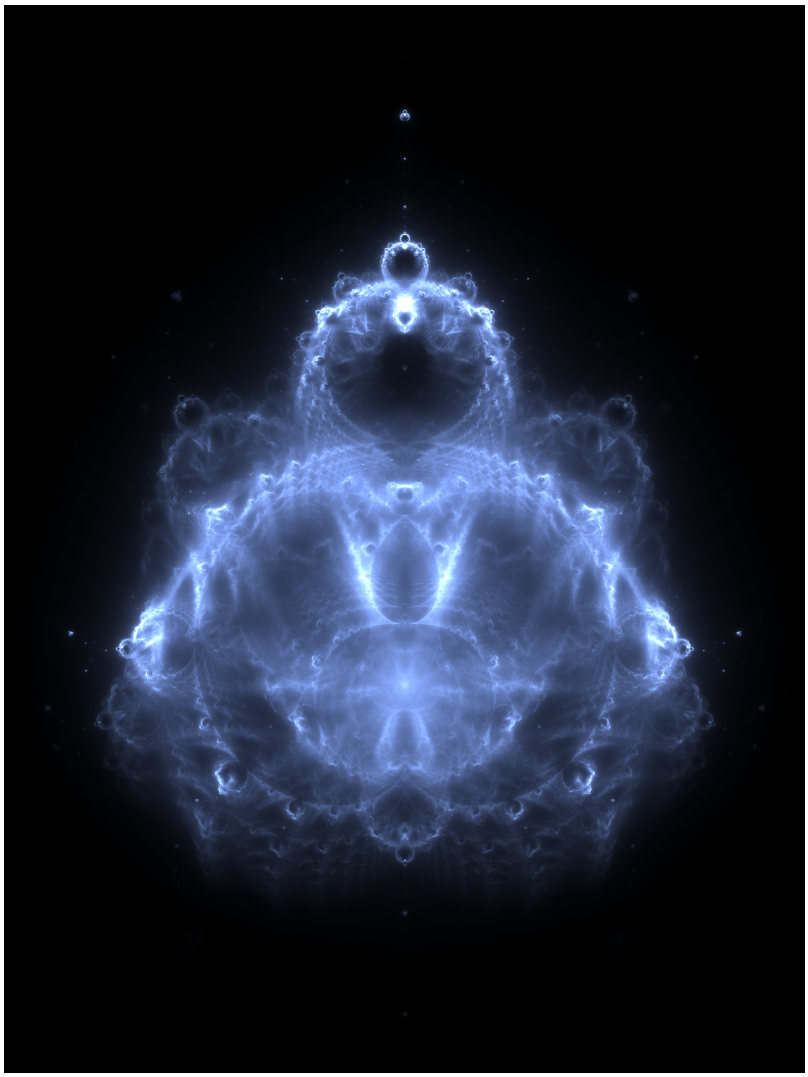
JULIA AND MANDELBROT

An Eternal Love Story in the Fractal World



A Project Report on Generating Mandelbrot and Julia sets by

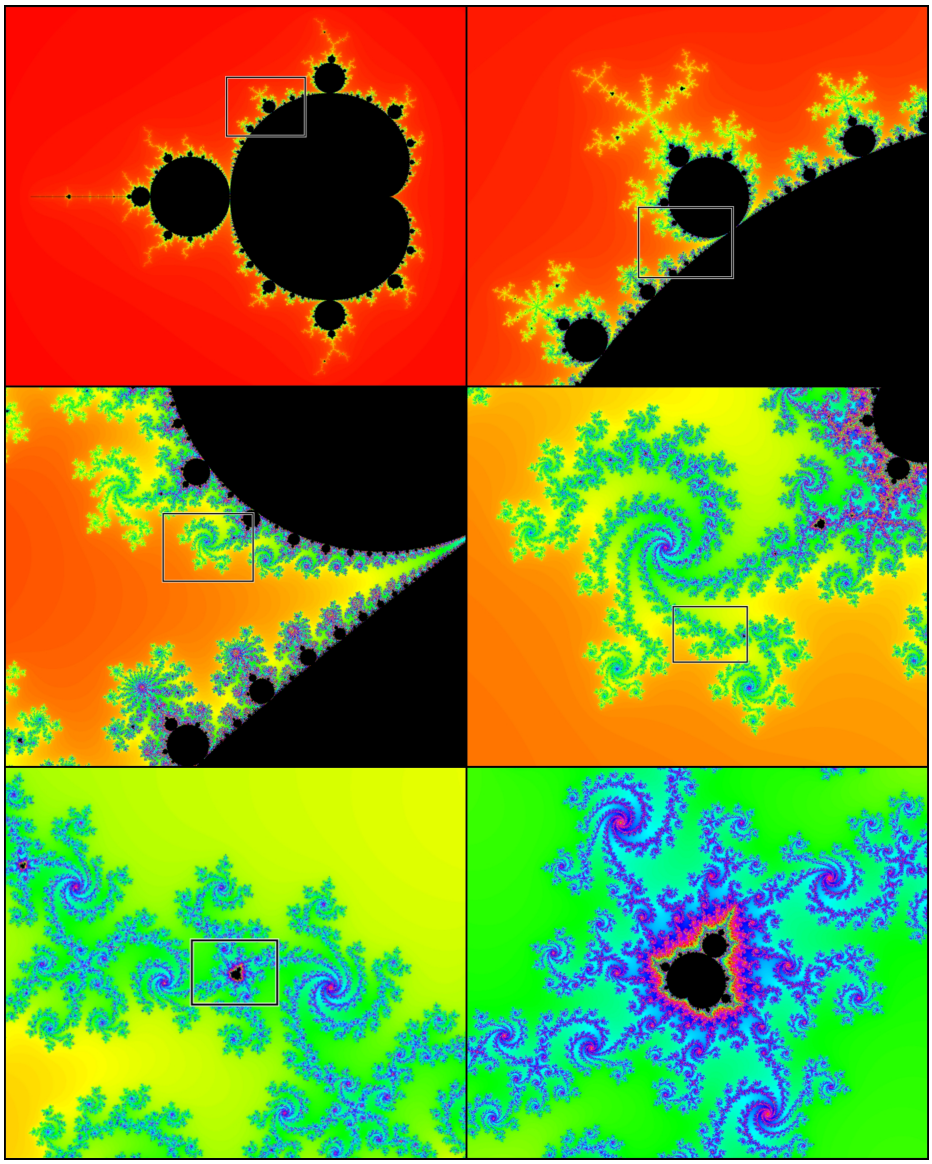
PRUTHWIN P B [22PHPH08]

SOMDEEP DEY [22PHPH17]

GUGULOTH CHANTI [22PHPH28]

"Bottomless wonders spring from simple rules, which are repeated without end."

-Benoît Mandelbrot



Contents

1 Chaos: The Calm IN The Storm

True nature of the ideal universe is geometrically impeccable, systematically symmetrical and stupendously stable. The disorder in the atmosphere, turbulence of the ocean waves, population fluctuations of species, etc. are all of null significance for reality is only an approximation of the truth. Or is it?

The onset of chaos unravels the truth, giving birth to the new ideal, only to make the observer realise their limit of perception. The true universe is dark, or rather kept in the dark, merely to simplify the problem under the lens. In fact, this dark side of reality is sporadic and discontinuous. The universe is really fickle; deterministic masquerading as having free will. The idea of taming this chaotic side was initiated by physicists, mathematicians, biologists, economists; chaos is EVERYWHERE. The fact that this revelation bleeds directly into our natural world formed the basis of what we call "Chaos Theory" today. To some physicists, chaos is a science of process rather than state, of becoming rather than being.

2 Fractal: The Ticket to Infinity

Chaos simply arises due to non-linearity in any system of interest. If the trajectories of the variable parameter of this non-linear system is traced, we get a plot that capture a fantastic and incredibly intricate structure. And thus, chaos spawns a language of stability (or a lack thereof), called the *fractal*.

It all began when a French mathematician, Gaston Julia, along with Pierre Fatou started working on the "Fatou set" - a set of complex numbers for which an iterative rational function is locally regular or convergent. And eventually, "Julia sets" were characterized as chaotic counterparts of Fatou sets, where the function was highly sensitive to initial points[1].

The term "*fractal*" was coined by Benoît B Mandelbrot, a French-American mathematician, working under Julia, much later in the 1960s; taken from Latin word, "*fractus*", meaning, to break. His studies of irregular patterns and exploration of infinitely complex shapes had an intellectual intersection: a quality of self-similarity. A symmetry across scale, implying a nested structure. Hence, a fractal is a way of seeing infinity. Staring into this infinite abyss, the abyss stared back into Mandelbrot, a new concept, which he would later call, *Fractal Geometry*[2].

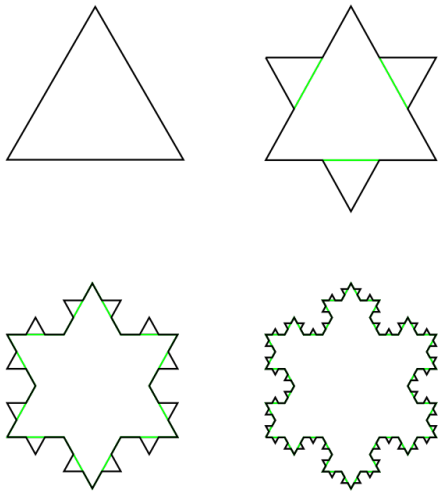


Figure 2.1: Koch Snowflake: A simple fractal where the middle $\frac{1}{3}^{rd}$
of a line of a triangle is made into a smaller triangle, exhibiting self-similarity.
(Source: Wikipedia)

3 The Mandelbrot Set: The Emblem of Chaos

The Mandelbrot set is a sacred object in mathematics. So sacred, it's often called the "Thumbprint of God". An eternity would not be enough to see it all. The infinitely zoomable digital imagery of this set seems *more fractal* than fractals. Mandelbrot started paying attention to the Julia sets and began to explore its properties *in depth*. Mandelbrot began using a computer to map out Julia sets, which are generated by plugging complex numbers into iterative functions with fixed constants. He tried to *literally* look into the Julia sets and played around by modifying the rules of generating Julia sets. Some Julia sets are filled, some are broken into regions, the others are still disconnected dusts. All of them exhibited self-similarity, implying the "*fractality*" of the Julia sets. Also, he himself created a fractal by simply varying the constant of the iterating function and discovered another set, which would soon be known as the *Mandelbrot set*[3].

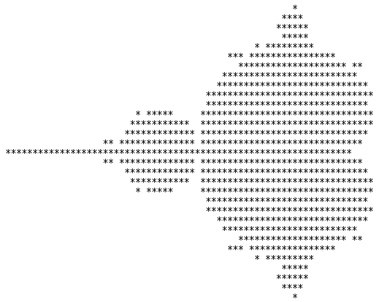


Figure 3.1: The First [published] Mandelbrot set.
(Source: Wikipedia

A closer look at the Mandelbrot set reveals that it's borders do not form crisp lines, but shimmer off like lighting, hinting chaos. Repeated magnification of the borders plunges one into a bottomless phantasmagoria of baroque imagery[4]. And a cardioid-like shape keeps recurring with subtle differences, depicting "*fractalness*". The Mandelbrot set was indeed a fractal!

The rules of THE Mandelbrot set are simple:

1. Take the simplest non-linear iterative function, $z_{n+1} = z_n^2 + c$. The terms z and c are complex numbers.
2. Assign a value to c, and calculate z_1 with $z_0 = 0$.
3. Repeatedly iterate the equation n times.
4. Repeat the process for a different value of c.

For some values of c, iterative function produces outputs that swiftly soar towards infinity. Other values of c produce converging or oscillating outputs. And then there are cs that lie in between, for which the outputs eternally skitter about. This third set of cs constitutes the Mandelbrot set. An escape algorithm is used to color-code the Mandelbrot set, where the converging points are painted black, and a color is assigned to a c based on how quick it produces an infinity, forming a sort of gradient for high iterations.

Part of the charm of this set is that it springs from a very basic *non-linear* equation, and yet, the moment non-linearity creeps in, the system can and will exhibit chaotic behaviour. This is a major give away that connects chaos with reality, as the true reality is purely non-linear.

The beauty of non-linearity clicks into computer screens when the set is to be code-plotted: Generation of the entire Mandelbrot set (infinite) is possible with just a few dozen lines of code (finite).

3.1 Mandelbrot set Generator Code

```
# Python code for generating the Mandelbrot Set

# Importing necessary libraries
from PIL import Image
import matplotlib.pyplot as plt
import colorsys
import time

# start time
start_time = time.time()

# Image size (pixels)
WIDTH = 600
HEIGHT = 520

# Maximum number of iterations
MAX_ITER = 100

# Defining the region of the complex plane
RE_START = -2
RE_END = 1
IM_START = -1.3
IM_END = 1.3

# Creating a blank image
img = Image.new('RGB', (WIDTH, HEIGHT), (0, 0, 0))

# Iterating over each pixel in the image
for x in range(WIDTH):
    for y in range(HEIGHT):
        # Map pixel coordinates to complex plane coordinates
        c = complex(RE_START + (x / (WIDTH - 1)) * (RE_END - RE_START),
                    IM_START + (y / (HEIGHT - 1)) * (IM_END - IM_START))
        z = 0
        for i in range(MAX_ITER):
            if abs(z) > 2:
                break
            z = z*z + c
        # Mapping the number of iterations to a color
        if abs(z) > 2:
            # Mapping in HSV Colour Space
            hue = (i / MAX_ITER) * 0.5
            saturation = 1
            value = 1 if i < MAX_ITER else 0  # Make the set black
            # Converting HSV to RGB for the ease of pixel painting
            r, g, b = [int(255 * j) for j in colorsys.hsv_to_rgb(hue,
                  saturation, value)]
            color = (r, g, b)
        else:
            color = (0, 0, 0)
        img.putpixel((x, y), color)
```

```
# Displaying the image
plt.imshow(img)
plt.show()

# Saving the image with file name having iteration information
file_name = 'Mandelbrot'+'{}.png'.format(MAX_ITER)
img.save(file_name)

# end time
end_time = time.time()

# calculating runtime
runtime = end_time - start_time

# printing runtime
print("Runtime:", runtime, "seconds")
```

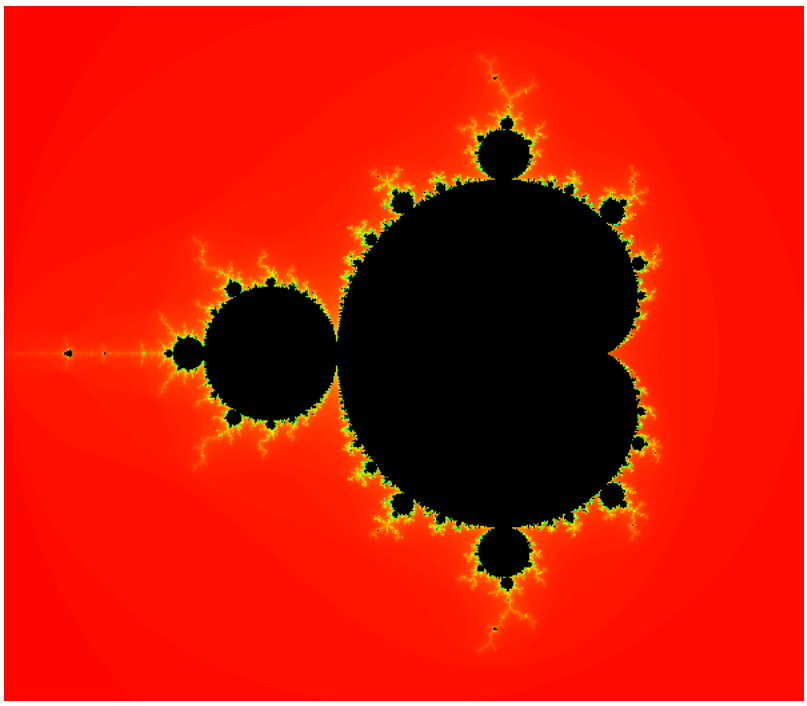


Figure 3.2: Output: The Mandelbrot set with 150 iterations.

Note: Zooming is possible by shifting the frame to the new origin (point of interest) and dividing (to zoom in) or multiplying (to zoom out) the endpoints by a zoom-factor. Zoom-factor is chosen in such way that doesn't affect the aspect ratio. Zooming in is demonstrated in the second page of front matter to illustrate Mandelbrot's quote.

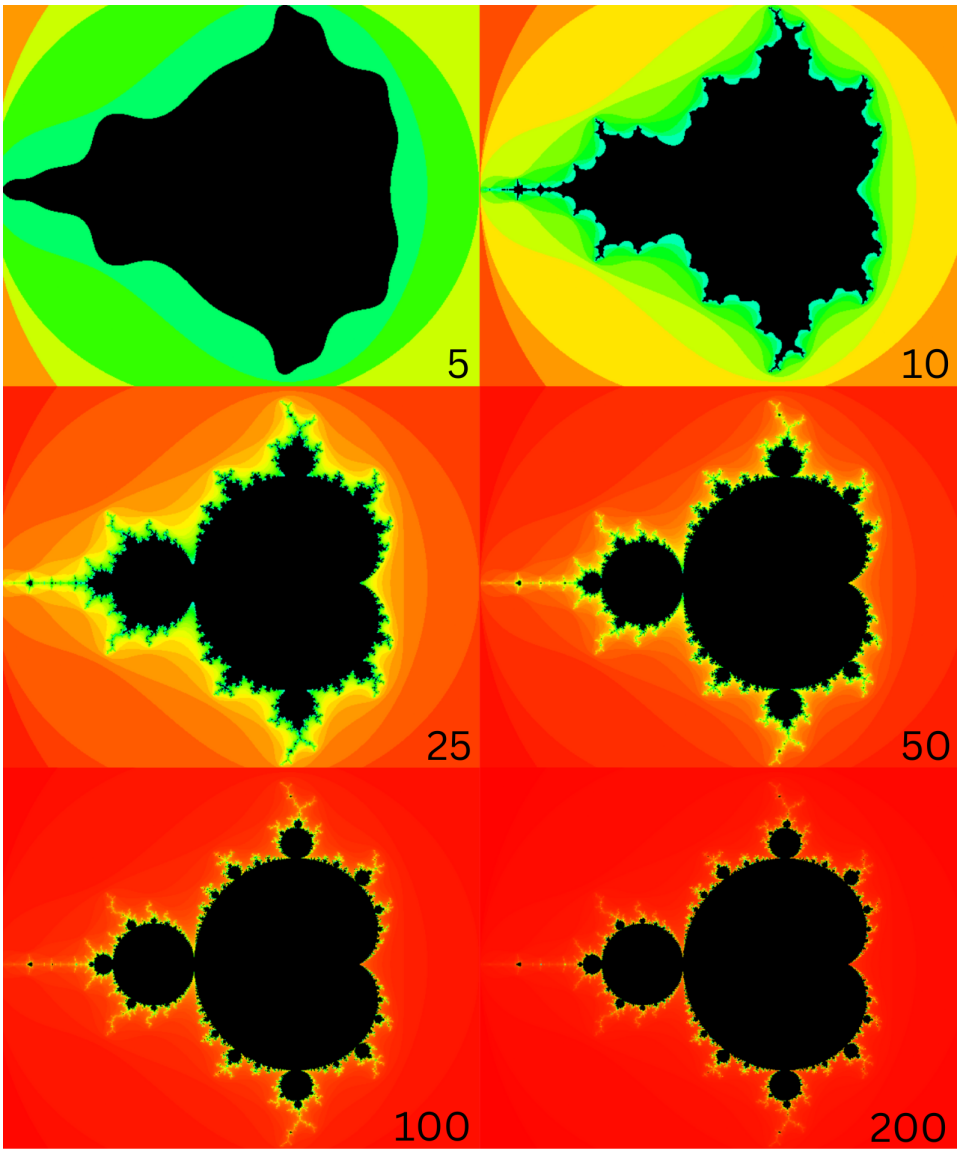


Figure 3.3: Mandelbrot set with various number of iterations (bottom-right).

However, Mandelbrot wasn't done with Julia sets.

4 The Julia Set: "*A*" Better Half

The Julia Set is the other part of the charm (and the reason why the first part is charming). A Julia set is a fractal set that is defined by an iterative function applied to a chosen complex number. For some points in the complex plane, the outputs do not escape to infinity under repeated iterations. Much like the Mandelbrot set, the Julia set is the boundary between complex numbers that remain bounded and those that escape to infinity under the iterative function.

The algorithm for THE Julia set:

1. Take the simplest non-linear iterative function, $z_{n+1} = z_n^2 + c$. The terms z and c are complex numbers.
2. Assign a value to c, and calculate z_1 with $z_0 = 0$.
3. Repeatedly iterate the equation n times.
4. Repeat the process for a different value of z.

Everything is the same as for Mandelbrot set, but here, z is changing. The color-coding is also quite similar—black points converge/oscillate and coloured gradient points narrate their escape-to-infinity experience.

4.1 Julia set Generator Code

```
# Python code for generating Julia Sets

# Importing necessary libraries
from PIL import Image
import matplotlib.pyplot as plt
import time
import colorsys
import os

# start time
start_time = time.time()

# Image size in pixels
WIDTH = 900
HEIGHT = 780

# Maximum number of iterations
MAX_ITER = 100

# Defining the region of the complex plane
RE_START = -1.5
RE_END = 1.5
IM_START = -1.5
IM_END = 1.5

# Choosing a complex number for the Julia set using dynamic input
rp = float(input("Enter the real part: "))
ip = float(input("Enter the imaginary part: "))
c = complex(rp, ip)

# Creating a blank image
img = Image.new('RGB', (WIDTH, HEIGHT), (0, 0, 0))

# Iterating over each pixel in the image
for x in range(WIDTH):
```

```
    for y in range(HEIGHT):
        # Map pixel coordinates to complex plane coordinates
        z = complex(RE_START + (x / (WIDTH - 1)) * (RE_END - RE_START),
                    IM_START + (y / (HEIGHT - 1)) * (IM_END - IM_START))
        for i in range(MAX_ITER):
            if abs(z) > 2:
                break
            z = z*z + c
        # Mapping the number of iterations to a color
        if abs(z) > 2:
        # Mapping in HSV Colour Space
            hue = (i / MAX_ITER) * 0.5
            saturation = 1
            value = 1 if i < MAX_ITER else 0  # Make the set black
            # Converting HSV to RGB for pixel painting
            r, g, b = [int(255 * j) for j in colorsys.hsv_to_rgb(hue,
                  saturation, value)]
            color = (r, g, b)
        else:
            color = (0, 0, 0)
        img.putpixel((x, y), color)

# Flipping the image vertically (Image from PIL has origin at the top-left)
img = img.transpose(Image.FLIP_TOP_BOTTOM)

# Displaying the image
plt.imshow(img)
plt.show()

# Saving the image with file name having complex coordinates
file_name = '{}+i({}).png'.format(c.real, c.imag)
img.save(file_name)

# end time
end_time = time.time()

# calculating runtime
runtime = end_time - start_time

# printing runtime
print("Runtime:", runtime, "seconds")
```

A Julia set has a wide variety of infinite shapes and patterns, depending on the chosen C. Some Julia sets are simple and connected, while others are complex and highly fragmented.

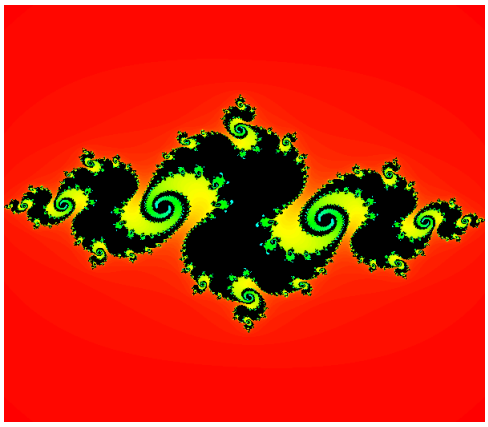


Figure 4.1: Output: $c = -0.7676 + i0.093$

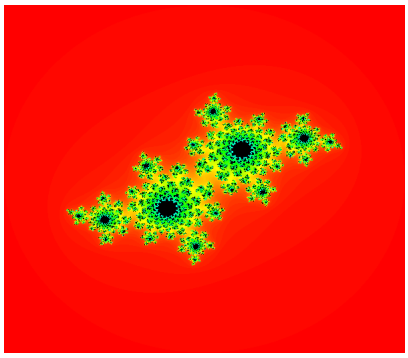


Figure 4.2: $c = 1.57 - i0.6$: Julia Dust.

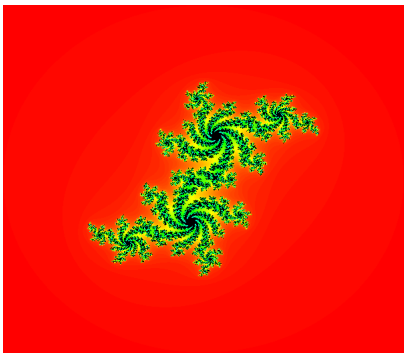


Figure 4.3: $c = 0.02 - i0.5$: A Filled Julia.

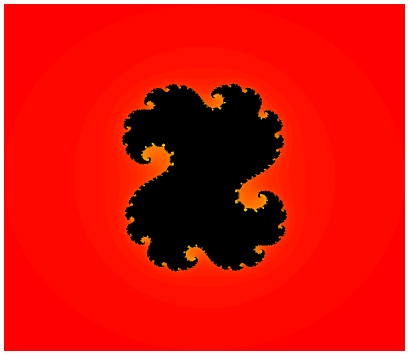


Figure 4.4: $c = 0.2929 - i0.0269$: A Filled Julia.

Figure 4.5: $c = -0.57 + i0.19$: Another Filled Julia.

5 Mandelbrot's Epiphany: The Amorous Entanglement

The Julia sets and the Mandelbrot set both stem from the same equation. Mandelbrot wanted to visualise a link between the Julia sets and Mandelbrot set. In 1979, he discovered that one could create one image in the complex plane that would serve as a catalogue of Julia sets, a guide to each and every one. As he tried calculating in finer and finer detail, instead of becoming sharper, the pictures became messier. To his surprise, the growing messiness was the sign of something real. Sprouts and tendrils spun languidly away from the main island. Mandelbrot saw a seemingly smooth boundary resolve itself into a chain of spirals like the tails of sea horses. The irrational fertilized the rational[5].

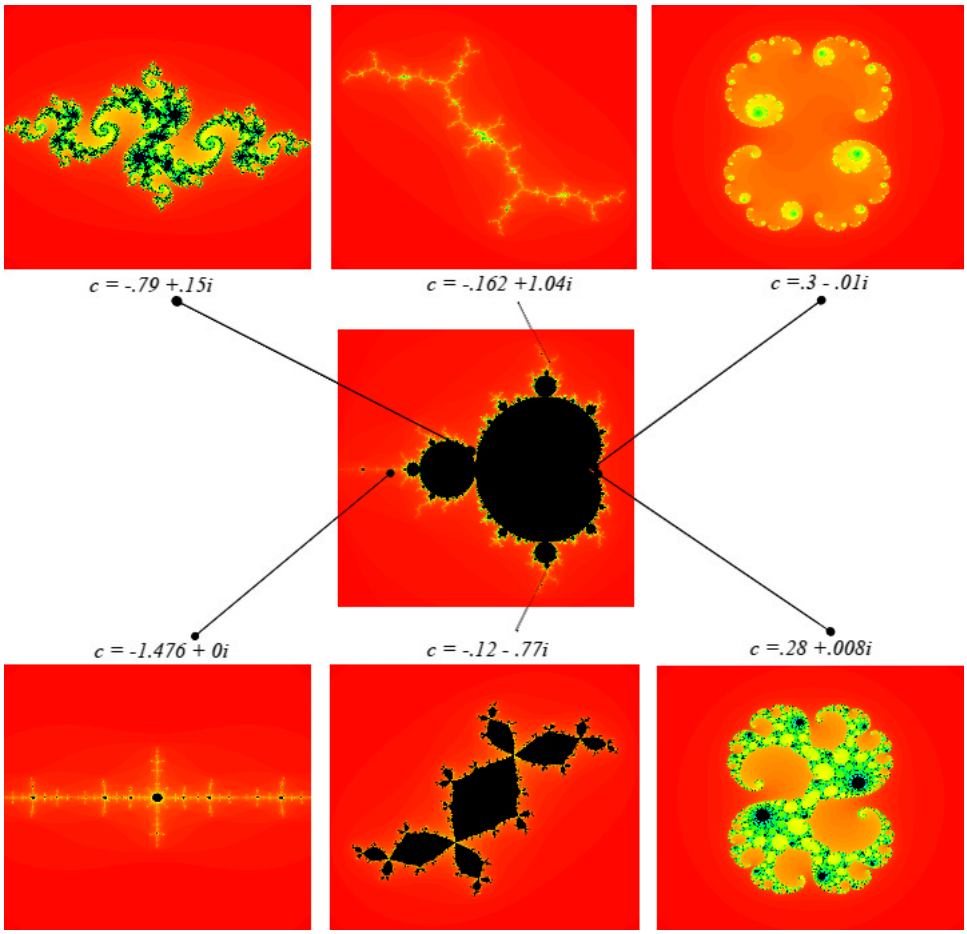


Figure 5.1: Every point on the Mandelbrot set contains a Julia set.

When Mandelbrot plotted the points in the complex plane that did not escape to infinity under iteration of a particular quadratic polynomial, he noticed that they formed a connected and highly intricate shape, which was indeed, the Mandelbrot Set.

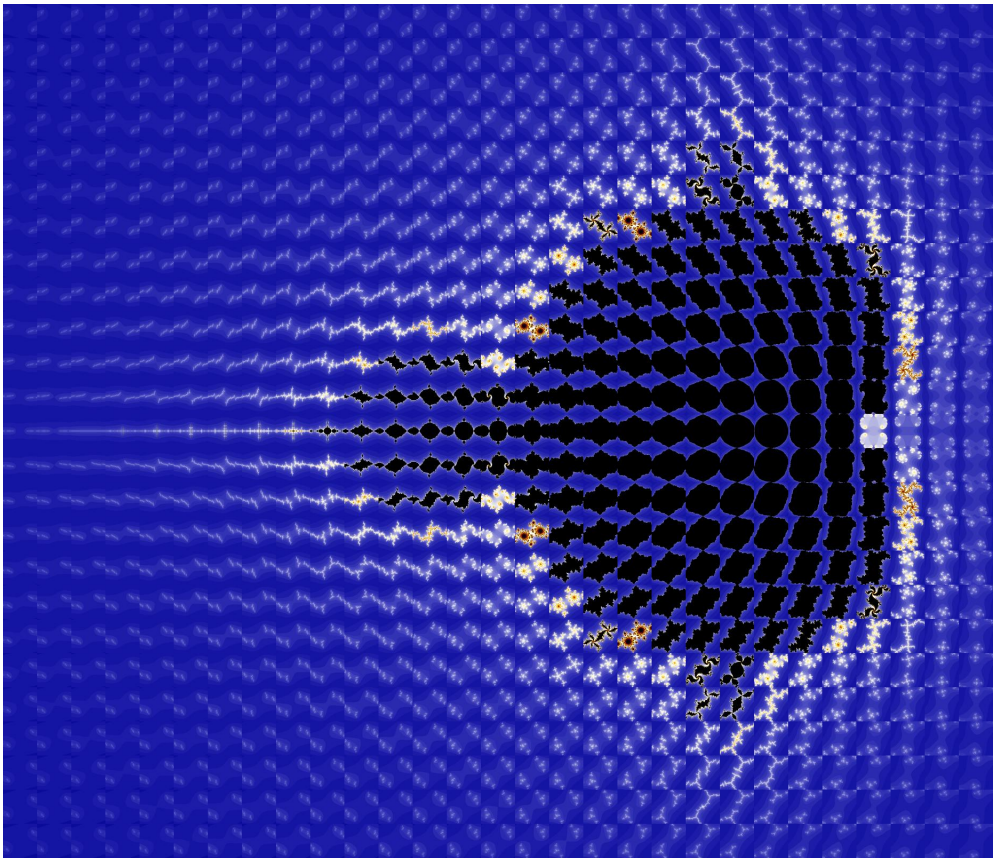


Figure 5.2: The Complete Julia Set Catalogue.
Together, they resemble a Mandelbrot set.
(Source: i.imgur.com/bc7vL.jpg)

6 One step further: Decoding the Logistic Map

The concept of a Logistic Map boils down to a plot between the growth rate of a population and the equilibrium population attained after generations[6]. This is applicable and valid in population biology[7]. So, the population of the next generation, x_{n+1}, with a growth rate, r, is given by $x_{n+1} = rx_n(1-x_n)$. This can be rearranged as $z_{n+1} = z_n^2 + c$, where $z_n = r(\frac{1}{2} - x_n)$ and $c = \frac{r}{2}(1 - \frac{r}{2})$. We're now in real space[8]. If the number of iterations is plotted on the z-axis, the z-x plot is, fascinatingly (can't stress it enough), a LOGISTIC MAP!
Pardon the pun, but the Mandelbrot set *also* talks about population dynamics in a different dimension! This was realised by yet another French mathematician, Adrien Douady and his student, John H. Hubbard, who studied the Mandelbrot set in *depths and angles*.

THE MANDELBROT SET IS UNIVERSAL.

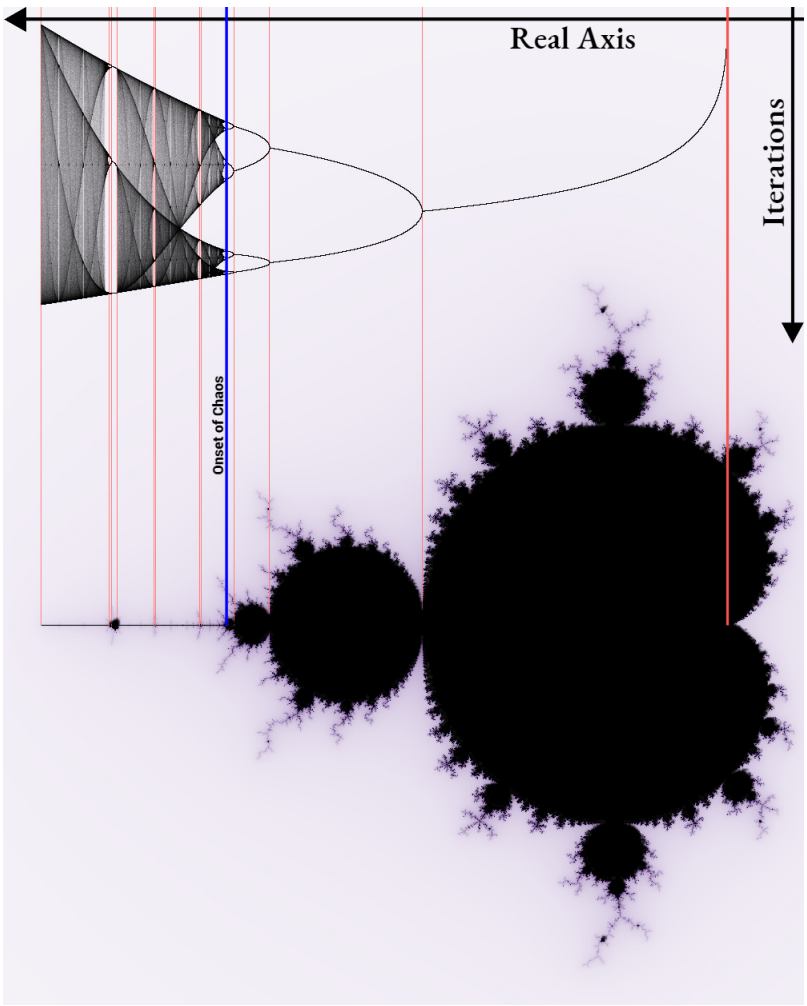


Figure 6.1: Evidence of presence of Logistic Maps in the Mandelbrot set. The number of curves along the vertical at any point on the real axis is the number of converging points inside the Mandelbrot set. Onset of chaos is around 3.57 on the real axis.

References

[1] Scott Sutherland. An introduction to julia and fatou sets. In *Fractals, Wavelets, and their Applications: Contributions from the International Conference and Workshop on Fractals and Wavelets*, pages 37–60. Springer, 2014.

[2] Benoit B Mandelbrot and Benoit B Mandelbrot. *The fractal geometry of nature*, volume 1. WH freeman New York, 1982.

[3] Heinz-Otto Peitgen, Hartmut Jürgens, Dietmar Saupe, and Mitchell J Feigenbaum. *Chaos and fractals: new frontiers of science*, volume 106. Springer, 2004.

[4] John Horgan. Mandelbrot set-to. *Scientific American*, 262(4):30–35, 1990.

[5] James Gleick. *Chaos: Making a new science*. Penguin, 2008.

[6] May Robert. Simple mathematical models with complicated dynamics. *Nature*, 261:459–467, 1976.

[7] Robert Mccredie May. Chaos and the dynamics of biological populations. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 413(1844):27–44, 1987.

[8] M Frame, B Mandelbrot, and N Neger. *Fractal Geometry*. Yale University, 2000.