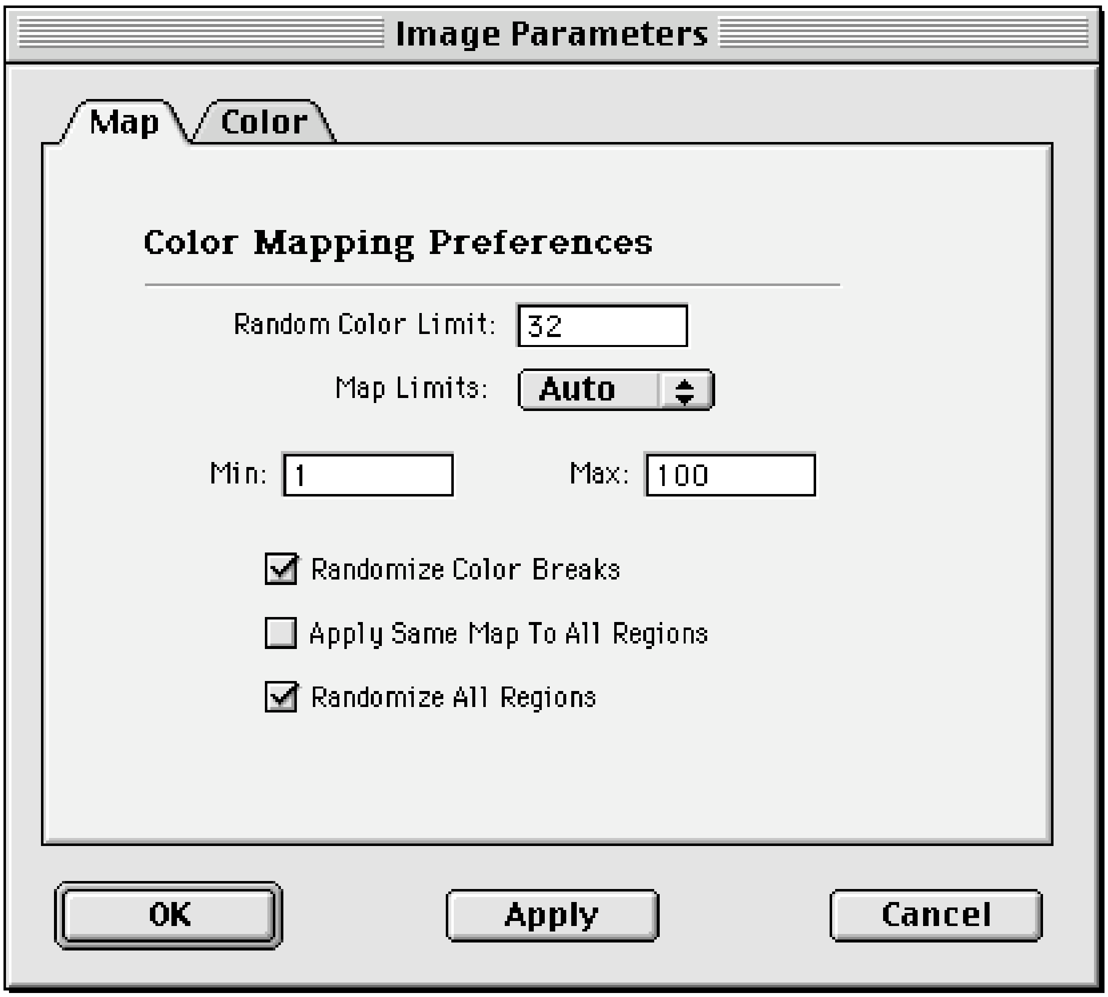


For Julia fractals, there is an additional tab panel labeled **Julia**. There are two fields to set in this window, for the real and imaginary parts of the Julia set constant C used in the iterative operation z^2 + C -> z that defines the Julia set.

5.4.2 Image Dialog

Choosing the menu item Image… from the Fractal menu brings up a tab dialog with two panels labeled **Map** and **Color**. It has **OK**, **Apply** and **Cancel** buttons that work as the buttons in the **Parameters…** dialog work. Consult section 5.4.1 for a description.

5.4.2.1 Map Panel



This panel controls the way in which color maps from the **Palette** menu or the Color Map Editor are applied to the regions of a fractal, and also controls how random color maps are generated.

Random Color Limit Determines the maximum number of color breaks in a random color map. The number of color breaks in a random map will vary randomly between two and this number.

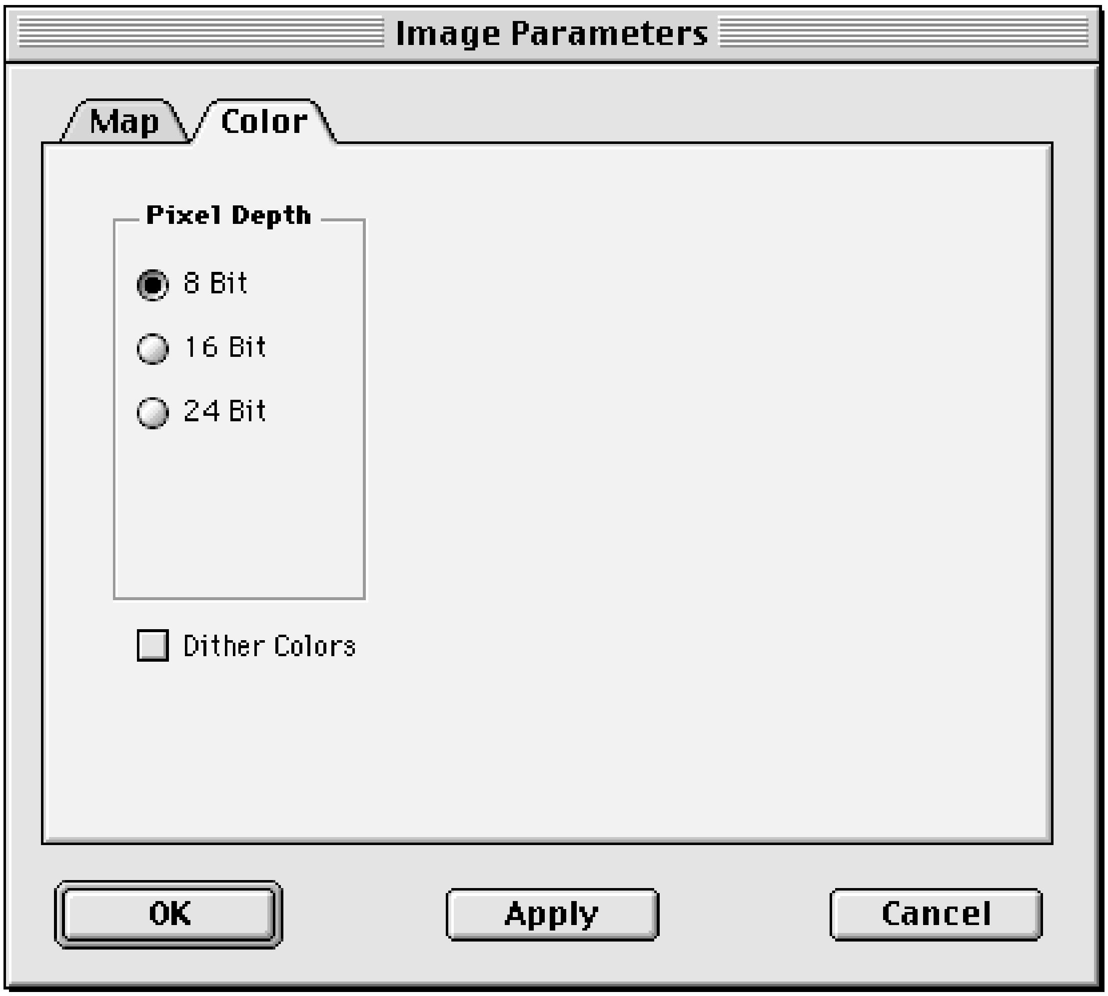
Map Limits This item is intended to control how the maximum and minimum dwell limits mapped vary with a zoom into the fractal.

Randomize Color Breaks If this is not checked, color breaks in a random color map will be spaced uniformly between the minimum and maximum dwell. When checked, the spacing in random color maps will also be randomized.

Apply Same Map To All Regions Causes a map applied to any region to be applied to all other regions automatically.

Randomize All Regions If not checked, random maps are generated only for the current regions (see the section on the color map editor to see how to change the current region). If checked, invoking the menu command **Randomize** generates a different random map for each existing region. However, this option is overridden by **Apply Same Map To All Regions**.

5.4.2.2 Color Panel

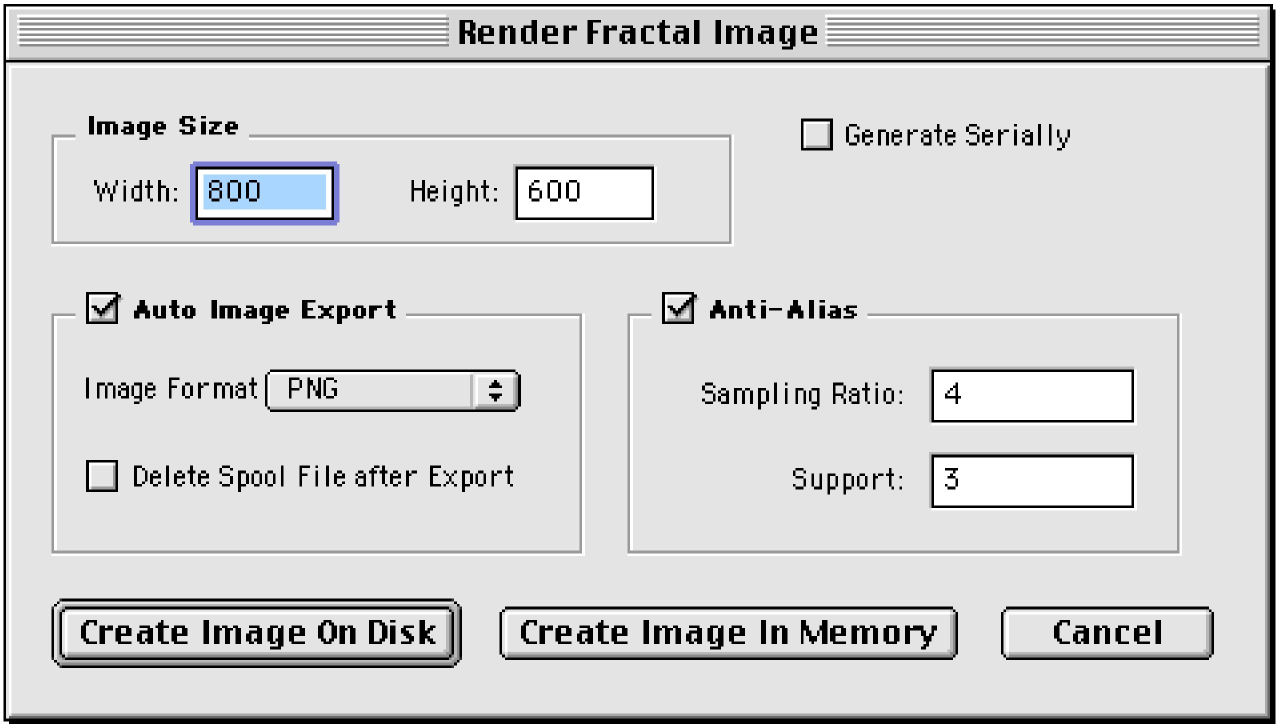


Macs with 24 bit displays will automatically generate 24 bit color fractals. This dialog box allows Macs with displays of lower depth generate higher depth fractals off screen. The resulting map will then be copied to the screen with optional dithering.

The value set in this window also affect the **Print…** and **Save As PICT…** commands. The picture sent to the printer or PICT file will be generated at the depth currently selected in this window. You can change the depth at any time, even after the dwells have been generated.

5.4.3 Render Image Dialog

Whenever a fractal editing window is visible on the screen, the **Render Image…** command is enabled in the **Fractal** menu. Selecting it brings up the following dialog box:



This dialog enables you to produce a higher-quality image of the fractal and/or spool large images to disk.

Image Size

These fields are initially set to the same values as the fractal window the image will be derived from, but you can set them to whatever you want. Generally this is used to generate an image that is much larger than the image in the fractal window.

Width Horizontal dimension of rendered image in pixels.

Height Vertical dimension of rendered image in pixels.

Anti-Alias

Turn this on to produce an anti-aliased version of your fractal image. Several samples are generated for each pixel in the image, and an anti-aliasing filter is applied. Currently, a Lanczos-windowed truncated sinc filter is used. Future versions of Fractal Domains may allow a choice of filters.

Filter Parameters

Sampling Ratio Determines how many samples (in both dimensions) should be generated for each pixel in the original image. This is important because it controls both the quality of the anti-aliasing and the amount of time required for generation.

Support This is used to determine where the sinc function is truncated. If you don't know what this means, you don't have to worry about it. Generally, this doesn't need to be changed.

It takes longer to generate an anti-aliased image than an image generated without anti-aliasing. The time factor involved depends mainly on the sampling ratio. The table below shows the approximate extra computation time:

If Sampling Ratio = 2, image takes four (4) times as long to generate.

If Sampling Ratio = 3, image takes nine (9) times as long to generate.

If Sampling Ratio = 4, image takes sixteen (16) times as long to generate.

Generally, sampling ratios above 4 don't yield noticeable improvement in quality. A sampling ratio of 3 is often a good compromise. Try all three values on some small fractal images and see if you think the improvement in quality is worth the extra time.

Auto Image Export

Turn this on to automatically export the image as a graphics file after the image has been completed.

Image Format Chooses which image format to automatically generate.

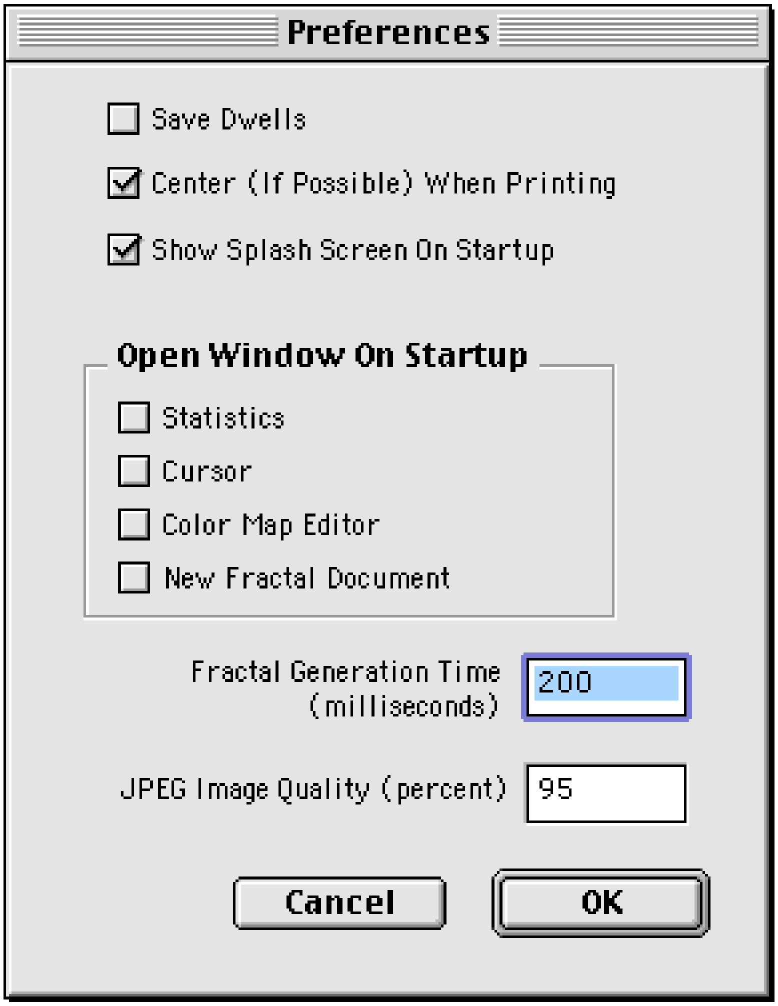
Delete Spool File If this option is chosen, then after the file has been successfully exported, the original spool file will be deleted...

Create Image On Disk If this button is pressed, the image will be spooled to a disk file with the image size and anti-alias options that you specified. You will be prompted for the name of the spool file. See section 2.6.2 for information on image spool files.

Create Image In Memory An image will be rendered in memory rather than on disk. A window will be opened that shows the generated image, but you will only be able to save as PICT and will not be able to resume a partially generated image once you close it. This is mainly useful for quickly checking the effects of anti-aliasing on small images.

Cancel Cancel **Render Image…** operation.

5.4.4 Preferences Dialog



Save Dwells

If checked, the dwell values are saved when a fractal is saved to a file, so they don't have to be regenerated when the file is reopened. Since this option makes the file very large and many fractals can be regenerated fairly quickly, you probably won't want to use this option very often.

Center (If Possible) When Printing

If the total area of your fractal image is greater than the printable area provided by the printer and Page Setup options you have selected, then this option has no effects; the fractal will be printed on multiple pages, as many as it takes to print the image.

However, if your fractal image is smaller than the printable area of one page, then checking this option instructs Fractal Domains to center the image on the page. If this option is not checked, the image will be printed justified to the top left corner of the printable area of the page.

Show Splash Screen On Startup

The default is for this option to be on. If you are annoyed by waiting for the splash screen to go away, you can uncheck this option.

Note: The ability to disabled this option is only available to registered uses of Fractal Domains. In addition, the splash screen stays up twice as long if the copy of Fractal Domains is unregistered.

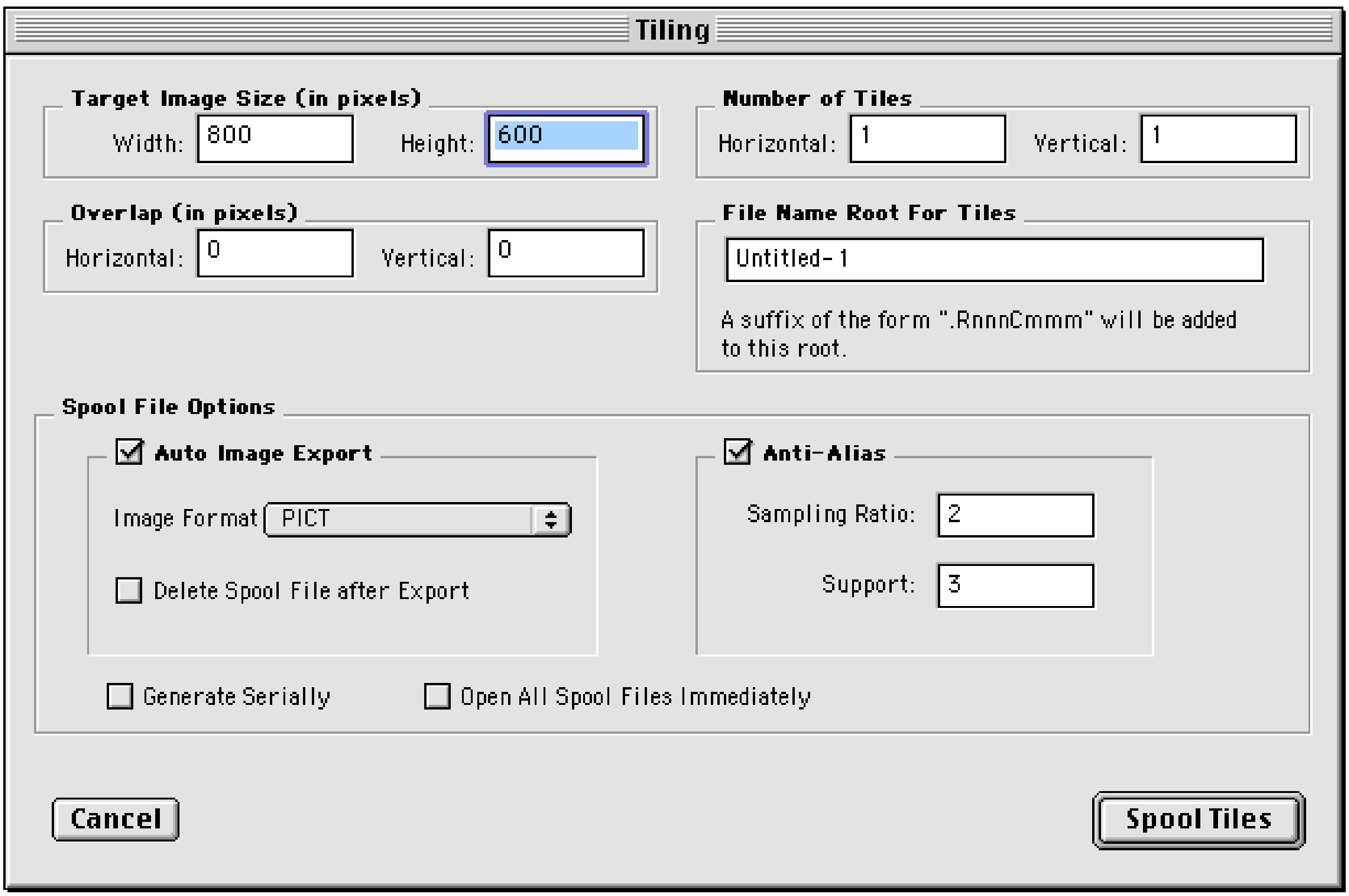
Open Window On Startup

These options allow you to determine which windows will be opened initially when Fractal Domains is started.

	Statistics	Statistics floating window is visible on startup.
	Cursor	Cursor floating window is visible on startup.
	Color Map Editor	Color Map Editor window is visible on startup.
	New Fractal Document	A new fractal is opened on startup.
Fractal Generation Time	The time in milliseconds that is spent calculating dwell values before allowing other process to have CPU time. Increasing this value speeds up fractal generation somewhat, although very large values will make the computer less responsive.	
JPEG Image Quality	Quality figure used when an image is exported as a JPEG file.	

5.4.5 Tiling Dialog

Whenever a fractal editing window is visible on the screen, the **Tiling…** command is enabled in the **Fractal** menu. Selecting it brings up the following dialog box:



This dialog box enables you to generate a series of spool files. The image in the editing window is divided into “tiles” and a spool file is generated for each file.

Target Image Size (in Pixels)
The width and height of the target image (the image created by fitting together all of the tiles).

Number of Tiles
The number of tiles in the horizontal and vertical direction. For instance, if the horizontal number is 3 and the vertical number is two, then there will be six tiles (three columns and two rows) and therefore six spool files created.

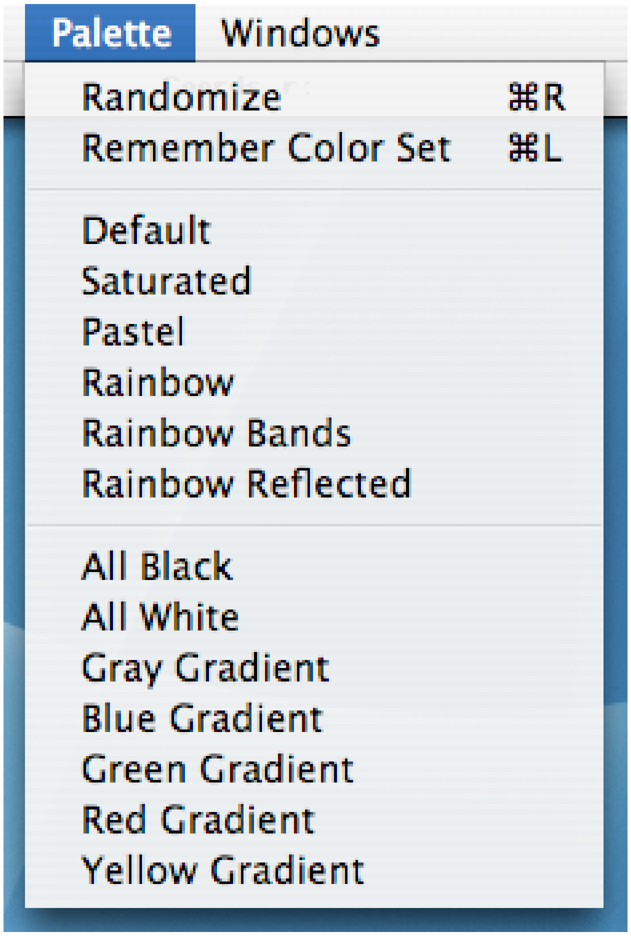
Overlap (in pixels)
The amount of overlap between adjacent tiles. For instance, if the horizontal overlap is zero, then the tiles don't overlap at all but concatenate right at the edge. If N pizels of horizontal overlap is specified, the left tile extends N pixels into the right tile's portion and the right tile extends N pixels into the left tile's territory.

File Name Root for Tiles
Every tile will have a spool file that begins with this root. A string will be appended with the row and column of the tile embedded in the string.

For instance, if the file name root is "aardvark," then the tile that is three across and two down will be generated with the name "aardvark.R002C003"

The rest of the options are the same as you would specify for the Render Image… command, and establish the default options when each of the tile spool files is created

5.5 Palette Menu



Commands chosen from this menu will replace a color map in the frontmost fractal window with a new color map. In the current version of Fractal Domains, this action is not undoable, so it is important to understand what effect the command will have.

If there is only one region in the fractal, the color map for that regions will be replaced. If there is more than one region in the fractal, then a command from this window will have the following action:

- If **Apply Same Map To All Regions** is checked in the **Map** panel of the **Image…** dialog, then color maps of all regions will be replaced with the chosen map.

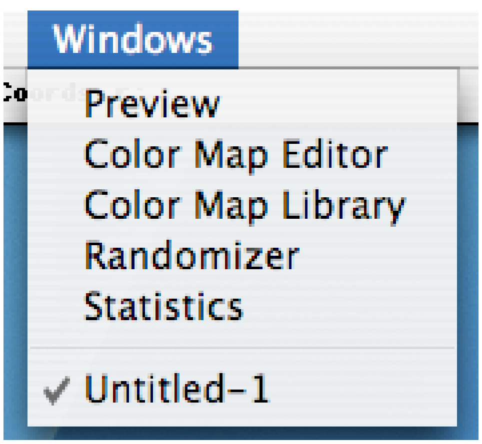
- If **Apply Same Map To All Regions** is not checked, then the color map of the region selected in the **Color Map Editor**. This is true even if the **Color Map Editor** window is not visible. If you aren't sure which region is currently selected, bring up the **Color Map Editor** to check before applying a new color map, as this action is not undoable.
- If **Randomize All Regions** is checked in the **Map** panel of the **Image…** dialog, then the Randomize command will generate a different random color map for every region in the fractal.

- If **Randomize All Regions** is not checked, then the Randomize command will generate a random color map for the region selected in the **Color Map Editor**. This is true even if the **Color Map Editor** window is not visible. If you aren't sure which region is currently selected, bring up the **Color Map Editor** to check before applying a new color map, as this action is not undoable.

After you apply a color map, you may edit it using the Color Map Editor. Following is a description of each Palette command.

Randomize	Generate a random color map and apply it to the fractal. See section 5.4.2.1 for an explanation of options that will influence the operation of this command. The keyboard shortcut Command-R can be used to execute this command. Often it is more convenient to use the keyboard shortcut than the menu when trying out different random maps on a particular fractal.
Default	Applies the default color map; this is the color map that comes up for a new fractal.
Saturated	A color map consisting of gradients between the pure primary colors red, blue, green.
Pastel	A color map with less saturated colors.
Rainbow	A color map consisting of a succession of gradients that start at black, ramp up to a color, and ramps down to black again. This is sometimes useful for various fractals when a continuous dwell method is selected.
Rainbow Bands	Like Rainbow, but with black bands in between.
Rainbow Reflected	Like Rainbow, but endpoints are at the peak of a band of color rather than at black.
All Black	Makes all pixels in the region black.
All White	Makes all pixels in the region white.
Gray Gradient	Color map is a two-color gradient, from black to white.
Blue Gradient	Color map is a two-color gradient, from light blue to dark blue.
Green Gradient	Color map is a two-color gradient, from light green to dark green.
Red Gradient	Color map is a two-color gradient, from light red to dark red.
Yellow Gradient	Color map is a two-color gradient, from light yellow to dark yellow.

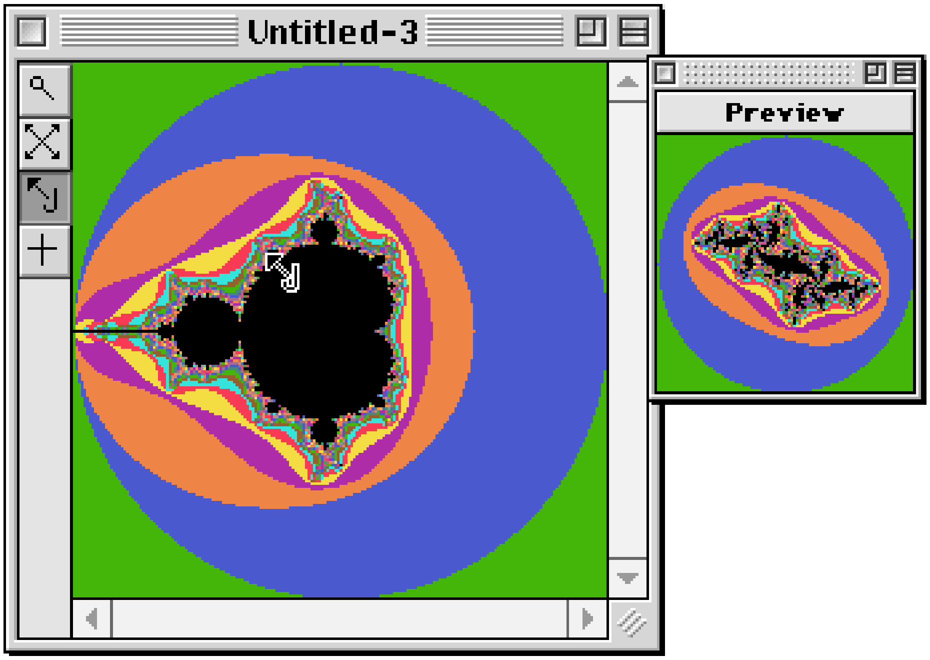
5.6 Windows Menu



There are four floating windows that show information about the fractal in the frontmost fractal window. Each window has an entry in the **Windows** menu. If the floating window is visible, the corresponding entry in the **Windows** menu is checked.

All open Fractal Domains documents are also listed in this menu (this includes fractal editing windows and spool windows). Any of these windows may be brought to the front by selecting it in the menu. The current frontmost window is checked in the menu.

5.6.1 Preview Window



The preview window can be used with any fractal that has the Julia tool enabled. As you move the cursor over the fractal window, the preview window shows a thumbnail sketch of the Julia set that would be created if you clicked the Julia tool on the point the cursor is passing over. The preview window is updated in real time. On the slower Macintoshes and/or with computation-intensive fractals, you may have to wait for the window to update to see the entire preview.

Any settings effective in the “master” window are applied to the preview image. This includes color maps, region splits, orbit trap settings, and dwell methods. This makes it an excellent tool for exploration, especially of rational fractals with various parameter settings.

The preview window initially shows an image that is 64 pixels square in size. You also have a choice of a 96 pixel or 128 pixel image for the preview window. Clicking on the preview window’s zoom box rotates through the three available image sizes.

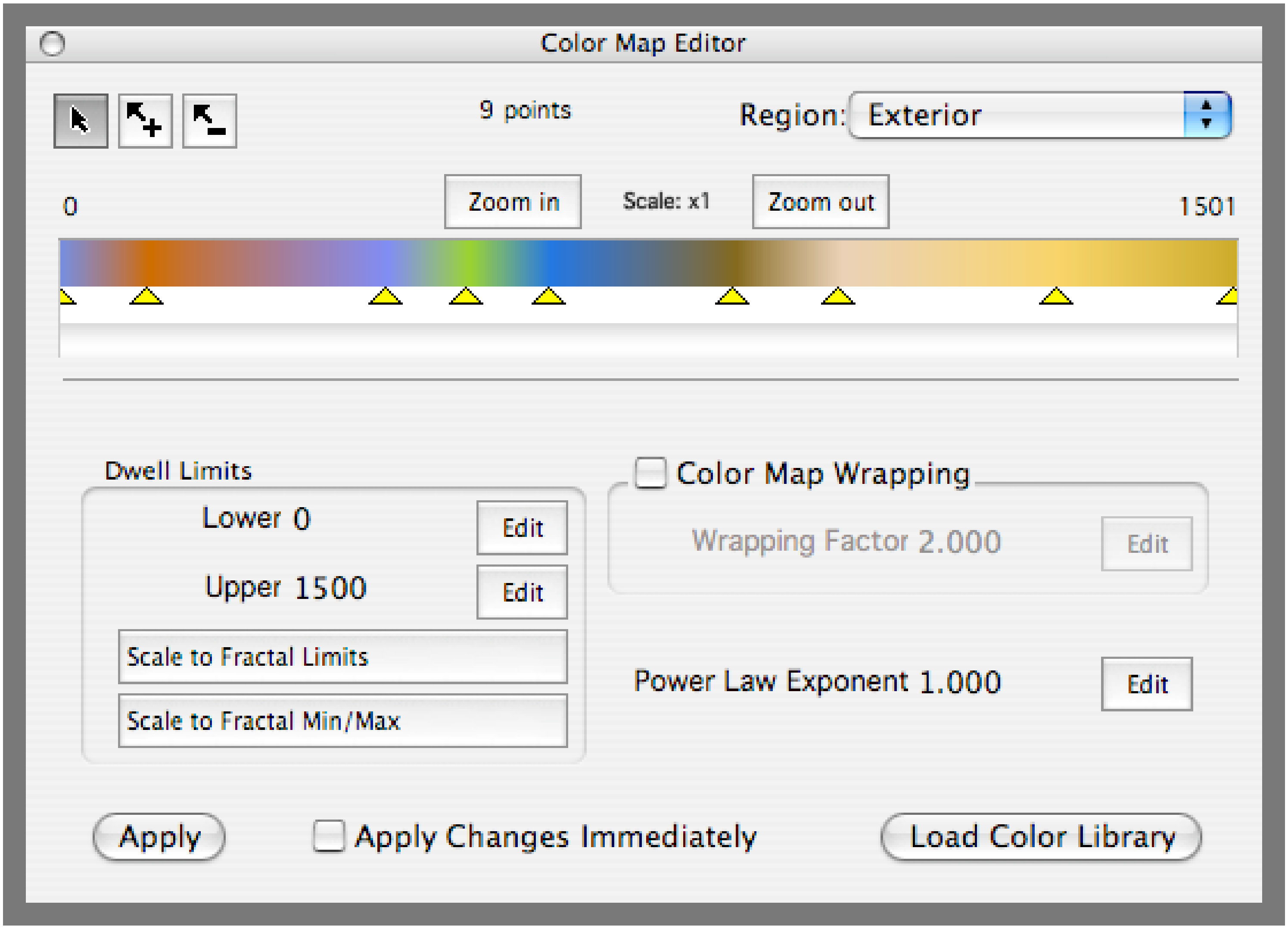
5.6.2 Color Map Editor

Before reading the following, you may want to review section 2.3 on color maps.

The Color Map Editor is used to modify the color map. Each region in the fractal image has a color map, which consists a sequence of color breaks at different dwell values. A color is associated with each color break. Between each adjacent pair of color breaks, the dwell values are mapped to a gradient of colors from the color of the lower color break to the color of the higher color break.

In the Color Map Editor, each color break is represented by a yellow triangle. At any time a single color break may be selected, the triangle turns blue to indicate it is the selected color break.

The picture below shows the appearance of the Color Map Editor when no color break is selected.



Cursor Selection The arrow cursor allows you to select a color break. The arrow with a "plus" sign allows you to add a new color break wherever you click on the dwell scale. The arrow with a "minus" sign allows you to delete a color break. After using the plus or minus cursor once, it reverts to the regular arrow (selection) cursor.

Num. of control points Displays the number of color breaks in the color map. Adding and deleting color breaks changes this number.

Region This popup menu selects the current region. This sets the region not only for the color map editor but also for the commands in the Palette menu.

Color Break Cursors These are the small yellow triangles; they allow you to set the dwell value and color of a color break. Drag a triangle to change the dwell value. Click on a triangle to select it and change the color associated with that color break.

Scale to Fractal Limits Adjusts the color map so that the first and last dwells of the color map correspond to the limits set for that fractal.

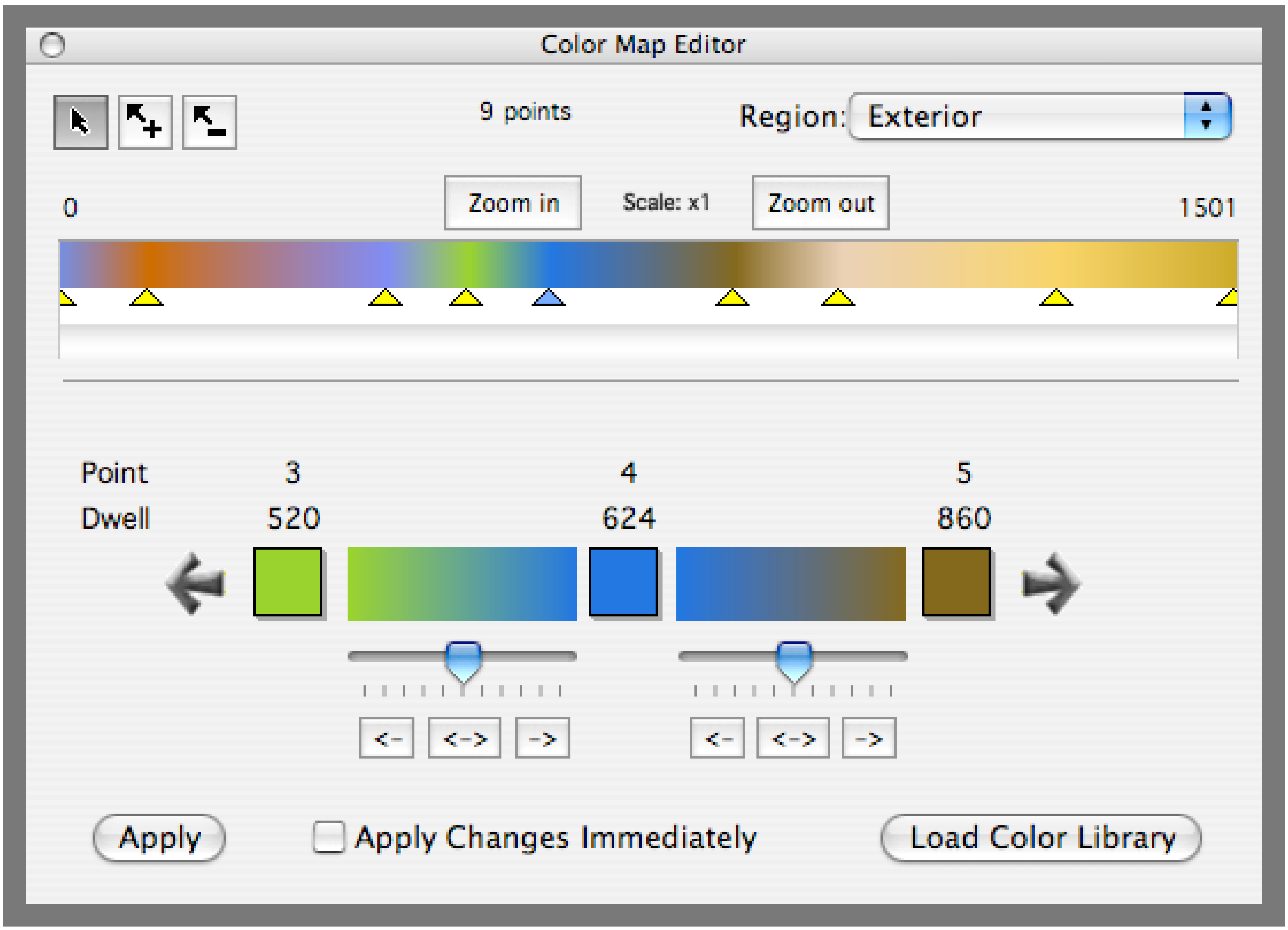
Scale to Fractal Min/Max Adjusts the color map so that the first and last dwells of the color map correspond to the minimum and maximum dwells that actually exist in the fractal.

Power Law Exponent Determines the current power law exponent for this color map. This can also be increased or decreased by pressing the right arrow and left arrow keys respectively. The mapping exponent can be reset to one by pressing the up arrow key. These keys have the same effect even if the color map editor window is not currently visible.

Apply Apply the changes made in the color map editor to the current region.

Apply Immediately When checked, any change in the color map editor is applied immediately to the fractal image.

When a color break is selected, the appearance of the window changes to show controls that apply to that color break and its immediate neighbors. See illustration below.



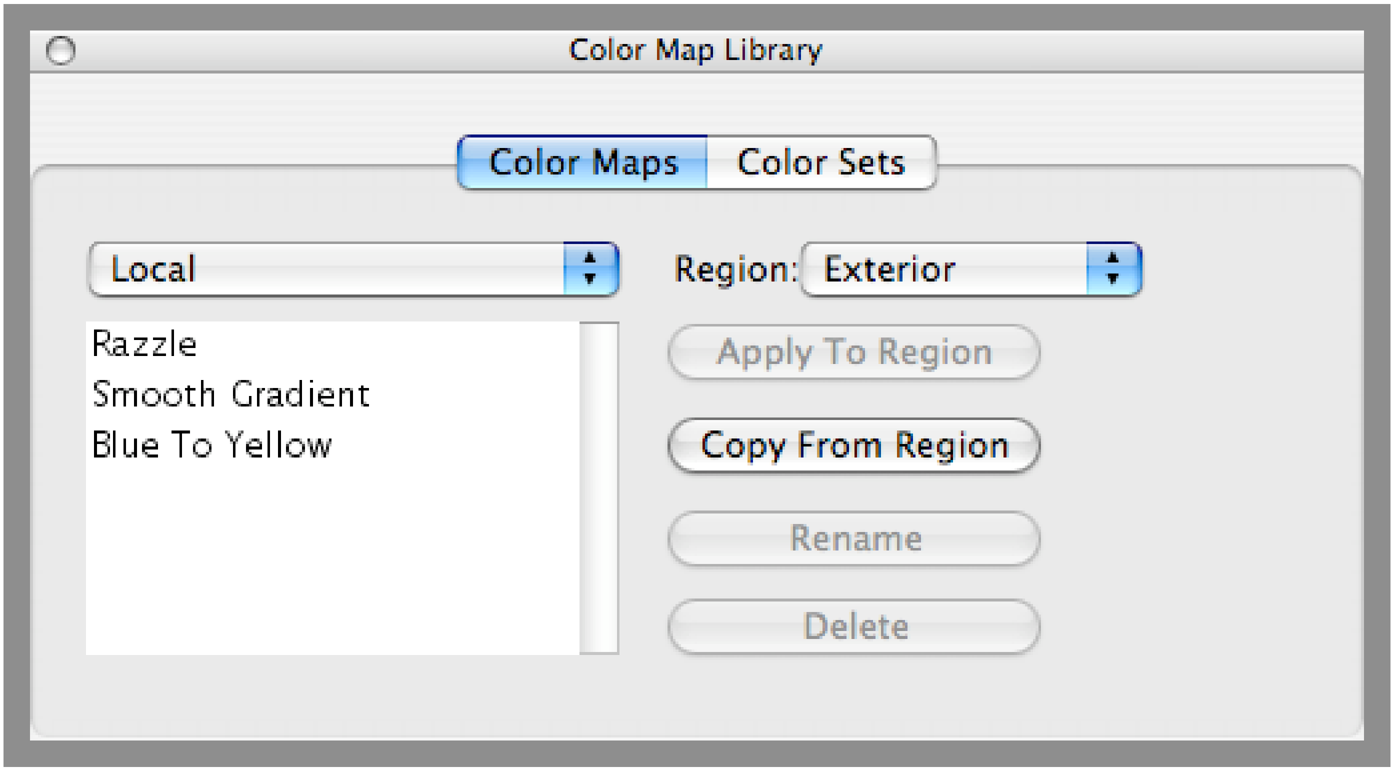
The blue triangle is the color break selected. In the lower half of the editor, the center square is the color of the dwell itself. You can click on the color square to change the color. The squares on the end at left and right show the color of the neighboring breaks; you can also change those if desired by clicking on the squares. The rectangles between the squares show the appearance of the gradient between the colors.

The arrows at either end allow you to scroll through the dwell points one at a time. Clicking on either arrow changes the dwell selected.

The slider controls underneath each gradient rectangle allow you to modify the rate of change of the gradient. Try sliding these back and forth to see the effect on the appearance of the gradient.

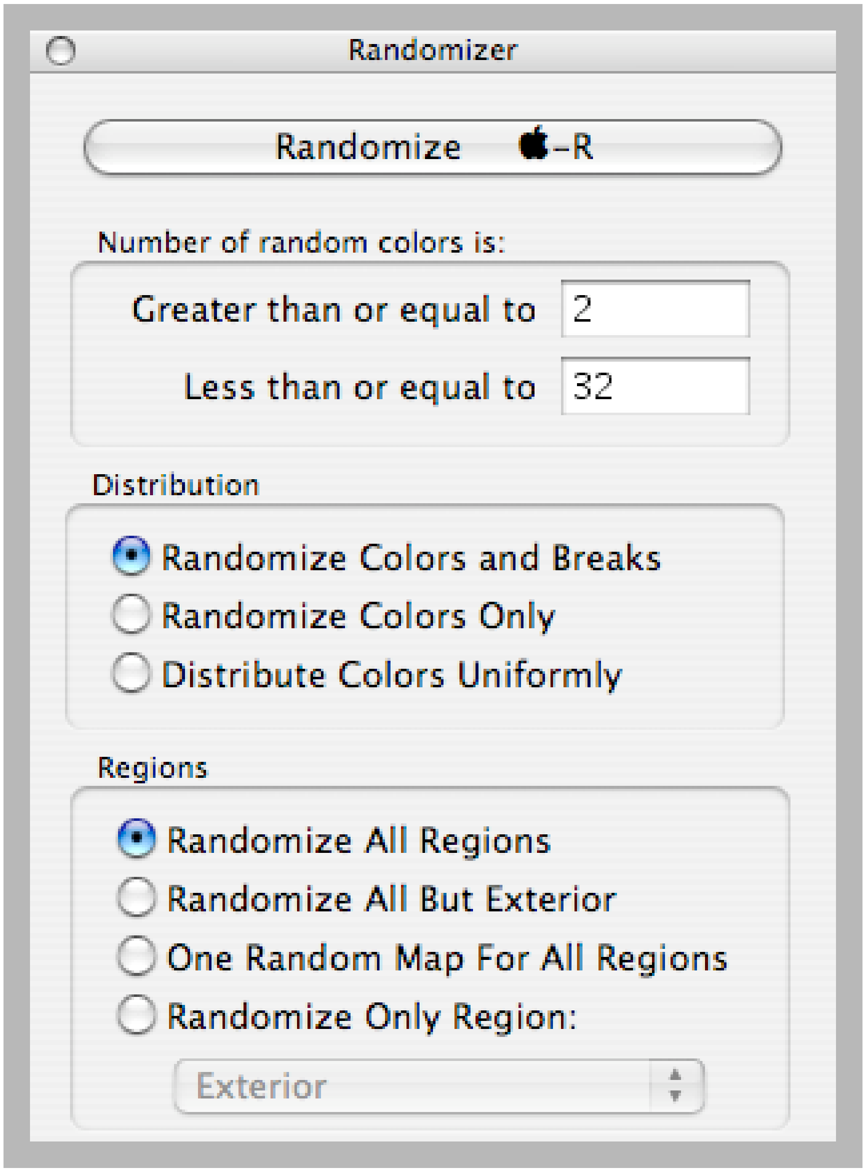
Underneath the sliders are some buttons that allow you to conveniently copy colors from one break to another. The “<->” buttons will swap colors between two color breaks. The other buttons will copy the color from one break to the other.

5.6.3 Color Map Library

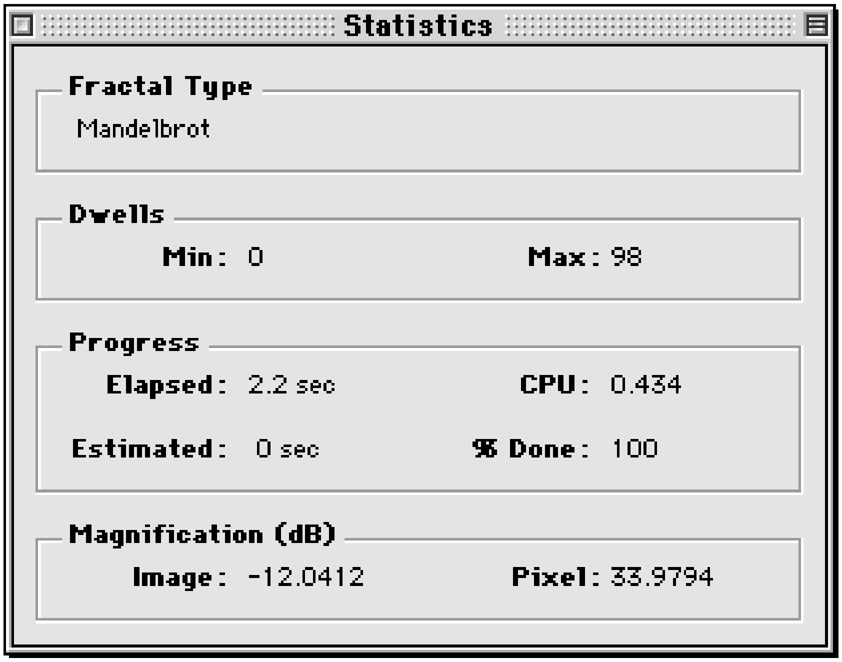


Documentation to be written.

5.6.4 Randomizer



5.6.5 Statistics Window



The Statistics window shows some facts about the fractal in the frontmost window:

Fractal Type
Type of fractal being generated. Currently the only possible choices are Mandelbrot and Julia.

Dwells

Min The minimum dwell value present in the fractal plot.

Max The maximum dwell value present in the fractal plot.

Progress

Shows progress information for the frontmost fractal.

Elapsed The amount of time that has elapsed since fractal generation started.

CPU The amount of time the computer has spent calculating the fractal. This is usually much less than the elapsed time, since the computer splits its time between fractal generation and other tasks. The elapsed time for a given fractal can vary greatly depending on what else the computer was called upon to do during the generation of the fractal, but the CPU time for a given fractal will be about the same every time it is generated.

Estimated Estimates time until fractal is finished, based on the amount of elapsed time spent on the fractal so far. You should wait a few seconds or so after starting a fractal for this number to settle to a realistic value. Even then, it should be taken as a very rough estimate, since there are many factors that can cause this to be inaccurate. Even so, it is very useful for giving an order-of-magnitude estimate for large or slow fractals.

% Done The percentage of the entire fractal that has been generated so far.

Magnification (dB)
As you zoom into a fractal, the magnification factor increases very rapidly. It is customary in engineering circles to use a logarithmic scale to express quantities that can vary over many orders of magnitude. One example of such a scale is the Richter scale used to gauge the intensity of earthquakes. Another example is the decibel, used to measure intensity of sound, electric signals and many other quantities. I decided to use decibels to display the magnification of a fractal plot. The formula I use gives the area in decibels (it's really dBArea, the ratio of the area to a unit area measured in the complex plane).

$$\text{Magnification}_{\text{dB}} = -10\log_{10}(\text{Area})$$

Image	Scale of the entire image (that is, the area of the image is plugged into the formula for magnification).
Pixel	Scale of a single pixel in the image (that is, the area of one pixel is plugged into the formula for magnification).

When zooming by a factor of two, the magnification in decibels increases by approximately six (6).

6. Registration

Fractal Domains is shareware. Some features are restricted and are not available until you register your copy. You may use the unregistered copy for an unlimited period to evaluate it. In order to have use of the restricted features, you must register. The current shareware fee is $20. Payment may be made electronically with credit card, or by mail with check, money order or credit card. Further instructions are in the "Read Me" file that accompanied the Fractal Domains distribution, or on the web at:

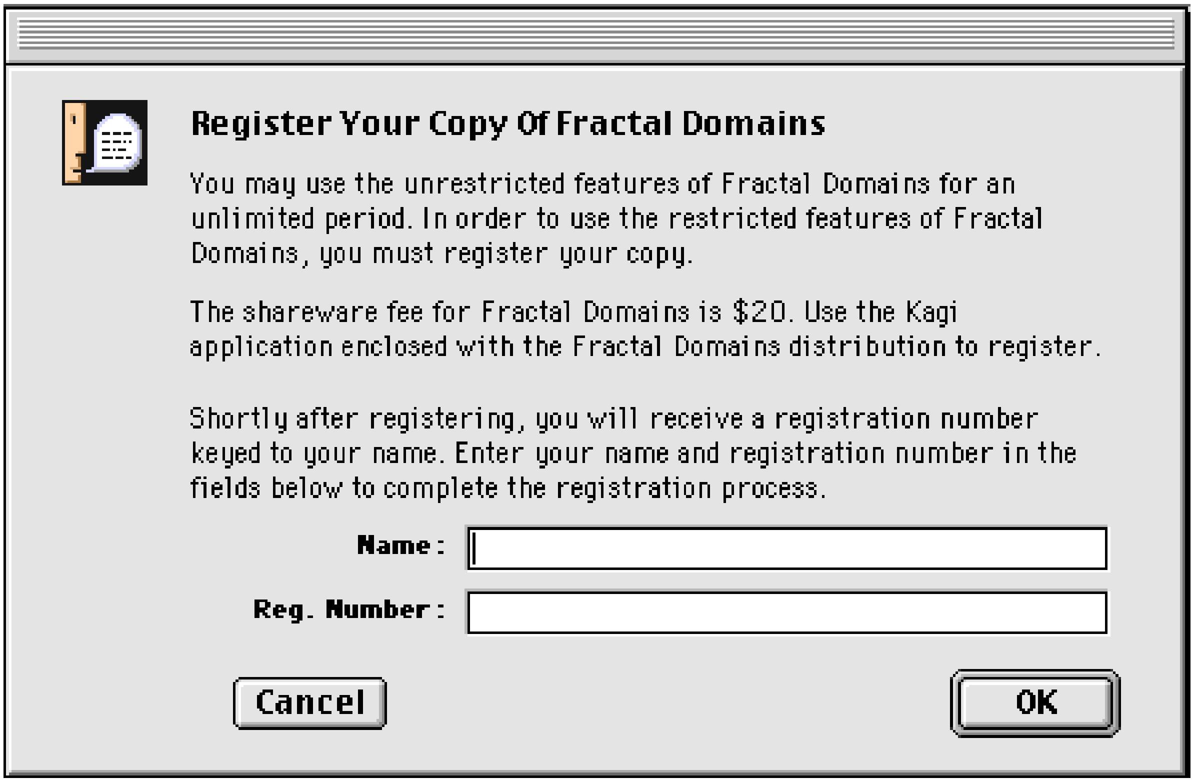
`http://www.fractaldomains.com/register.html`

6.1 Restricted features

An unregistered copy of Fractal Domains may generate all available fractals with all available dwell methods and orbit trap types. Saving and printing is restricted, however. You may save or print the parameter file or PICT file of any Mandelbrot or Julia fractal generated with the standard "Escape Time" dwell method and with either no orbit trap or the cross (stalk) orbit trap. Fractals generated with any other dwell method or orbit trap cannot be saved or printed. In addition, any image generated with the "Render Image…" command, whether generated in memory or spooled to disk, cannot be saved as a PICT file with an unregistered copy. However, if you generated a spool file with an unregistered copy of Fractal Domains, you can save the spool file and register Fractal Domains later. When your copy of Fractal Domains is registered, you can generate a PICT file from your saved spool file.

6.2 Entering your registration number

After you have paid the registration fee via the Kagi application, you will receive by email a registration number that is keyed to your name. Save a copy of this email in a safe place, in case you ever have to reregister your copy. In order to register your copy, start Fractal Domains and choose "About Fractal Domains…" from the Apple menu. There is a button on the "About" dialog box labeled "Register." Click this button. You will see the following dialog box:



Enter your name and registration number exactly as you received them and click OK. All previously restricted features of Fractal Domains will now be enabled:

6.3 Important note about registration

Your registration information is stored in a file that resides in the "Preferences" folder in your System Folder if you are running OS 9, or in your Library/Preferences folder if you are running OS X. If you move your copy of Fractal Domains to another computer, this preferences file will not follow it. Also, you may lose this file if you have a hard disk crash or if you do a clean install of the operating system. For these reasons, please make some sort of backup of your registration information. If you lose it for some reason, contact the author at the email address listed at the end of this document.

7. Fractal Domains On-line

7.1 Fractal Domains Web Site

The latest information can be found on the Fractal Domains web page, which is currently:

`http://www.fractaldomains.com`

This web page has pointers to many items related to Fractal Domains. You can find the latest updates and latest version of the program here. You can also browse through a gallery of images generated with Fractal Domains. Large format images suitable for use as desktop pictures or for a full screen slide show can be found from the download page. Finally, you will find a set of high quality links to other fractal web sites.

7.2 E-mail Addresses

The author's name is Dennis C. De Mars. You can always email me with any questions or bug reports via the following email addresses:

`support@fractaldomains.com`

7.3 Fractal Domains Forum

There is an online forum for discussing Fractal Domains with other users, submitting bug reports, and discussing fractals in general. Simply go to:

`http://www.fractaldomains.com/forums/`

8. Suggestions for further reading

Peitgen, H., Richter, P. (1986) *The Beauty of Fractals*. Springer: Berlin

Peitgen, H., Saupe, D. (1988) *The Science of Fractal Images*. Springer: New York

Peitgen, H., Jürgens, H., Saupe, D. (1992) *Chaos and Fractals: New Frontiers of Science*. Springer: New York

Pickover, C. (1990) *Computers, Pattern, Chaos and Beauty*. St. Martin’s: New York

Pickover, C. ed. (1996) *Fractal Horizons: The Future Use of Fractals*. St. Martin’s: New York