The Mandelbrot Set

The Mandelbrot set consists of all complex numbers `c` so that the Julia set of `z^2 + c` is a connected set. If you play with my Java Julia Set Generator, you will find that the Mandelbrot set classifies Julia sets in the sense that connected components of the Mandelbrot set lead to topologically similar Julia sets. The simplest algorithm to generate the Mandelbrot set is based on the following theorem.

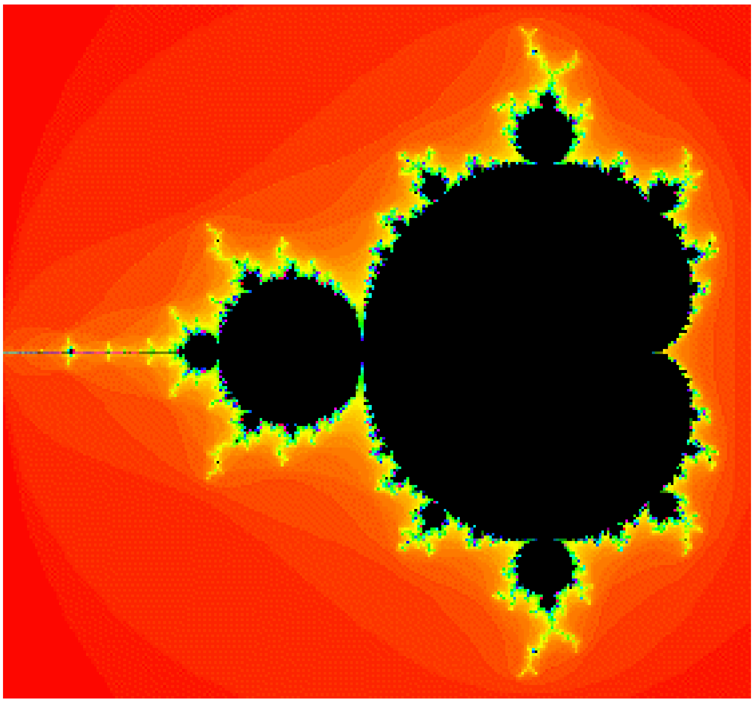
> The Julia set of `f(z) = z^2 + c` is connected if and only if the orbit of the critical point, `0`, is bounded by `2`. Otherwise, the Julia set is totally disconnected.

The orbit of zero is simply the sequence `{0,f(0),f(f(0)),...}`. Now, consider the following Mathematica function.

```
Mandel[x_,y_] := Length[FixedPointList[
        (#^2 + x + y*I) &, 0, 50, SameTest->(Abs[#]>2&)]];
```

Given a complex number `c = x + y*I`, `Mandel[x,y]` iterates the function `f(z) = z^2 + c`, terminating after 50 steps or if the term exceeds 2 in abolute value and then returns the number of steps required. We can now plot the Mandelbrot set as follows.

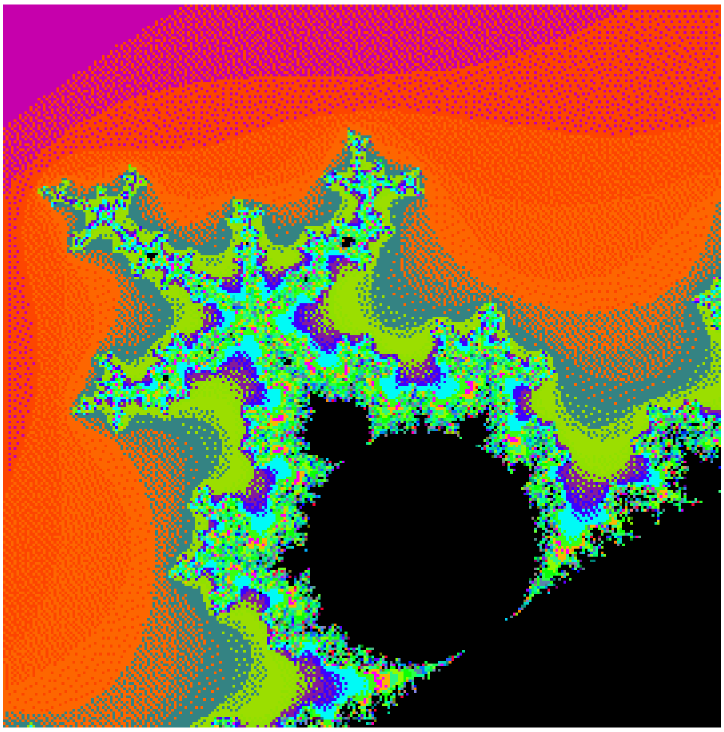
```
DensityPlot[Mandel[x,y], {x,-2,0.6}, {y,-1.3,1.3}, Mesh->False,
  AspectRatio -> Automatic, Frame -> False, PlotPoints -> 300,
  ColorFunction -> (If[#==1, RGBColor[0,0,0], Hue[.9#]]&)]
```



We can speed up the process considerably using Mathematica 3.0's `Compile` function. This allows us to up the iteration count and zoom in.

```
compiledMandel = Compile[{{c,_Complex}},
        Length[FixedPointList[#^2 + c &, c, 150,
                SameTest -> (Abs[#] > 2.0 &)]]];
DensityPlot[compiledMandel[x + I y], {x,-.65,-.4}, {y,.5,.75},
        PlotPoints -> 300, Mesh -> False, Frame -> False,
```

```
        ColorFunction -> (If[#==1, RGBColor[0,0,0], Hue[#]]&)]
```



Finally, we can get a 3D image by replacing `DensityPlot` with `Plot3D`.

```
Plot3D[compiledMandel[x+y I], {x,-2.8,1.2}, {y,-2,2},
        PlotPoints->200, Mesh -> False, Boxed -> False, Axes -> False]
```