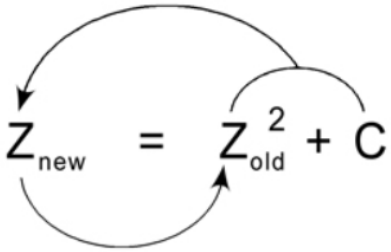
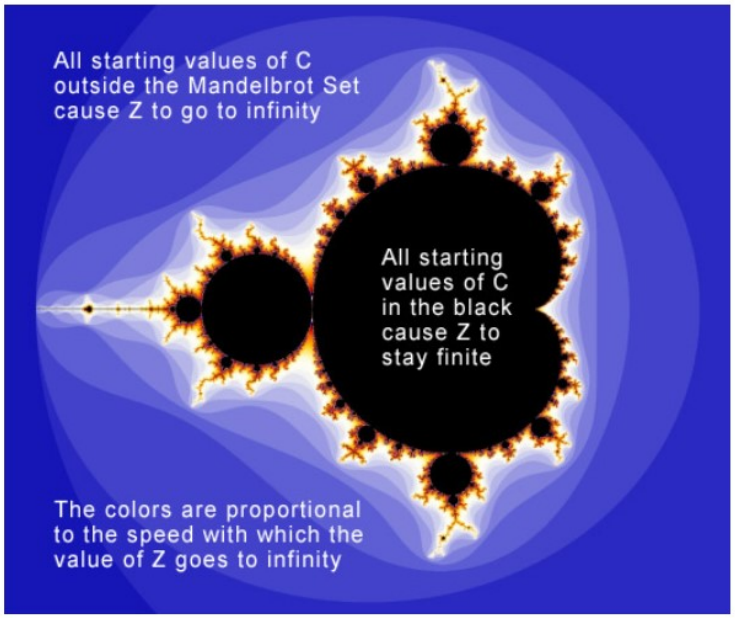
Creating an incredibly complex pattern by repetition of a simple process.

To generate the Mandelbrot Set, we use a computer program to calculate a simple equation over and over. Because we can **iterate** the equation as many times as we want, the fractals we create this way can be arbitrarily detailed, meaning we can zoom in as deep as we like.

When we iterate the following equation (calculate it over and over again), we generate the amazing Mandelbrot Set. To calculate the original image at the resolution of your screen, we only need to iterate the equation a few dozen times before it comes into focus. But to explore some of the super-deep images, we may need to compute the equation millions or even billions of times.



DEFINITION: The Mandelbrot Set is the collection of all starting values C that stay finite when iterated through the equation $Z_{n+1} = Z_n^2 + C$. Where Z and C are Complex Numbers.



We are interested in the fate of Z for different starting values of C. The Mandelbrot set lies within a circle of radius 2, so the entire width of the image should have length 4.

We assume that the sequence Z*n* is not bounded if the modulus of one of its terms is greater than 2.

mandelbrot_serial.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
#include <fstream>
#include <complex>
#include <sstream>

using namespace std;

float width = 600;
float height = 600;
float XMIN=-1.5;
float XMAX=-0.5;
float YMIN=-0.5;
float YMAX=0.5;


int value (int x, int y)  {
    complex<float> point((float)x*(XMAX-XMIN)/width+ XMIN, (float)y*(YMAX-YMIN)/height+YMIN);
    complex<float> z(0, 0);
    int nb_iter = 0;
    while (abs (z) < 2 && nb_iter <= 20) {
        z = z * z + point;
        nb_iter++;
    }
    if (nb_iter < 20)
       return (255*nb_iter)/20;
    else
       return 0;
}
```

$$x_{k+1} = x_k^2 - y_k^2 + \operatorname{Re} c$$
$$y_{k+1} = 2x_k y_k + \operatorname{Im} c$$