WXML Final Report: Shapes of Julia Sets

Malik Younsi, Peter Lin, Chi-Yu Cheng, Siyuan Ni, Ryan Pachauri and Xiao Li

Spring 2017

Abstract

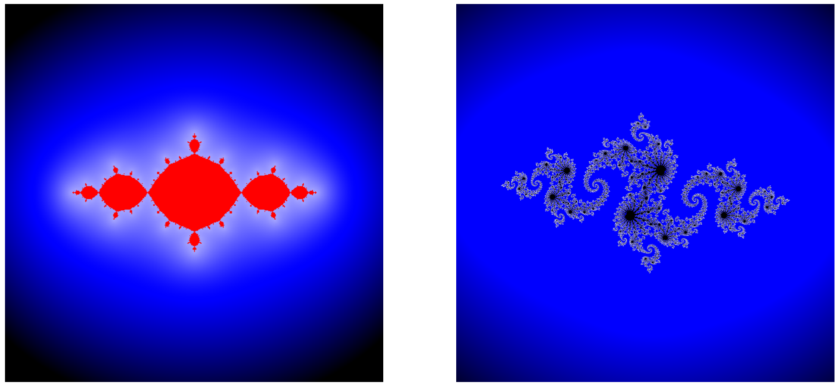
In general, the complex graphical representation of Julia set is generated by a simple polynomial after iterations. The goals of our project, shapes of Julia sets, are to find a better computational algorithm for plotting filled Julia set and to approximate any set of disjoint Jordan curves on the complex plane by plotting the Julia set of polynomial.

1 Introduction

The filled Julia set of a polynomial is defined by

$$K(P) = \{z \in C : P^m(z) \not\to \infty \text{ as } m \to \infty\}$$

where $P^m(z)$ is the nth iterate of polynomial applied to z for m a natural number. The Julia set of $f(z) = z^n + c$ is the set of those complex number z, such that $f^n(z)$ does not approach infinity as n goes to infinity. For example, suppose that we have a quadratic polynomial $f(z) = z^2 + c$, where c is a complex parameter, then we plug initial value z_0 to the polynomial and get z_1 such that $z_1 = f(z_0) = z_0^2 + c$, and plug z_1 to the polynomial and get z_2...... We keep doing this process n times as $n \to \infty$, if z_n converge to infinity, then z_0 is not in the filled Julia set of $f(z)$. Otherwise, it's in the Julia set. As the parameter c varies, the filled Julia set varies as well, hence we can produce the different shapes of Julia set. (Figure 1)



$c = (-1, 0)$ $\qquad$ $c = (-0.8, 0.156)$

FIGURE 1. The shapes of Julia set of polynomial change as c changes

2 Plotting Julia sets

2.1 The obvious method

The "obvious method" to determine if a complex number belongs to a Julia Set is to iterate points in the complex plane to the polynomial n times and check when $n \to \infty$, will the result converge to infinity as well.

This method is straight forward, but at the same time, the drawback of it is also easy to see: in real life, we can't iterate a point infinite times, we have to choose a certain number, for example, in our program, 100 times, and set a threshold, in our program, 10, and check after 100 times of iterations, will the magnitude of the result greater than 10. But this will bring us a new issue: there exists some number that after 100 iterations, they are still less than 10 although they actually will converge to infinity at last which means they are not part of the Julia set, but in our program, we will treat this kind of points as part of it. And this would make our plot lack of accuracy.

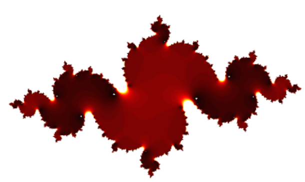


FIGURE 2. $z^2 - 0.8 + 0.156i$, $c = (-0.8, 0.156)$, by obvious method

FIGURE 3. $z^2 - 1.2 + 0.156i$, $c = (-1.2, 0.156)$, by obvious method

2.2 The distance estimation method

Distance estimation method (DEM) is a powerful technique for plotting Julia sets. Basically, DEM is based on various behaviors after iteration of polynomial. For each initial value of z_0 and polynomial $f(z) = z^2 + c$, we can form a sequence z_n such that $z_1 = f(z_0), z_2 = f(z_1), z_3 = f(z_2)$ââ$z_n = f(z_{n-1})$, where n is the number of iteration.

The process of DEM:

The sequence z_n converges to the limit radius r, where r is a small positive real number, this means all the points of the sequence z_n are close to z_0 with a sufficiently small neighborhood. In this case, we say that z_0 is in the Julia set and we label it as 0.

The sequencez_n diverges to the limit radius r, then we compute and iterate its derivative ${z_n}' = 2z_{n-1}{z_{n-1}}'$. If the magnitude of ${z_n}'$ is greater than or equal to defined threshold, then we say that z_0 is close to the Julia set and we label it as -1. Otherwise, we estimate the distance of z_0 by the following equation

$$d = 2\frac{|z_n|}{|{z_n}'|}\log|z_n|$$

and label it as 1.

Then we set the different colors depend on its label.

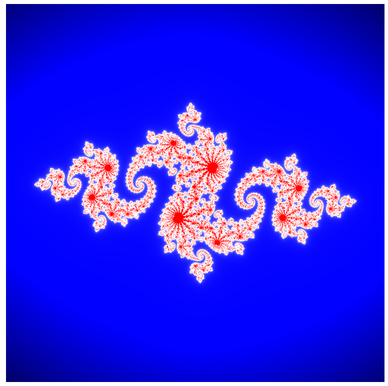


FIGURE 4. $z^2 - 0.8 + 0.156i$, $c = (-0.8, 0.156)$, by DEM

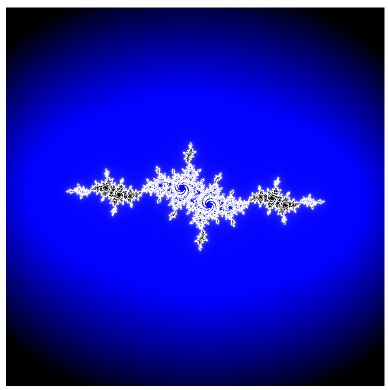


FIGURE 5. $z^2 - 1.2 + 0.156i$, $c = (-1.2, 0.156)$, by DEM

3 Approximation

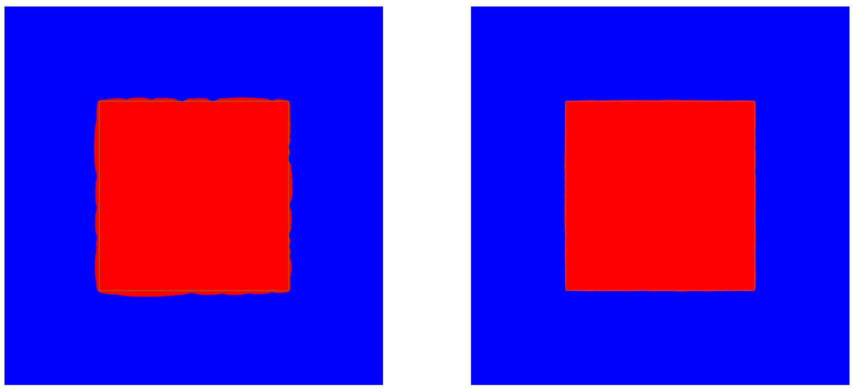
From a paper our mentor M.Younsi co-wrote, entitled "Fekete Polynomials and Shapes of Julia Sets", we have a polynomial function:

$$P_{n,s}(z) := z\frac{e^{-ns/2}}{cap(E)^n}\prod_{j=1}^{n}(z - z_j)$$

In this function, E is the set of all points of the given Jordan curves, z_js are called leja points, and is defined as following: consider z_1 to be a random point in E, then z_2 is the point which could maximize the value $|z_2 - z_1|$, and z_3 is the point that maximizing $|z_3 - z_1||z_3 - z_2|$, n is the number of leja points, so for z_n, we want to find the point that could maximize $|z_n - z_1||z_n - z_2|\cdots|z_n - z_{n-1}|$. $cap(E)$ is called the logarithmic capacity, it is calculated by

$$\lim_{n\to\infty}(\prod_{j=1}^{n}|z_{n+1} - z_j|)^{1/n}$$

and s is any small positive number. Here, n and s are the parameters that determine the accuracy of our approximation. Our first guess is that to produce more accurate approximation, we should set the n to be large integer and s to be small positive real number $\left(\text{we normally set s} = \frac{1}{n} \text{ for computational purpose}\right)$ because that if we have a larger n, this is, we have more leja points, then we have more understanding of the Jordan curves. Thus, we could produce more accurate approximations. (Figure 6)

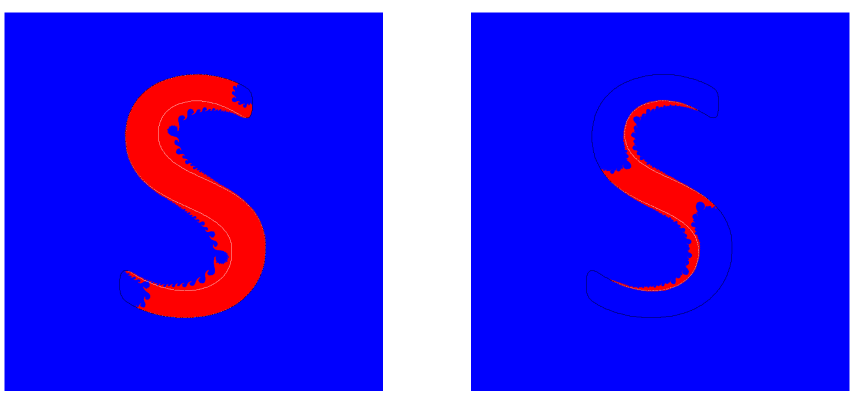


$n = 100$ and $s = 0.0100$ $\qquad$ $n = 600$ and $s = 0.00167$

FIGURE 6. An approximation of a square with different values of n

3.1 Changing n

At first glance, a higher number of leja points would mean that we should arrive at a more accurate picture. However, we found that it was possible to have too many leja points.



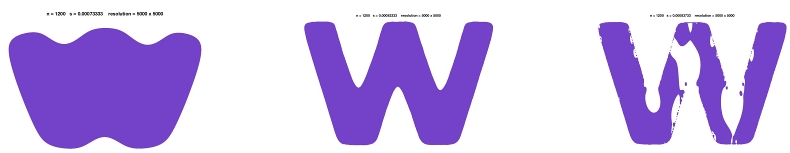
$n = 1200$ and $s = 0.0008333$ $\qquad$ $n = 1800$ and $s = 0.000555$

FIGURE 7. An approximation of a letter with different values of n

As you can see with the above pictures, with more leja points (and keeping the $n : s$ ratio the same), fewer points in the complex plane are in the Julia set. Many points that we would have wanted in the Julia set are not included.

3.2 Changing s

In general, we set $s = \frac{1}{n}$ to produce a better approximation. We found if we slightly change the value of s, then we will get the dramatically different results. (Figure 7)



$s < \frac{1}{n}$ $\qquad$ $s = \frac{1}{n}$ $\qquad$ $s > \frac{1}{n}$

FIGURE 8. An approximation of a letter with different values of s

4 Critical Points

By definition, a critical point of a differentiable polynomial is any value in its domain where its derivative is 0. We want to find and plot the critical points of polynomial to see the behavior of the critical points. It is a theorem that states all the critical points should be inside the filled Julia set if the shape has only one connected component. In Figure 8, we can clearly see that 100 critical points (red dots) are inside the boundary, this is, they are all inside the filled Julia set. However, the theorem does not hold for the multiple connected components. In Figure 9, we can see that not all critical points are inside the boundary of each component, and some of them are outside each components.

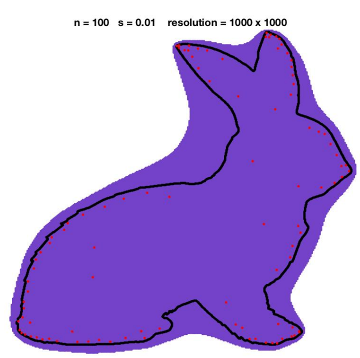


FIGURE 9. 100 critical points of a rabbit

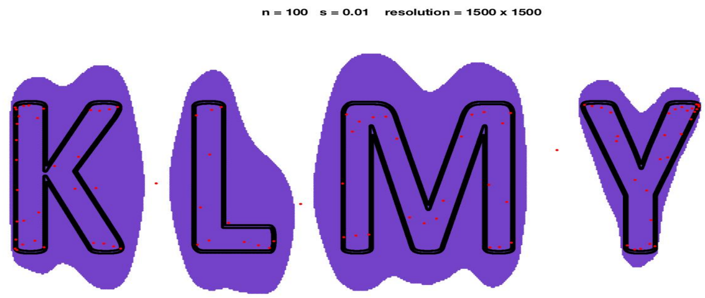


FIGURE 10. 100 critical points of letters

5 Fractal pattern

One interesting thing about Julia set is that it is a fractal that exhibit self-similarity, this is because that any filled Julia set either has a unique component or infinitely many component which would result in that the whole has the similar shape as its parts. Speaking intuitively, when we zoom in around the edge of fractal, we should still be able to see the similar shapes as the outlier appear to be. Below is an example showing this feature.

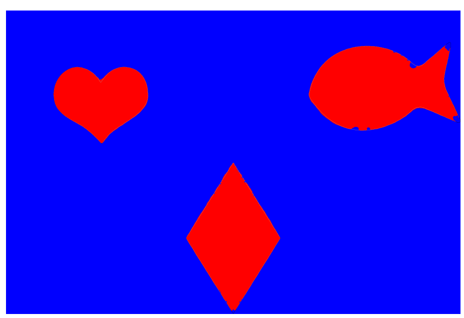


FIGURE 11. Original shape

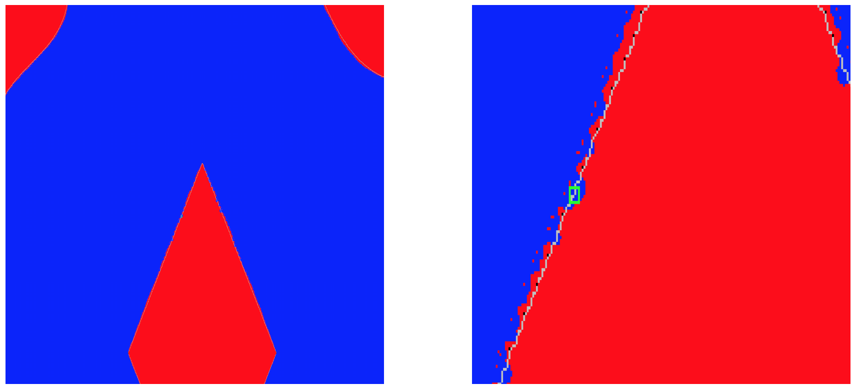
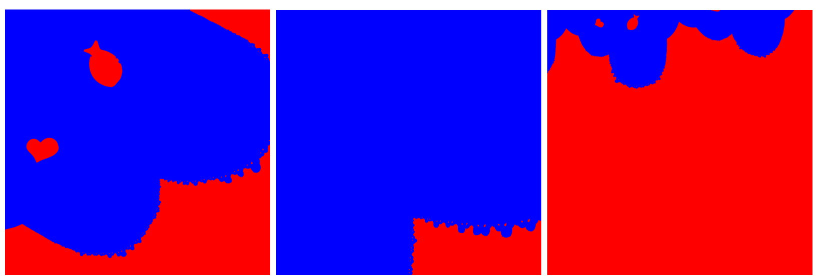


FIGURE 12.

The above images are simply to show where we evaluated our polynomial (i.e. these are zoomed in versions Figure 11, but they are not exact results of evaluating our polynomial). The green box is where we decided to evaluate our polynomial.



200x zoom 400x zoom 2000x zoom

FIGURE 13.

The above images are the result of evaluating our polynomial on a smaller range of complex values at a higher resolutions. As you can see, there are copies of shapes from the original image albeit they are skewed. The only ones we can see, however, are a heart and a fish. We do not see zoomed in version of the diamond because we are along the edge of the original diamond. A point of interest moving forward would be if we can see tiny diamonds along the edge of the zoomed in heart/fish.