Intro to Fractals

David Quarel

1 Background

The Mandelbrot Set is the collection of all points $c \in \mathbb{C}$ such that the sequence $z_0, z_1, z_2, \ldots$ defined by

$$z_n = \begin{cases} 0 & \text{if } n = 0 \\ z_{n-1}^2 + c & \text{else} \end{cases}$$

always remains bounded (i.e There exists $\epsilon \in \mathbb{R}$ such that $\forall n \in \mathbb{N}, |z_n| < \epsilon$). This is a surprisingly simple definition, but the emergent behaviour we see is very surprising. Below is a picture of the Mandelbrot set on the complex plane.
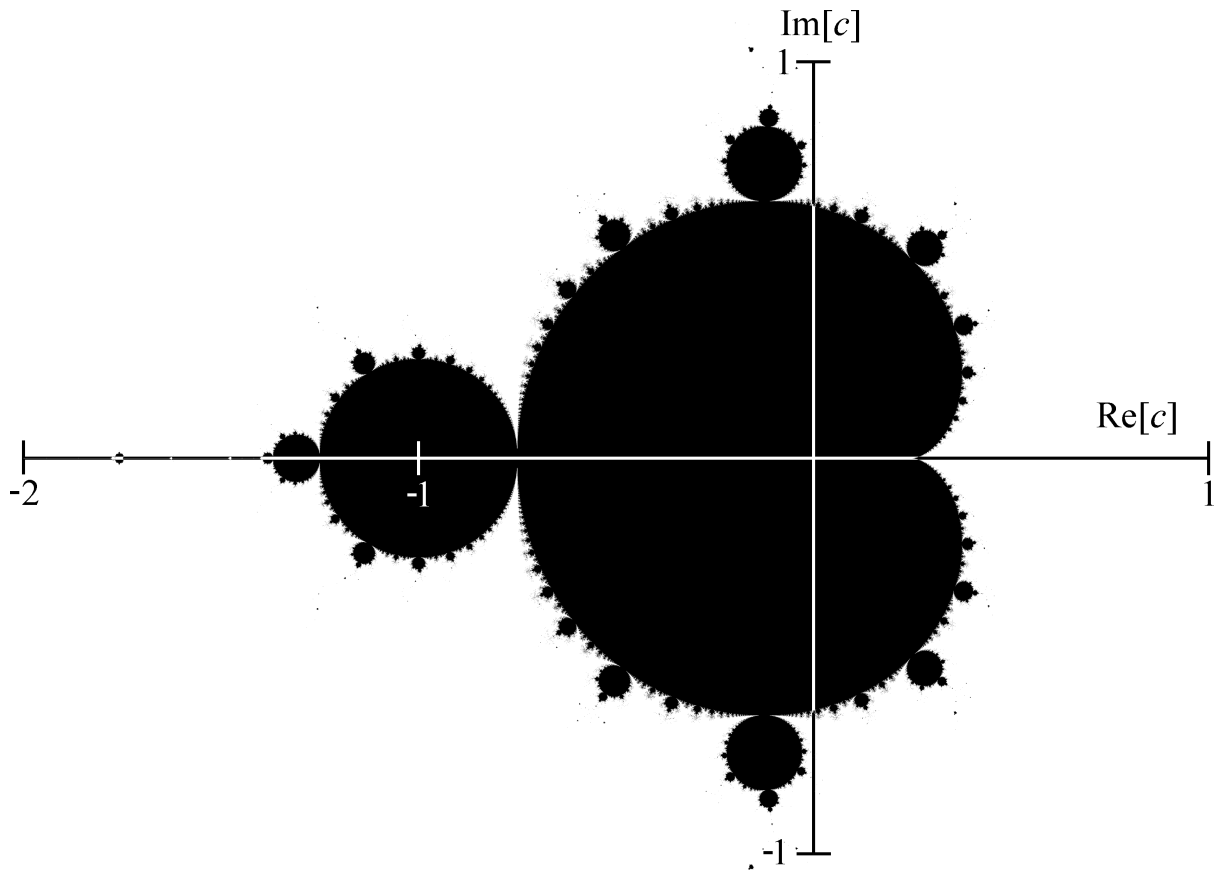

Figure 1: A picture of the Mandelbrot set[1]

The fractal is very complex, and has self similar properties, zooming in on one small part

of it looks like the entire set. We can be more specific with our definition by choosing $\epsilon = 2$ as the following theorem shows.

Theorem 1.1 (Mandelbrot Bound). *This theorem proves a bound on the size of the Mandelbrot set, which can also be used to show that if any term in the iterative sequence is outside this bounding ball, then that sequence diverges and the value of c for that sequence is not in the Mandelbrot set.*

Proof. Assume $z_n > 2$. Expressed in polar form, $z_n = re^{i\theta}$ for some $r > 2$ and some θ. Squaring z_n results in $z_n^2 = r^2 e^{i2\theta}$. Now we will show that $|z_n^2 + c| > 2$. c is an unknown constant, but we can represent it as an arbitrary complex number in polar coordinates. $c = |c|e^{i\phi}$. Now we choose ϕ to be the worst case such that the magnitude $|z_n^2 + |c||$ is minimised, and any other choice of ϕ will be bigger than this. The worst case is where c points in the opposite direction of z_n^2, or when $\phi = 2\theta + \pi$. So $|z_n^2 + c| = |r^2 e^{i2\theta} + |c|e^{i\phi}| \geq |r^2 e^{i2\theta} + |c|e^{i(2\theta+\pi)}| = |r^2 - |c|| \geq |4 - |c||$. Now if $|c| < 2$ then $|4 - |c|| > 2 \Rightarrow z_{n+1} > 2$ and we are done. If $|c| \geq 2$ then it does not matter where z_n^2 lands in the circle of radius 2, as translating any point in that circle by 2 units in any direction will leave the circle, so $|z_{n+1}| > 2$ and we are done. Hence $|z_n| > 2 \Rightarrow |z_{n+1}| > 2$. $\qquad\square$

This means that if a point c is in the Mandelbrot set, then $|c| < 2$, which gives us a bounding circle to choose points from to iterate over, as well as that if any point in the iterative sequence has a magnitude greater than 2, then that point will never renter the circle. Given the map $f : \mathbb{C} \mapsto \mathbb{C}, f(z) = z^2 + c$ is an increasing function for $|z| > 2$, being outside the circle guarantees divergence. This is useful as now when we numerically calculate the Mandelbrot set, we can run the iterative sequence until either one of the z_n is outside the bounding circle, where we can immediately halt and say for certain that the point is outside the set, or after some maximum number of iterations the point still has lot left the set, in which case we can say that the point is probably in the set. A greater number of iterations provides a more accurate representation of the true Mandelbrot set.

2 Computing the Mandelbrot set

Computing the Mandelbrot set is simple to implement, a simple implementation is to draw a box of length and width 4, centred at $(0,0)$, and then cut that box into a grid of squares. The coordinate point for each square is run through the Mandelbrot algorithm, and is coloured black or white depending on whether it is in the set or not. Obviously, cutting the box into smaller pieces results in a higher resolution image, and increasing the maximum iteration count will result in the picture looking very close to the actual Mandelbrot set, but in practise 50 iterations is more than enough for an accurate image. An implementation of this algorithm was done in C, which can be viewed at [2].

A better demonstration rather than just checking if a point is in the set or not and the colouring it black or white is to colour each point depending on how many iterations it takes for the point to exceed the bound (or leave it black if after the maximum permitted iterations it still does not leave the set). This results in some great looking images as seen below.
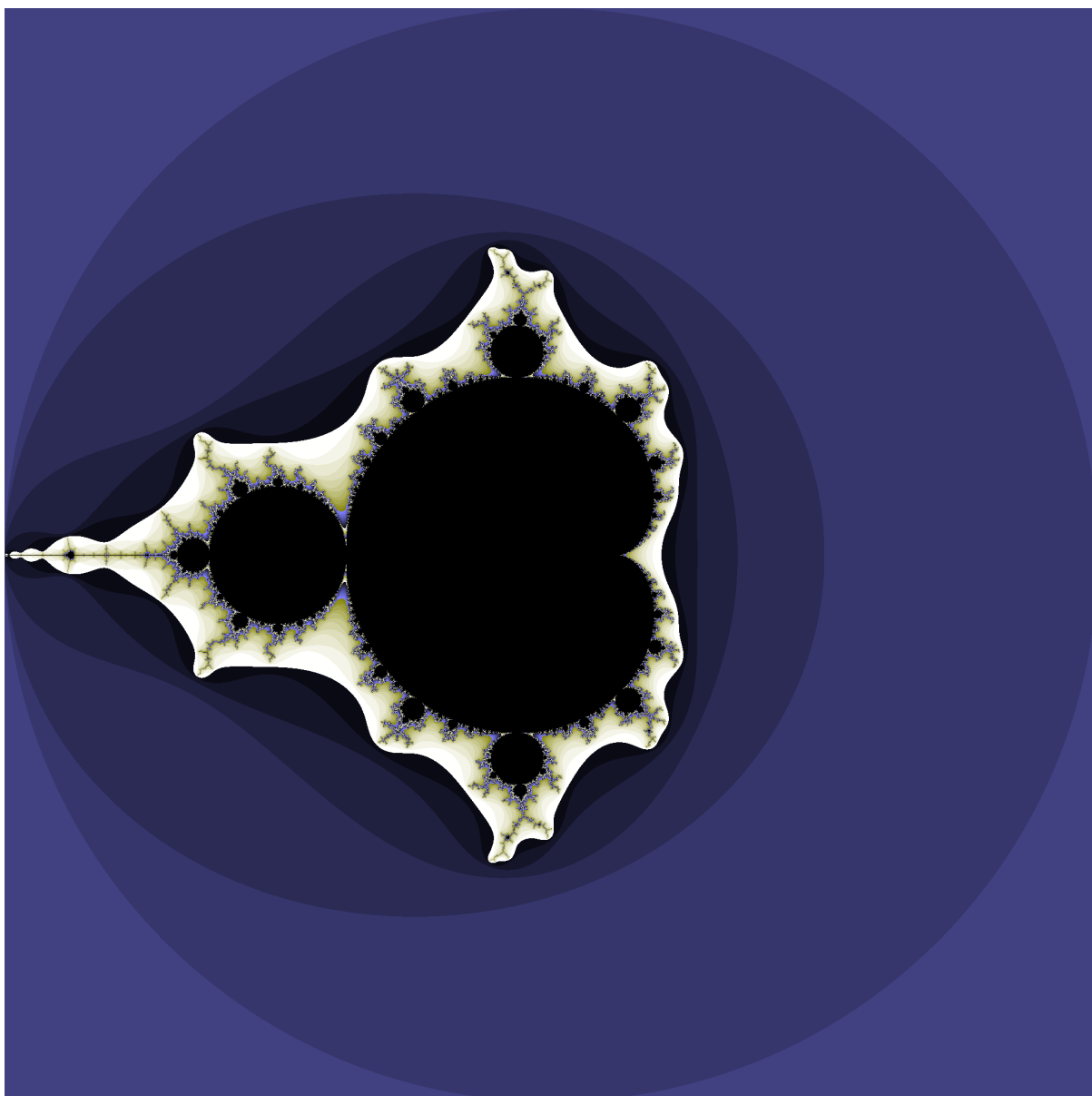
Figure 2: A picture of the Mandelbrot set with colouring algorithm applied

References

[1] https://en.wikipedia.org/wiki/Mandelbrot_set

[2] https://gitlab.cecs.anu.edu.au/u5354853/mandelbrot