Instructions for Fractal Grapher developed by Jason Lisle, Ph.D.

Introduction

The Fractal Grapher is a Windows program that allows users to explore a wonderful aspect of God's creation that was unknown before the year 1978: the Mandelbrot set. This creation has been hidden in the abstract world of numbers from the beginning of creation. But it is now available to all who have access to even a very modest computer. Maps of certain sets defined by simple iteration turn out to be infinitely complex and remarkably beautiful when plotted in the Argand plane. Fractal Grapher allows you to explore these amazing shapes from the comfort of your computer. Since each of these fractals is infinite in complexity, you could spend a lifetime exploring its beauty without ever exhausting the depth of its riches.

What are Fractals?

Fractals are shapes that contain smaller versions of themselves, or of similar shapes. So, if you zoom-in on a small region of a fractal, you may find a shape that looks exactly like, or very similar to, the entire fractal. This property is called scale-invariance. Certain mathematical sets of numbers appear fractal when they are plotted in 2-dimensions. The first fractal discovered in this way is called the Mandelbrot set, after its discoverer Benoit Mandelbrot.

The Details

For those who have not yet read *Fractals: The Secret Code of Creation,* I offer here a very brief description of the Mandelbrot set, and how the software plots it. The Mandelbrot set is defined by iteration (iteration means doing a mathematical operation repeatedly). In this case, a number is selected, (let's call this number c). To find out if our selected number (c) belongs to the Mandelbrot set, we run it through a formula $z^2 + c$, where z is a *different* number that is initially zero. Adding these two terms results in a new number, which we declare to be the new value of z. We then take this new value of z and run it through the formula again (that is, we square z and add it to c), resulting in yet another value of z which might be larger or smaller (or the same) as the previous value. We repeat this process many times. If the value of z is always small (say, less than 10), then the number we selected (c) is part of the Mandelbrot set. If the sequence of z gets increasingly large without limit, then the number (c) is not part of the Mandelbrot set.

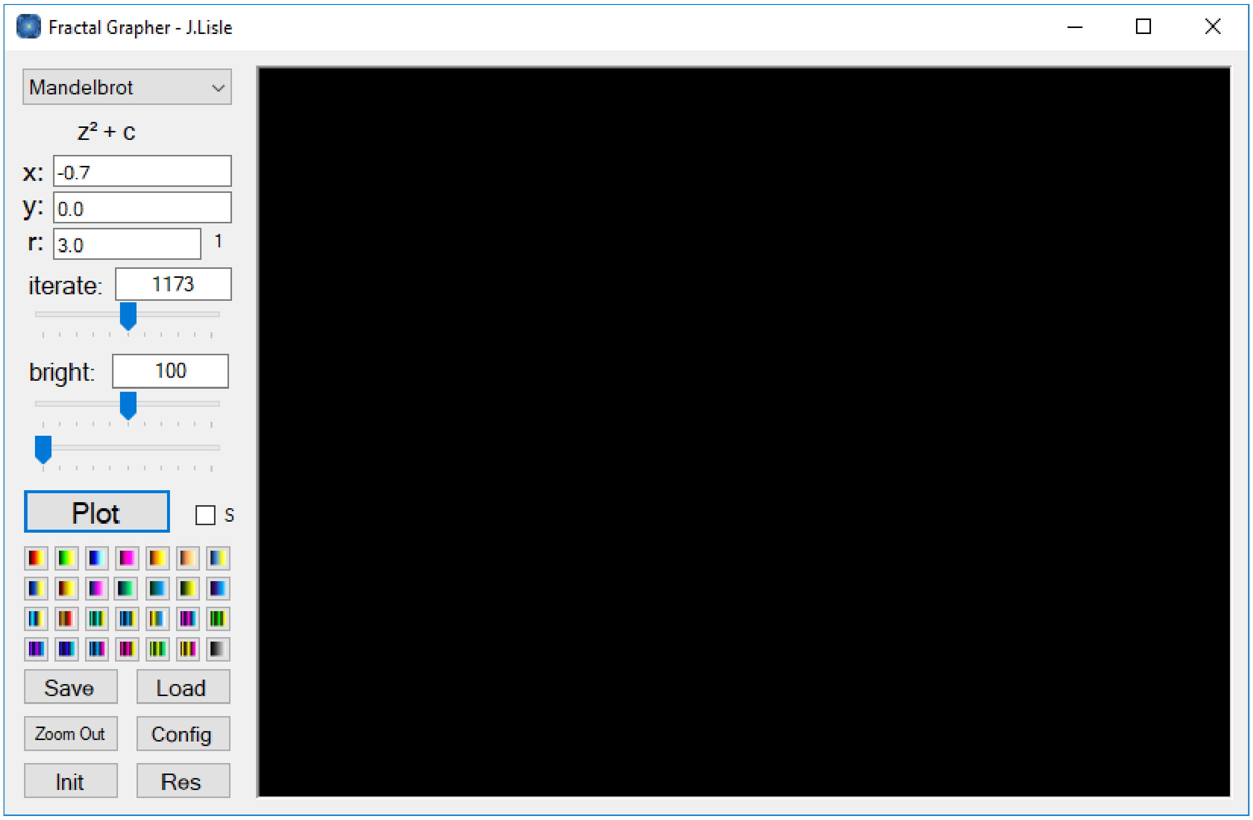
Note that the number "c" can be a simple number like, -2, 0, 0.23, etc. But it can also be a complex number like 3 + 2i where i is the square root of negative one: the so-called "imaginary number." (The term is misleading; "imaginary" numbers do indeed exist!) All complex numbers have a "real" part, like 3, and a so-called "imaginary" part, such as 2i. In some cases, one or both parts are zero. In principle, we can plot every complex number on an infinite 2-dimensional plane, with the x-coordinate representing the "real" part, and the y-coordinate representing the "imaginary" component. This is called the Argand plane.

Fractal Grapher systematically checks all points within a given range to see if they belong to the Mandelbrot set. It plots the result in the large central window, which represents a section of the Argand plane. The vertical axis represents the imaginary component, and the horizontal axis represents the real component of each number (c). By convention, points that belong to the set are colored black. Points that do not belong are given some other color that depends on how "far away" (how few iterations it takes to escape the bailout criterion) they are from being on the set. Points that are very far away are usually given a darker color, like red, whereas points that are closer are given a brighter color, such as yellow.

But you don't really have to know any of those details in order to appreciate the result. The program is doing all the math for you. You are free to explore the amazing shape that results without any further difficulties. Just keep in mind that what you are viewing is not the artwork of any human hand. It was built into numbers by the Creator of numbers.

Using the Program

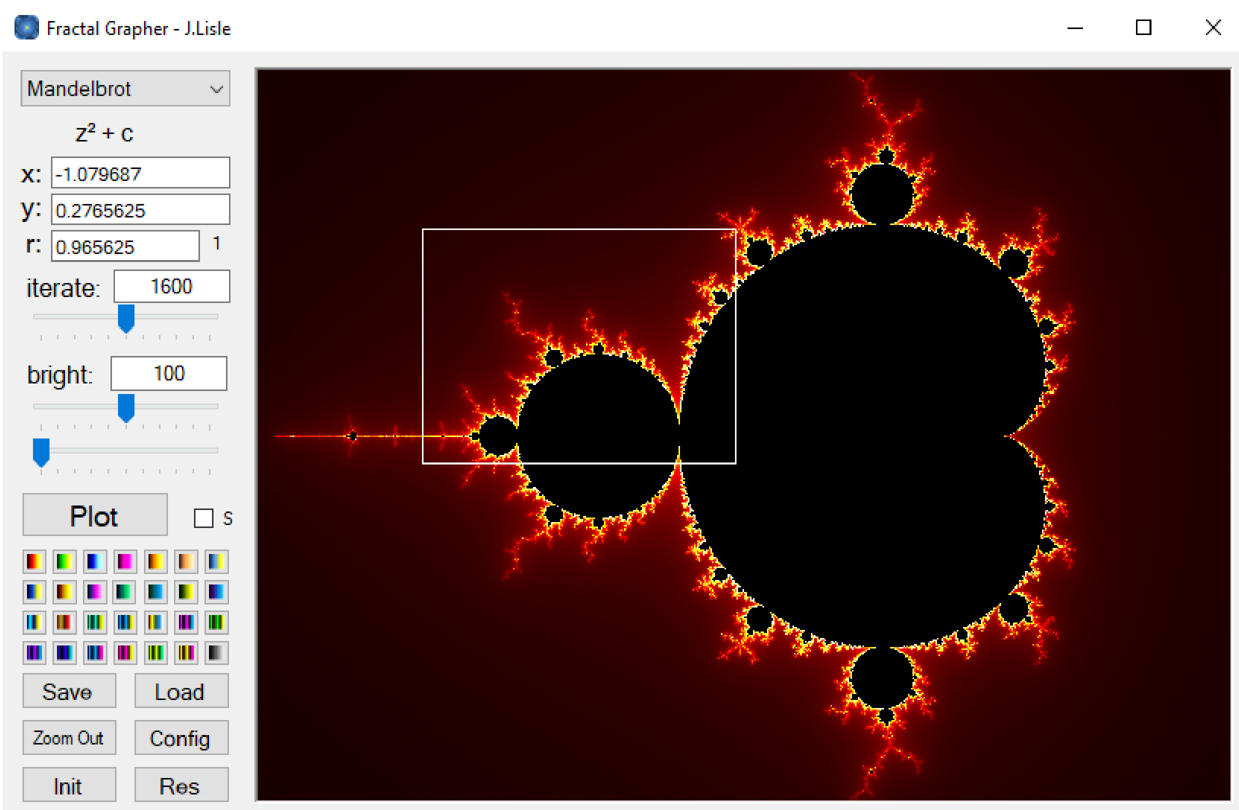
Double click on the "Fractal" icon  to start the program. (The program does not require any installation or setup). The large black area is the plotting window whereas the panel on the left displays information about the set to be plotted. Upon starting the program, all the parameters are set to plot the entire Mandelbrot set at default values. Go ahead and press the "Plot" button to plot the Mandelbrot set in the plotting window. The program will first quickly graph a low-resolution version, followed immediately by the full-resolution graph.



The top tab on the information panel displays the name of the set that is going to be plotted. Just below it is the iteration formula for that set. Let's stick with the Mandelbrot set for now. Below that are three boxes containing numbers that represent what part of the Argand plane is to be plotted in the window. In other words, these numbers specify exactly where in the Mandelbrot set we are, and how "zoomed-in" we are. The first two boxes (labeled "x" and "y" respectively) specify the complex number coordinates of the center of the plotting window, with x representing the "real" component, and y representing the "imaginary" component. By default, the window is centered at (-0.7 + 0i). The next box (labeled "r") indicates the horizontal range of the plotting window. Its default value is 3 because this allows us to plot the entire Mandelbrot set. As you zoom in on various regions of the set, this number will become very small. All these numbers indicate the area to be graphed when you press the "Plot" button. So, changing any of these numbers will not affect the current graph – only the next one to be plotted.

Navigating the Set

You can manually change any of the x, y, and r values to change the region to be plotted. But there is a much easier way to navigate the various regions of the Mandelbrot set. First, move the mouse over somewhere in the plotting window. Then press and hold the left mouse button, and while holding it, move the mouse to draw a box, then release the mouse button. This box represents the section of the Mandelbrot set that will fill the plotting window the next time you press "Plot." Notice that the x, y, and r values have changed, reflecting the coordinates of the new area to be plotted.



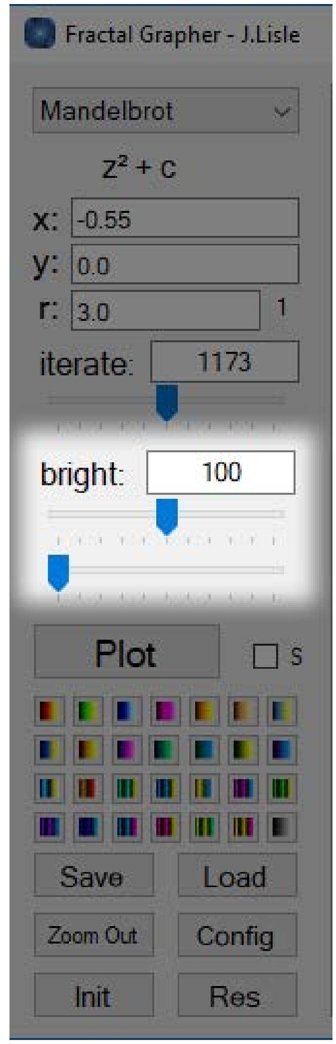
Once the box is drawn, you can easily move or resize it if you wish. To move it, left-click and hold anywhere inside the box, and move the mouse to change location. To change the size of the box, click and hold the *right* mouse button and move the mouse up or down. You can also use the mouse wheel to resize the box. Alternatively, you can scrap that box by drawing a new one with the mouse anywhere outside the original. Then press "Plot" to view the region you have boxed. (Alternatively, you can do a single left-click within the box to plot the new region). Using this method, you can easily zoom in on any region of the Mandelbrot set again and again.

As you zoom-in further and further, the images will take longer to plot. This is due to the increasing number of iterations and increasing precision required for accurate results. A low-resolution graph will always display first, while the program begins to compute the full-resolution plot. While graphing, the “Plot” button becomes a “Cancel” button, by which you can cancel the current plot at any time. The Cancel button may take a few seconds to shut down all the computation threads.

If at any point you get lost or want to abandon your current zoom, simply press the “Init” button on the lower left. This resets x, y, and r to the initial values and plots the entire Mandelbrot set. You can then begin a new zoom. Just above the “Init” button is the “zoom out” button, which zooms out by a factor of two in each dimension. This is useful since you cannot easily draw a box larger than the plotting window. Of course, you can always manually double the value in the “r” box to achieve the same effect.

That is really all you need to know to begin exploring the Mandelbrot set. What regions you choose to zoom in on are entirely up to you. However, I have found that the various “valleys” or “cusps” tend to contain some of the most beautiful shapes, particularly in their deeper regions. These include Seahorse Valley, Double Spiral Valley, Elephant Valley, and others. See the book *Fractals: The Secret Code of Creation* chapters for more information on these. But to get the most accurate and beautiful graphs, we now discuss some of the other options on the control panel.

Brightness, Contrast, and Colors



Immediately above the “Plot” button are two sliders. The upper slider controls the “brightness” of the resulting plot. Its default value is 100 (percent) which is displayed in the box above the slider. To see how this works, press “Init” to get the default plot of the entire Mandelbrot set, then move the slider to the left or right. This feature doesn’t simply brighten or dim the plot. Rather, it shifts the color spectrum. Recall, the colors are based on how many iterations are necessary to escape the bailout value (see the book for details). The brightness slider adjusts which color value matches a given iteration value. Since there is no conventional color scheme for points exterior to the Mandelbrot set, you can basically adjust this slider to whatever looks best.

You can also manually type in the brightness value in the box above the slider. The slider allows values between 25 and 400. However, you can use larger or smaller values by typing them directly into the brightness box.

Below the brightness slider is a contrast reducing slider. Moving it to the right reduces the range of the color spectrum. Generally, when you are viewing the entire Mandelbrot set, it looks best with this slider all the way to the left in its default position. If you move it to the right, you can see that it tends to wash out the colors. However, when you zoom in very deep to view extremely intricate structures, and are using more complex color schemes (discussed below), you may find that nudging this a bit to the right improves the view. In any case, both of these sliders are merely shifting the color scheme, and have no affect on the underlying math or shape. So, go with whatever looks best.

Just below the “Plot” button are 28 colorful squares. These select the color scheme for the plotted fractal. Go ahead and click on some of them to see how they affect the colors of the plot. The colors in the square give you an idea of what the colors in the plot will look like. In fact, they are exactly the colors used, but compressed into a small space. Colors on the left side of the square indicate colors used for points far away from the set (points that escape the bailout criterion quickly). Colors on the right side of the square are brighter and indicate points very close to the set, always culminating in white.

The upper two rows of colors represent either a simple single-color gradient or a dual-color gradient. A single-color gradient scheme starts with one dark color (such as blue) and gradually brightens it culminating in white. The default scheme (upper left square) is a dual-color gradient; dark red brightens and shifts to yellow, which then brightens to white. These simple single and dual themes are best for understanding the true shape of the Mandelbrot set because lighter colored points are *always* closer (in terms of more iterations) to the set than darker colored points.

The third and fourth color rows contain more complex color schemes. These tend to result in prettier maps, particularly for deep zoom regions of high complexity. When using the more complex color schemes, it may take some fiddling with the brightness and contrast sliders to produce the most attractive results. The final (lower right) square applies a simple grayscale scheme unless replaced with a user-defined color scheme. More on that below.

Just to the right of the Plot button is a small checkbox labeled "s." Clicking that box will toggle "stripes mode." This mode highlights subtle differences in the blander regions of the plot by using a slightly different color scheme for iteration escape numbers that are even rather than odd. The effect can be particularly striking in valleys and within spirals. On the other hand, it can be a bit too much on certain plots. Experiment.

Unlike the vast majority of fractal plotting software, Fractal Grapher uses a dynamic color range, so that points very near the perimeter of the Mandelbrot set are always brightly colored, and points far away are always darker. This ensures that each plot looks crisp and vibrant no matter what the zoom level.

Saving Images

It's easy to save any image you plot. Simply click on the "Save" button just below the color squares to bring up the *Save Menu*. You specify the filename.

You have several options in terms of image file formats. The default option is the Portable Network Graphics (*.png) format which is a versatile and widely used format, suitable for almost all purposes. The *.png format is lossless, meaning the file image is an exact replica of the original. Yet this format has good (lossless) compression, meaning that the resulting file doesn't take up a lot of disk space. Bitmaps (*.bmp) are also lossless (thereby perfectly preserving your image), but they have no compression, and therefore take up a lot of disk space.

You can also save as a Jpeg (*.jpg) which is one of the most common formats used on the internet. This format has very good compression and therefore takes up little disk space. But it is lossy – meaning the resulting image will be an approximation of the original, and not an exact replica. It may look good enough to the eye, but zooming in may reveal some undesirable blemishes. The *.tiff and *.gif formats are also available.

Loading Images (and continuing where you left off)!

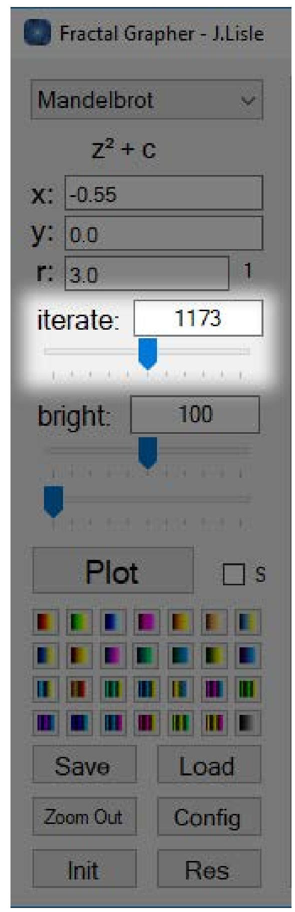
One of the neatest features of the Fractal Grapher is its ability to load previously saved images and continue where it left off. So, when you save an image, you don't need to remember what equation you were plotting, what the coordinates were, or what color and brightness settings you used. All that information is stored in the image file. When you load the image file, Fractal Grapher automatically sets all the parameters to what they were when you first plotted that image. You can pick up right where you left off.

This means that you can continue to zoom in (or out) on and explore any images you have saved. You can either use the "Load" button to load in an image, or you can simply drag-and-drop the image file anywhere onto the Fractal Grapher interface. You may then immediately zoom-in on any section or zoom out. Alternatively, you can change other settings, such as brightness and color schemes; *however, for these other settings to take effect you must first **replot** the image by pressing the "Plot" button.*

The ability of Fractal Grapher to pick up where it left off has two limitations. First, it only works on unaltered files. If you use third-party software like Photoshop to alter the image, and then save over the original file, the plotting parameters will be lost and Fractal Grapher will not be able to pick up where it left off. It will still display the modified image, but the coordinates will be lost, and therefore you cannot replot the image. Consequently, if you modify an image with third party software, I recommend saving the modification under a new name, keeping the original file in case you want to come back and continue exploring it.

Second, bitmap (.bmp) images work a bit differently. The equation parameters cannot be directly stored in such files. Therefore, when you save a file in the *.bmp format, Fractal Grapher also creates another file with the same name but with the extension ".meta." This file contains the graphing parameters. If you later load the *.bmp file, Fractal Grapher searches for the *.meta file with the same name and loads the parameters from it. So, if you delete the .meta file, then you will not be able to continue from the bitmap image.

Iterations: Speed and Accuracy



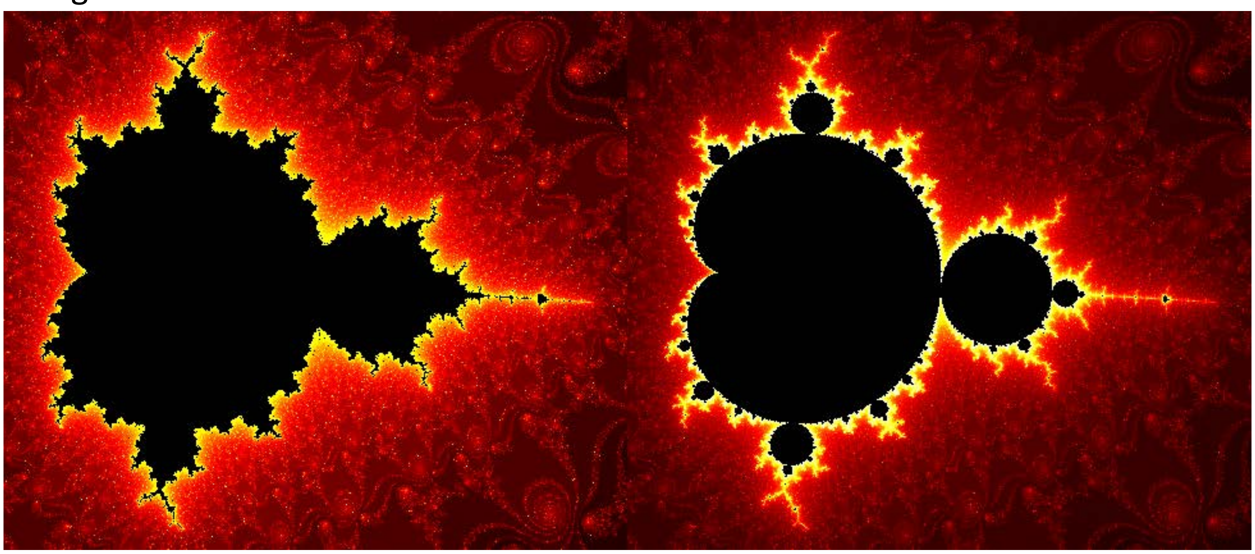
Recall, a number (c) belongs to the Mandelbrot set if, when run through a formula, the resulting values of z are always small – never exceeding a particular value (called the bailout criterion). But the only way a computer could determine that *no* value of z *ever* exceeds the bailout criterion would be to run it through the formula forever, which obviously it cannot do. So, in practice, we set an iteration limit. This is the number in the "iterate" box on the panel.

That is, we let the computer run through the formula, for say 1173 times, and if the value of z has remained small, it's a good bet that it always will. Therefore, the number (c) is very likely a member of the Mandelbrot set. However, there is some small chance that it really isn't! The higher the number of iterations, the more confident we can be that a point really does belong to the set. But, the computation will take longer.

So the number in the "iterate" box represents a trade-off between speed and accuracy. Higher numbers ensure greater accuracy, but take longer to plot. Note that the iteration number only affects regions of the actual Mandelbrot set, e.g. the points colored black. In the colorful regions far from the set, the iteration number has *no effect*; the graph is both fast and accurate in such areas.

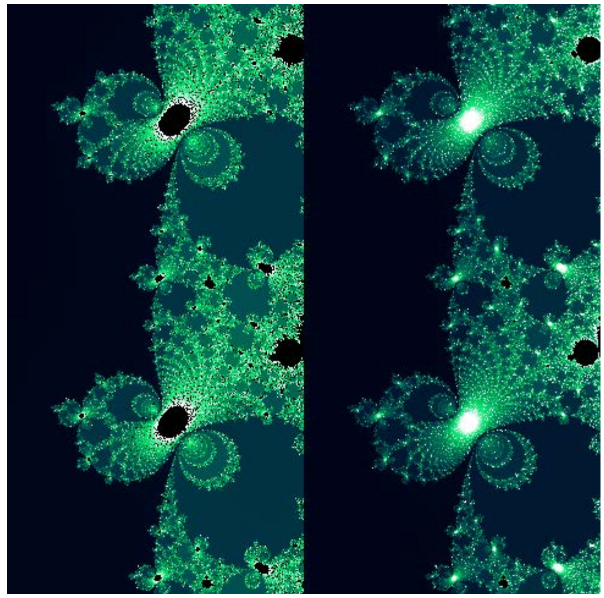
Generally, the more you zoom in on a portion of the Mandelbrot set, the higher the iteration number needs to be in order to ensure an accurate plot. But, unlike most other fractal graphing software, Fractal Grapher automatically adjusts the iteration number as you zoom in or out. *So, 95% of the time, you will not have to give this number a second thought.* But if you want to (or need to) adjust it, moving the slider to the left will tell Fractal Grapher that you prefer speed over accuracy, and it will adjust the iteration number accordingly. (It will continue to adapt as you zoom in, but will use a lower number than it would have otherwise.) Conversely, moving the slider to the right will cause Fractal Grapher to prioritize accuracy, and it will select higher values. You can also manually type in a value in the text box for the next plot. Generally, it's best to let the program select the value and leave the slider in the middle.

How do you know if the image is accurate enough? How do you decide when to move the iteration slider? Since there is no known mathematical formula that computes how many iteration cycles are sufficient, Fractal Grapher makes a "best guess." In a handful of situations, this guess will be wrong. There are two ways to recognize this:



First, if a mini-Mandelbrot looks "bloated" as it does on the left panel above, this means the iteration count is too low. Move the slider to the right to solve this, and replot the image. The new plot should look crisp as it does on the right panel. You might have to try several times to see how far you need to adjust the iteration count. Once the graph stops improving, there is no need to add further iterations as they will not improve the image further.

Second, a place where the program sometimes fails to predict a sufficiently high iteration count is deep in Seahorse Valley. If the central "web" of a "seahorse" contains a black circle or ellipse, the iteration count is too low, as in the left panel of the image. When this happens, move the "iterate" slider a bit to the right and replot. Repeat this until these black circles are gone, as in the right panel of the image. The Mandelbrot set does not contain any *isolated* circles; all genuine circles are attached to a larger circle or a cardioid. Therefore, if you see an isolated black circle or ellipse anywhere, it is an artifact – a blemish caused by an iteration number that is too low.

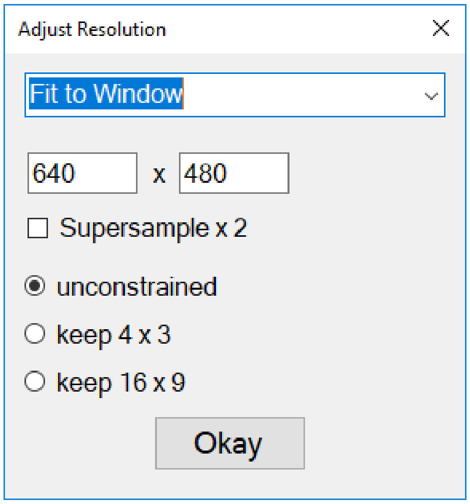


Resolution

The default size of the plotting window is 640 pixels wide by 480 pixels in height. Also by default, any image you save will match the resolution of the plotting window. However, both of these resolutions can be changed.

To change the size of the plotting window, simply grab any corner of the Fractal Grapher app, and drag it to change the size of the entire app – the plotting window will adjust accordingly. Since you can adjust the height and width independently, this means you can also change the shape – the aspect ratio – of the plotting window.

If you change the app size *after* plotting a graph, the graph will resize to fill the window, but its actual resolution will not change. To see an example of this, grab a corner and make the app as small as it can be; then press "Plot." Then grab a corner and enlarge the app to nearly fill the screen. The graph will increase in size to match, but it will look pixelated and blocky. This is because the image's true resolution is still very small. To fix this, just hit "Plot" again, and a crisp new image at the higher resolution will appear.



By default, the program plots the graph at the same resolution as the plotting window. But you can change this by pressing the “Res” button on the lower right of the panel. This will open a new “Adjust Resolution” window with many resolution options. The tab at the top defaults to “Fit to Window.” Click on this tab to see a dropdown list of other options.

One of these options is “1920 x 1080 [16:9] (HD)” which is a common resolution for many monitors and high-definition televisions. If we select that option, (and press “Okay”) then Fractal Grapher will internally plot all images in high definition, at 1920 x 1080 resolution. It will still *display* the images at the resolution of the plotting window. But when you save the image to a file, the image resolution will be 1920 x 1080. This way you can create images in higher resolutions without enlarging the plotting window. Custom resolutions are also an option. The standard version of the program has a maximum graphing resolution of 3840 by 2160 (4k).

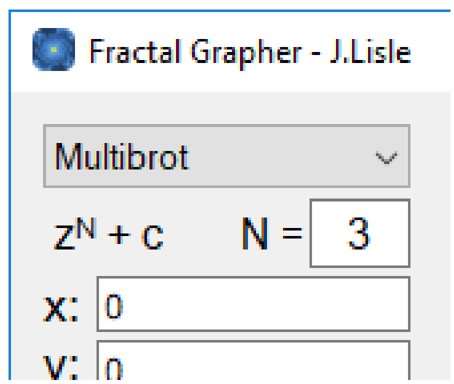
Higher resolutions take longer to plot because the computer has a larger number of pixels to evaluate. So, for slower computers, it may be best to use the “Fit to Window” option or a lower resolution when exploring the Mandelbrot set. And then switch to a higher resolution and replot once you have found a really interesting or beautiful graph that you want to save to disk. Another option for slower computers is the “Half Window (speed x4)” option. This makes the internal plotting resolution only half (in each dimension) that of the plotting window, resulting in slightly pixelated graphs. But it speeds up graphing time by a factor of four.

For the highest quality graphs, select the “Supersample x 2” checkbox in the Adjust Resolution window. This option will slow the graphing process by a factor of 4, but it results in beautifully crisp, anti-aliased images. Note that this option does not change the resolution of the final image. It simply improves the quality of that image by using quarter-pixel samples and averaging the result. Uncheck this box for speed; check it for quality.

Finally, the "Adjust Resolution" Window has three button options at the bottom which (optionally) constrain the aspect ratio of the plotting window. The default option ("unconstrained") means that you can independently adjust the height and width of the plotting window by grabbing and dragging any corner of the app. The other two options allow you to change the size of the window, but not its shape. They "lock" the aspect ratio. The 4:3 option is the aspect ratio used by older televisions, and 16:9 is the most common aspect ratio for modern televisions. Select one of these three options and click "Okay." Then grab any corner of the app and drag to see how these affect the plotting window.

Other Fractals

Fractal Grapher allows you to plot not only the Mandelbrot set, but nine other types of sets. The set to be plotted is indicated at the top of the panel. To select a different set, click on this tab to reveal a drop-down menu listing other sets. Let's select the second option – "Multibrot." The equation under this tab instantly changes to the Multibrot formula $z^N + c$. And a new option appears just to the right of this formula "N = 3" where the number "3" is in a text box that can be edited. This is because there is not just one multibrot. There are an infinite number of multibrots – one for each value of N. You explore a different multibrot every time you change this number. For now, let's leave it at the default value of 3.



Usually, when changing to a new equation, the first thing you will want to do is press the "Init" key. This will set the coordinates and range to view the entire set (for finite area sets), and will plot the new set. From there, you can explore this new set just as you did with the Mandelbrot set. To explore a different multibrot, change the power (N) in the top right box to something other than 3, and press "Init" to view the result. Higher powers gain additional "valleys" in the central area. You can also use fractional values (2.5), and even negative values. Try setting N = -2. The resulting map is amazing, particularly when you zoom in on a deep valley.

Setting N=2 results in the Mandelbrot set. Yes, the Mandelbrot is actually a multibrot of power N=2. However, if you want to explore the Mandelbrot set, it is best to set the formula to the Mandelbrot, rather than using the multibrot and setting N=2. Why? Fractal Grapher is optimized for the Mandelbrot formula. It will plot this set faster than any other, partly because there are certain mathematical tricks that can be used when N=2 that speed up computation. The multibrot formula does not use these tricks (even when N=2) because it allows for any value of N.

Note that the sets "Polynomial," "Lambda Variation," and "Julia" open a separate popup window that allows the user to specify additional parameters. For the Julia set (see below), this window may remain open. Only some parameters will result in a fractal. You'll have to experiment to see which sets are fractal, and which are not.

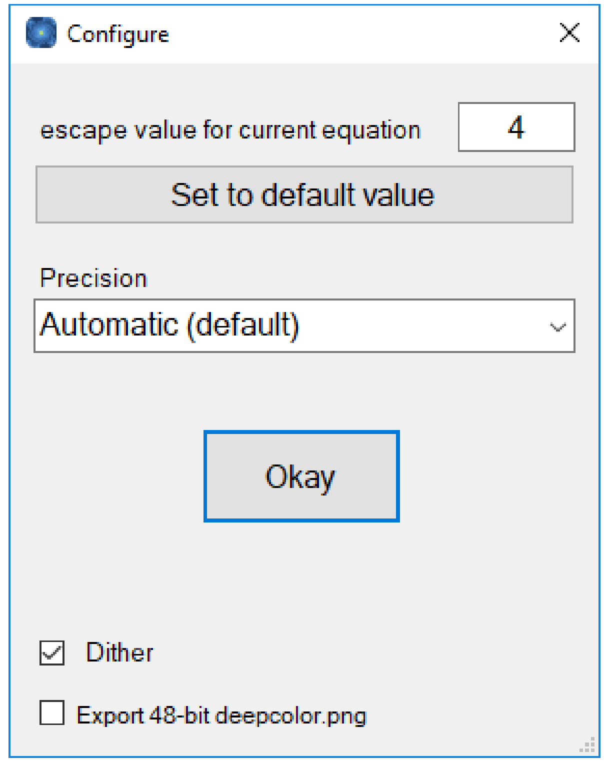
Plotting Speed

Since Fractal Grapher is optimized for the Mandelbrot set, you will notice – particularly on older computers – that the plotting speed is slower for other sets. This issue becomes more noticeable the deeper you zoom, because deeper zooms require more iterations and higher precision computations.

There are five ways to enhance speed. First, black points (those that belong to the set) take much longer to plot than colored points. This is because points that belong to the set must be checked for the entire iteration. This might guide what portions you choose to zoom in on. Second, try moving the "iterate" slider leftward. If it doesn't generate any "bloating" then the final result is accurate. Remember this only speeds up the plotting of points that belong to the set (areas in black). Third, reducing the resolution will speed up plotting as discussed above. Also, be sure to uncheck the "supersample" option. Fourth, reducing the value of the bailout criterion may speed up the plot slightly, as discussed below. However, this may reduce accuracy. Finally, reducing the computation precision level will speed up plotting, but may result in distortions or other less-than-accurate results. This is also discussed below.

Config

There are a few other adjustable parameters that you will probably never need to change – but you can if you want to. To access these, press the "Config" button. This opens a new "Configure" window with some additional options.



Bailout Criterion

The first option is the bailout criterion or "escape value" located at the upper right. This is the value that the sequence of "z" must eventually exceed in order for the point c to be marked as *not* a member of the set. Smaller numbers for the escape value result in slightly faster plots. For some sets, a larger value can result in more accurate maps – but only if the iterations are also increased. Increasing the bailout criterion without increasing the iterations can actually *reduce* accuracy.

Mathematicians have demonstrated that, for the Mandelbrot set, the bailout criterion can be as small as 2. That is, if any (absolute) value of any z in the sequence exceeds 2, then the sequence of z will increase without limit and the number c will not be a part of the Mandelbrot set. So, most fractal plotting software will set this value to 2. But you'll notice that Fractal Grapher sets the default value at four. The reason for this is another hidden feature in this software that results in extremely smooth color transitions.

Fractal Grapher has the ability to estimate non-integer escape iterations. Most fractal plotting software simply count the number of iterations of z it takes to escape the bailout criterion, 1,2,3, etc. and then assign a color. This results in distracting color bands where one color changes abruptly to another – particularly evident at larger scales. But Fractal Grapher has the ability to estimate the fraction of an iteration it would take to land exactly on the bailout criterion, such as 1.53 iterations, and can assign a color in between 1 and 2. This results in a smooth color transition across different iteration values. This feature is active for all sets except the exponential set and negative-powered multibrots.

This smooth interpolation feature works best if the bailout criterion is a bit higher than the bare minimum necessary value. Hence, the default value is 4 for the Mandelbrot, multibrots, and Lambda functions. If you set it to 2, the shapes will still be perfectly accurate, but you may see hints of color banding around the perimeter, particularly at low magnifications. The maximum allowed value for the bailout criterion is 255.

Bottom line: For the Mandelbrot, Julia, and (positive-powered) multibrot sets, you should generally leave the bailout criterion at its default value of 4 for smooth color transitions. You might drop it to 2 for deep zooms to slightly increase speed. Bumping the criterion up to 8 will make the color transitions even smoother. And there is no reason to increase the bailout criterion more for these sets. You might try higher numbers for the lambda set. For the sine, and exponential functions, you might try larger values, but remember to nudge the "iterate" slider to the right as well so that accuracy does not drop. For negative-powered multibrots, the bailout criterion strongly affects the size of the "pebbles." Larger values produce smaller pebbles. Experiment.

Precision

The next option in the "Configure" window is the precision option. By default, this is set to "Automatic," but if you click on this tab, a drop-down menu reveals five additional options. Fractal Grapher is capable of using single, double, triple, quadruple or arbitrary level precision in its computations. This basically refers to the number of digits that are computed following the decimal point. The more digits the computer needs to compute, the slower the computation. Modern computers are optimized to use single and double precision, so these modes are very fast. Triple precision is slower, but still fairly efficient. Quadruple precision is noticeably slower. Arbitrary level precision is painfully slow, but allows an unlimited number of digits. Arbitrary level precision is only available for the Mandelbrot and multibrot sets. The others will default to quadruple precision.

Generally, the more you zoom in on a set, the higher the precision needed to plot the result. Fortunately, Fractal Grapher knows what precision it needs for any given parameters in the Mandelbrot set. And if the precision option is set to “automatic” then the software automatically uses the minimum level of precision it needs in order to produce a fully accurate plot for maximum speed and efficiency. So, there is really no reason to ever change this for the Mandelbrot set. However, the program’s precision estimate may be slightly off for other sets, and so you do have the option to override.

If you are curious what precision level Fractal Grapher has selected for the next plot, this is displayed in the main window as a small text just to the right of the “r” (range) box. The numbers 1,2,3, and 4 indicate single, double, triple, and quadruple precision respectively. “AP” indicates arbitrary level precision. This symbol indicates the precision the program will use in the *next* plot *if the precision option is set to automatic.* So, when you draw a box in the plotting window, and that little indicator changes from 3 to 4, or from 4 to AP, you know the next plot will take significantly longer to render than the previous one (unless you manually override the precision level).

Dither

The next option on the “Configure” window is the Dither checkbox. This option is on by default, and basically makes the color transitions look very smooth on your monitor. Unless you really know what you are doing, this option should be left on.

So what is dithering? Nearly all computer monitors display color by combining three color channels: red, green, and blue. By mixing various amounts of these three colors, you can make any color on the screen. But each of these color channels has only so many possible values. On a standard monitor, each color channel has 256 possible colors, 0 for black, and 255 for the maximum color intensity. To store any possible value between 0 and 255 requires 8 bits of computer memory. And since there are three color channels, it takes 24-bits to store a color for a standard monitor – 8 bits for each of the three channels. So standard monitors are 24-bit, but they are sometimes marketed as 32-bit. This is because they are including a fourth transparency channel which is also 8-bit, although this is not a true color channel. (A few of the pricier monitors are 10-bit per channel, with 12-bit coming soon.)

A standard monitor can display 16.7 million unique colors. But, amazingly, the human eye can do even better. The eye can distinguish the subtle brightness difference between, for example, a value of 26 and a value of 27 in the red channel when the two are next to each other. Thus, a gradient in any color channel tends to exhibit a subtle banding effect as in the top panel of the following figure:

This is not a limitation of the software; it is a limitation of your monitor. Fractal Grapher is capable of generating 48-bit (or 64-bit with a transparency channel) images, resulting in 281 *trillion* colors. But your monitor cannot display them.

So, dithering is a “trick” to simulate higher color depth than the monitor can actually display. Suppose your monitor can display a red value of 26 or 27, but you want to display something in between – a value of 26.5 for example. This can be accomplished by coloring some of pixels in the region 26, and some of them 27. Since your eye cannot easily distinguish individual pixels, these values merge and you perceive a color of 26.5, even though your monitor cannot actually display a value between 26 and 27. The lower red bar in the above image is dithered. You can see that it completely eliminates any color banding.

High Color Depth (48-Bit Images)

Although no monitor can currently display 48-bit images (16 bits per channel), Fractal Grapher can produce them nonetheless. To do this, check the lower left box labeled “Export 48-bit deepcolor.png” in the “Configure” window, and then click the “Okay” button. With this option enabled, Fractal Grapher will automatically save a 48-bit image named “deepcolor.png” to its home directory every time you press the “Plot” button as soon as it has finished rendering the image to the screen, or after you adjust any parameter such as brightness or color palette. You do not have to press the “Save” button (unless you want to also save the 24-bit version). The 48-bit deepcolor.png image will have exactly the same resolution specified in the Resolution window – the same as if you saved the 24-bit version.

* If you like the result, you should *immediately move or rename the file* because the next time you plot a graph, Fractal Grapher will again save a 48-bit version to the filename “deepcolor.png”, overwriting any existing file (with no confirmation). This will continue every time you Plot a graph until you uncheck the “export 48-bit deepcolor.png” option in the Configure window.

* Unlike the 24-bit files you create using the “Save” button, deepcolor.png does ***not*** contain any plotting information. You cannot load it back in and continue where you left off. It’s just an ordinary image file. (But you can save a 24-bit version using the “Save” button which does contain that information.)

*The 48-bit deepcolor.png image is *never* dithered, regardless of whether the “Dither” box is checked. It ignores this option because dithering is not necessary for high color depth images.

Why 48-Bit?

There are two reasons why someone might want to produce a 48-bit color image. First, some high-quality monitors are capable of displaying 30-bit or even 36-bit images. Although they cannot display the full color spectrum of a 48-bit image, they will nonetheless look better than a standard 24-bit image. (Standard monitors can still display image files that have 48-bit color depth; but they will display them as 24-bit).

Second, if you plan to print the image, or otherwise adjust the colors using 3rd party software, 48-bit images are the way to go. Even though these images will look no different on a standard monitor (in fact, they may look slightly worse because they are not dithered and yet are displaying as 24-bit which may produce color bands), programs like Photoshop can still edit them. And multiple color adjustments will not result in accumulated round-off error as occurs with 24-bit images. When done making all color adjustments, you might then save the final result as a dithered 24-bit (8-bits per channel) image (Photoshop automatically dithers 8-bit per channel images by default).

Conversion to other color palettes will be far more accurate using 48-bit images. For example, most printers use a four-channel CMYK color scheme. Most printers will *accept* an RGB image, but they must internally convert the image to CMYK before printing. Additionally, software like Photoshop can also convert RGB images to CMYK directly, and this conversion is best done with 48-bit images.

User-Defined Custom Palette

Fractal Grapher offers the user 28 color scheme options shown in the squares just below the "Plot" button. The final square on the lower right is a simple grayscale going from 0 (black) to 255 (white) in each of the three color-channels. However, this grayscale scheme can be replaced with a user-defined color scheme.

To do this, you will need basic third-party software to create a bitmap file with the palette information.   This bitmap must be 24-bit (8-bits per channel) and exactly 7671 pixels in width, and 5 pixels in height. The position along the length indicates the color to associate with increasing iteration escape values. Colors near the left edge represent low values – points far off the set. Colors near the right edge represent high values, points very near the set. (No need to worry about points directly on the set since these are always set to black.) Generally, it looks best to have colors near the left of the image be darker, and colors near the right edge be extremely bright or even white – but it's up to you. The five vertical pixels are redundant – the color should not change going up or down.

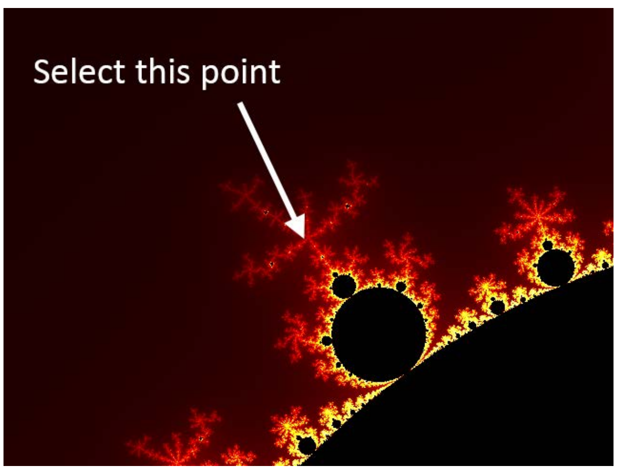
Since 7671 by 5 pixels is a very thin slice, it may be best to start with 7671 by 500 pixels so that you can better see the palette. Then, before saving the file, reduce the height to 5 pixels. In programs like Photoshop, you may want to use the gradient tool to create smooth horizontal gradients.

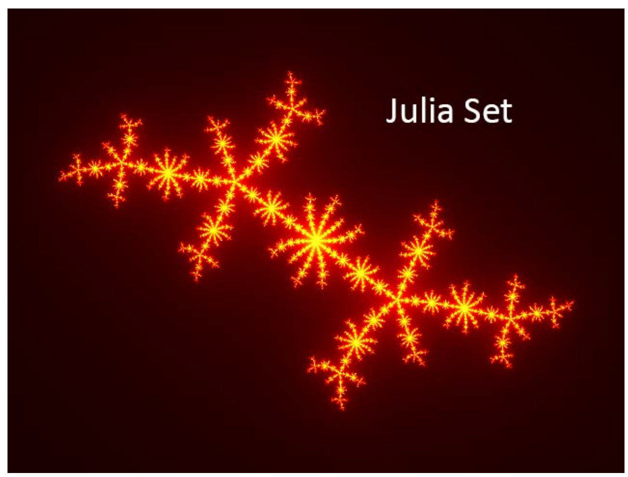
* For best results, you should first *disable dithering* in Photoshop, because it will interfere with Fractal Grapher's native dithering support. In some versions of Photoshop, this is done by clicking on the "Edit" tab and selecting the "Color Settings" option. Look for a box labeled "Use Dither (8-bit/channel images)" and *uncheck* the box. The resulting gradients may look banded due to the lack of dithering, but that's okay – Fractal Grapher will apply its own dithering. (It can also interpolate values to produce 48-bit images).

When you are happy with the resulting palette, remember to make sure the final image is exactly 7671 by 5 pixels, and is 24-bit color depth. The saved file must be named "palette.bmp" and placed in the same folder as the Fractal Grapher. Then restart Fractal Grapher and your new spectrum should appear in the lower right color box. If you delete palette.bmp, then Fractal Grapher will default back to its grayscale for the final color box upon restart.

Julia Sets

A Julia set is similar in many respects to the Mandelbrot set, except there are an infinite number of Julia sets. This is because each Julia set corresponds to a single point on the Mandelbrot map. Note that the point does not have to actually belong to the Mandelbrot set (it need not be black), but could be nearby. The resulting Julia set always resembles the region of the Mandelbrot map near the point on which it is based. For example, suppose you select a point on the Mandelbrot map that branches five ways, as in the left-hand plot below. The Julia set corresponding to this point will also have a five-fold branching pattern to it as in the right-hand plot below.





Every Julia set is based on the same equation as the Mandelbrot set ($z^2 + c$), but is plotted in a different plane. The Mandelbrot map represents different values of c, with the x-axis representing the real component, and the y-axis representing the imaginary component. The initial value of z is always set to zero, and the sequence of new values of z determines whether c belongs. But in a Julia set, c is fixed and based on one point in the Mandelbrot map. But the initial value of z is *not* required to be zero. A Julia set is a map of all the initial values of z that when run through the formula ($z^2 + c$) the sequence of z remains bound.

So, the Mandelbrot is a map in c-space, whereas a Julia set is a map in z_0-space. There is exactly one Julia set for each value c in the Mandelbrot map. And the Julia set always resembles the region of the Mandelbrot around that point c.

Interestingly, if c actually belongs to the Mandelbrot set (it will be colored black), then the Julia set based on that point c will be a *connected set*. That is, it will consist of black regions that are connected. If c does not belong to the Mandelbrot set, the corresponding Julia set will not be connected. Therefore, the Mandelbrot set is actually a map of all connected Julia sets!

Exploring Julia Sets

Since Julia sets are based on the Mandelbrot set, it is best to begin our exploration of them by selecting the Mandelbrot formula in Fractal Grapher. Then press “Init” to plot the complete Mandelbrot map. Next, zoom in on an area of interest such as a seahorse, an elephant, or a dendrite. Be sure to zoom in so that you are certain that the area of interest is well-centered. Then change the formula in the dropdown list to “Julia.”

A new window opens, showing the value of “c” you selected: first the real (x) and then the imaginary (y) component. This will be the “c” value for your Julia set. You can leave this window open if you wish. If you close it, you can re-open it by hovering over the formula tab.

Now press “Init” to center and plot the entire Julia set based on your selected value of c. Note that the x, y, and r values in the panel now represent the coordinates in z_0-space, not c-space. You can now zoom-in by drawing boxes and pressing “Plot” just as you did with the Mandelbrot set.

If you left the Julia window open or if you re-open it, you can manually change the values of c, thereby producing a different Julia map for the new value of c. But it may not be obvious where on the Mandelbrot set this new c is based. Therefore, it may be most helpful when plotting a new Julia set to briefly return to the Mandelbrot formula in order to select an interesting region.