The Mandelbrot Set: Kernel Review

Neill Warrington
PHYS 305
Spring 2012

Introduction

- What is the Mandelbrot Set?

$$M = \{c \in \mathbb{C}, \forall n \leqslant N, |P_c^n(0)| \leqslant s\}$$

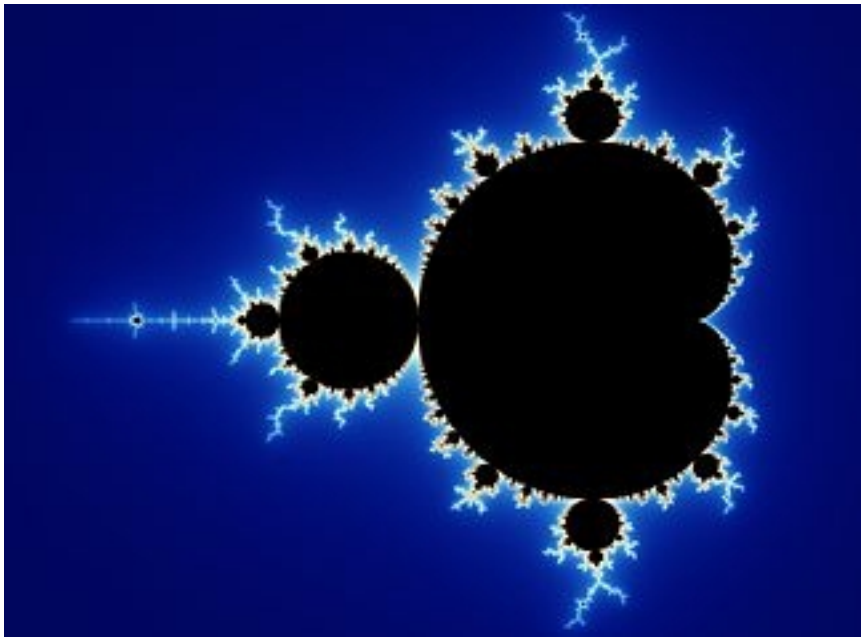
where,

$$c = a + bi$$

$$P(0): z \to z^2 + c$$

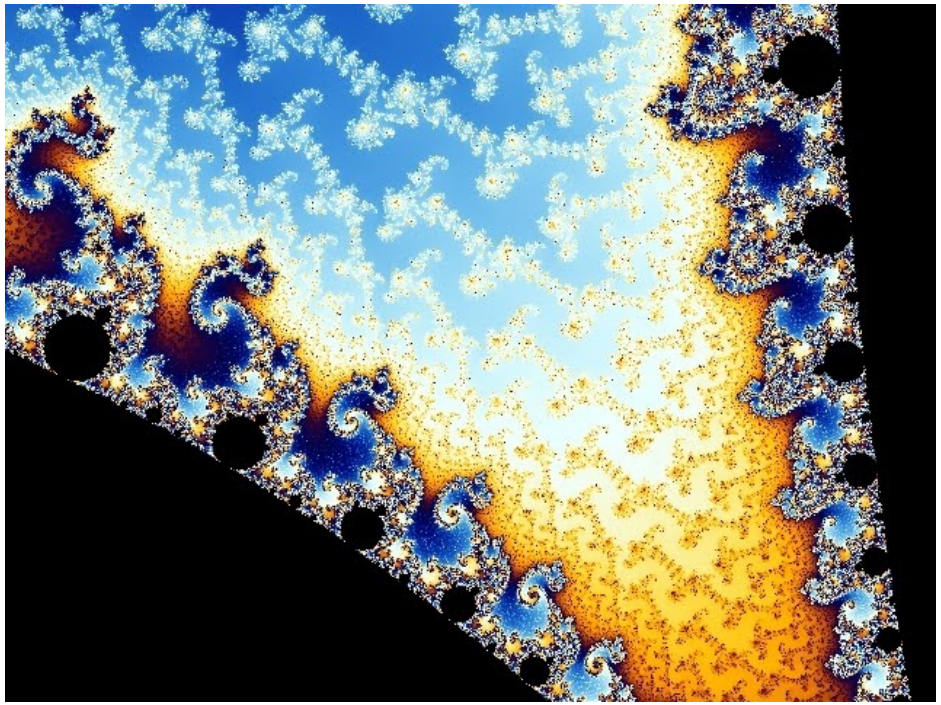
Overview

- In everyday terms?
  - It's just a collection of numbers that make a fractal!
- Black area = in the set
  - Everything else = not an element of the set
- Color due to speed of divergence (see whiteboard)



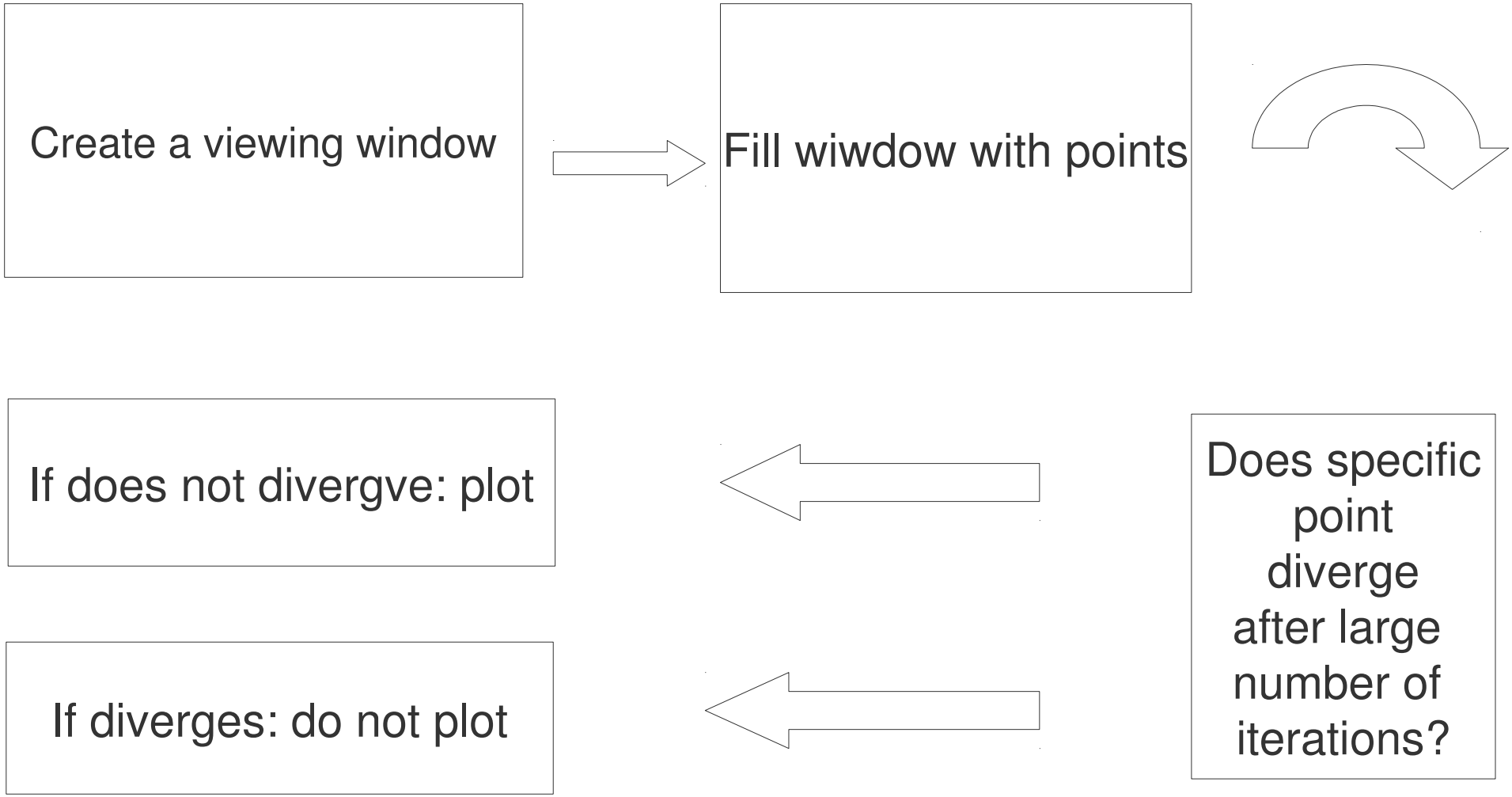
Problem/Goals

- Three parts to this project:
  - Initial plotting of the set (with and without color)
  - Dynamic zooming on the seahorse valley (shown)
  - Calculation of area using Monte Carlo methods

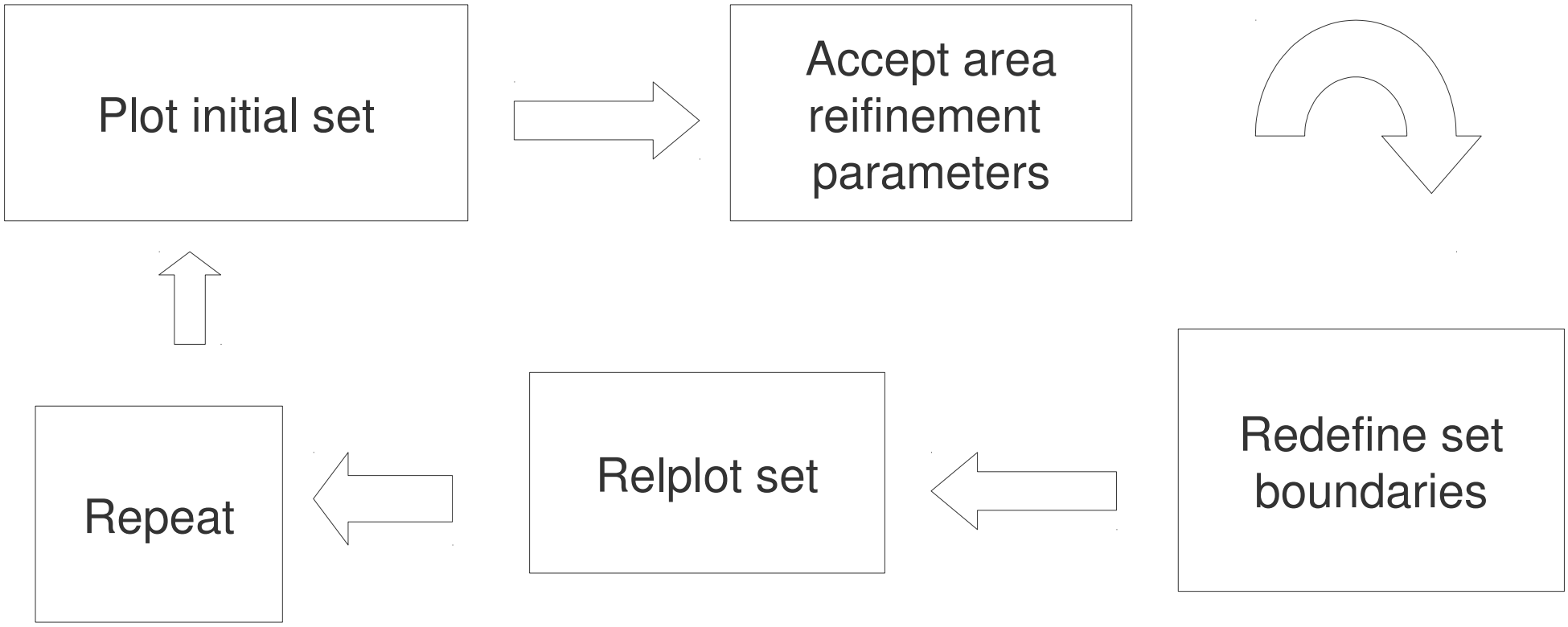


$$A = \pi\left(1 - \sum_{n=0}^{\infty} n\left(b_n\right)^2\right)$$

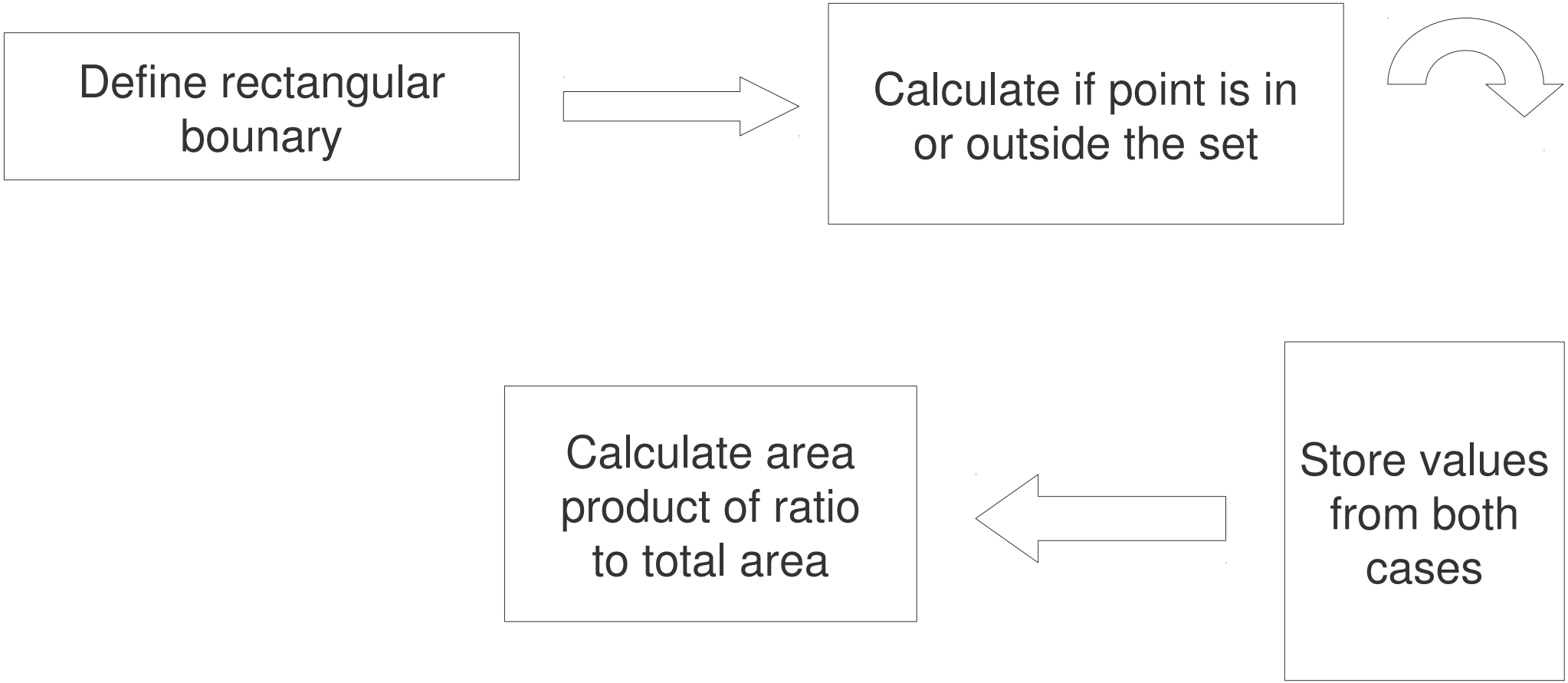
Part 1: Plotting Structure



Part 2: Dynamic Zooming Structure



Part 3: Area Calculation Structure



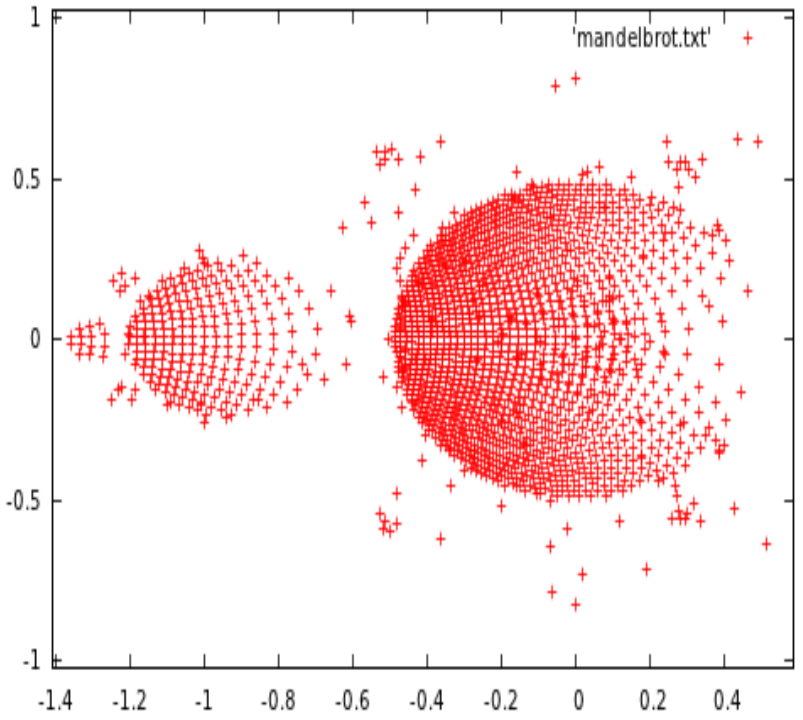
Part 1: Code

```
#include <iostream>

int main () {

        double min_real = -2.0;
        double max_real = 1.0;
        double min_im = -1.0;
        double max_im = min_im + (max_real- min_real) * height/width;
        double real_factor = (max_real - min_real) * width;
        double im_factor = (max_im - min_im) * height;
        int iterations = 50;

        for(unsigned  y = 0; y < height; y++) {
                double c_im = max_im - y * im_factor;
                for(unsigned x = 0; x < width; x++) {
                        double c_re = min_real + x * real_factor;
                        double z_re = c_re;
                        double z_im = c_im;
                        bool is_it_inside = true;
                        for(unsigned n = 0; n < iterations; ++n) {
                                double mag_re_part = z_re * z_re;
                                double mag_im_part = z_im * z_im;
                                if(mag_re_part + mag_im_part > 4.0) {
                                        is_it_inside = false;
                                        break;
                                }
                                z_im = 2.0 * z_re * z_im + c_im;
                                z_re = mag_re_part - mag_im_part + c_re;
                        }
                                if(is_it_inside) {
                                fout << z_re << "\t\t" << z_im << endl;
```



Part 1.1: Pseudocode

```
int main() {

    //define all variables for plotting the set


    //define a vector that assigns color
    //to divergence after a number of inerations

        for(unsigned  y = 0; y < height; y++) {
        double c_im = max_im - y * im_factor;
        for(unsigned x = 0; x < width; x++) {
            double c_re = min_real + x * real_factor;
            double z_re = c_re;
            double z_im = c_im;
            bool is_it_inside = true;


            for(unsigned n = 0; n < iterations; ++n) {
                double mag_re_part = z_re * z_re;
                double mag_im_part = z_im * z_im

            if(mag_re_part + mag_im_part > 4.0) {
                        is_it_inside = false;
                        plot(point, color = n)
                        break;
                }
                z_im = 2.0 * z_re * z_im + c_im;
                z_re = mag_re_part - mag_im_part + c_re;
            }
            if(is_it_inside) {
                plot(plot point, color = black)
```

Note: This code is quite similar to that of the previous slide.

Part 2: Pseudocode

```
double function calc_mandelbrot(....., ....., ....) {

        //make a function that calculates set
        }

int main() {

        double re_min = -2;
        double re_max = 1;
        double im_min = -1;
        double im_max = 1;
        double step = 0.001;
        int num_frames = 1000;

        for(int i = 0; i < num_frames; i++) {

        calc_mandelbrot(viewing parameters)

                for(double x = re_min ; x < re_max; x = x + step) {
                for(double xy= re_min ; y < re_max; y = y + step) {
                        plot(x, y)
                }
                }

        //redefine viewing parameters
```

Project Status

- Part 1: Inital plotting done
  - Need to add color
- Part 2: Not started
- Part 3: Not started