8-2003

Fractal Images Generated by Newton's Method."

Jennifer Corte
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Jennifer Corte entitled "Fractal Images Generated by Newton's Method."." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Jerry Dwyer, Major Professor

We have read this thesis and recommend its acceptance:

G. Samuel Jordan, Suzanne Lenhart

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Jennifer Corte entitled "Fractal Images Generated by Newton's Method." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Jerry Dwyer
Major Professor

We have read this thesis
and recommend its acceptance:

G. Samuel Jordan

Suzanne Lenhart

Accepted for the Council:

Dr. Anne Mayhew
Vice Provost and Dean of
Graduate Studies

(Original signatures are on file with official student records.)

Fractal Images Generated by Newton's Method

A Thesis

Presented for the

Masters of Science Degree

The University of Tennessee, Knoxville

Jennifer Corte

August 2003

Acknowledgments

I thank Dr. Dwyer for serving as my major professor and for his help in completing this thesis. I thank Dr. Jordan and Dr. Lenhart for serving on my committee, for making valuable suggestions and for being excellent teachers. I especially thank my parents for their support and belief in me. Many people in the Math department have encouraged and helped me including, but not limited to: Jason Howard, Pam Armentrout, Mike Saum, and John Iskra.

Abstract

We investigate the behavior of Newton's Method for finding roots applied to complex-valued functions of complex variables. This requires an analysis of iteration of rational functions. The fractal nature of Newton's Method in the complex plane gives us intricate and beautiful images. By investigating select functions we attempt to generalize a pattern of behavior.

Contents

1 Some Basic Ideas in Complex Analysis **1**

2 Complex Dynamical Systems **4**

1 Orbits 4

2 Fixed Points and Cycles 9

3 Julia Sets 10

4 Newton's Method 11

3 Matlab Code **14**

4 Iteration of Rational Functions **17**

1 The Fixed Point at Infinity 17

2 Newton's Method Applied to the Family of Functions $\mathbf{f(z) = \frac{az^n - b}{cz^n - d}}$ 20

2.1 The $n = 1$ Case 21

2.2 The $n = 2$ Case 23

2.3 The $n = 3$ Case 29

5 Compositions of Functions 34

6 Closing Remarks 43

Bibliography 45

Appendix 48

1 Code 49

2 Input Arguments 56

3 Coding Newton's Method 57

4 Plotting the Figures 58

Vita 60

List of Figures

1.1 The Mapping $w = z^2$ 2

2.1 Convergent Orbit for $g(x) = x^3$ 6

2.2 Divergent Orbit for $g(x) = x^3$ 6

2.3 A 2-cycle for $x^2 - 1$ 8

2.4 Newton's Method for $g(x) = \frac{1}{4}(e^x - 1)$ 12

2.5 Newton's Method for $f(z) = z^2 + 1$ 12

4.1 $f(z) = \frac{z-1}{z+i}$ 22

4.2 $f(z) = \frac{3z+6-9i}{2z+4+i}$ 22

4.3 $f(z) = \frac{2z-10-6i}{z-9-3i}$ 23

4.4 $f(z) = \frac{z^2-4}{z^2-1}$ 24

4.5 $f(z) = \frac{z^2+4}{z^2+9}$ 25

4.6 $f(z) = \frac{z^2-4}{z^2+1}$ 25

4.7 $f(z) = \frac{3z^2+5i}{z^2-30i}$ 27

4.8 $f(z) = \frac{z^2+8i}{z^2-9.6+5.12i}$ 27

4.9 $f(z) = \frac{z^2+8i}{z^2-12+16i}$ 28

4.10 $f(z) = \frac{z^3-8}{z^3-125}$ 29

4.11 $f(z) = \frac{z^3-8}{z^3+125}$ 30

4.12 $f(z) = \frac{z^3-1}{z^3+125i}$ 32

4.13 $f(z) = \frac{z^3-125}{z^3-8}$ 32

4.14 $f(z) = \frac{z^3+125}{z^3-8}$ 33

5.1 $f(z) = log(z^2 - 1)$ 35

5.2 $f(z) = sin(log(z))$ 37

5.3 Zoomed in picture for $f(z) = sin(log(z))$ 39

5.4 Zoomed out picture for $f(z) = sin(z^{\frac{1}{3}})$ 39

5.5 $f(z) = sin(z^{\frac{1}{3}})$ 40

5.6 Zoomed in picture for $f(z) = sin(z^{\frac{1}{3}})$ 40

5.7 $f(z) = sin(\pi e^z)$ 42

5.8 $f(z) = sin(log(sin(z)))$ 42

Chapter 1

Some Basic Ideas in Complex Analysis

Complex-valued functions of the complex variable $z = x + iy$ assign to each z in the domain exactly one complex number $w = f(z)$. The function $f(z) = f(x + iy)$ can be split into its real and imaginary parts, or $f(x+iy) = u(x, y) + iv(x, y)$, where u and v are each real-valued functions. A function $f(z)$ can be viewed as a mapping from its domain in the complex plane onto its range in another copy of the complex plane. For example, consider the function $f(z) = z^2 = x^2 - y^2 + i2xy$. The left side of Figure 1.1 shows a portion of the domain of f. The right side shows the image of the horizontal and vertical lines of the domain under the mapping $w = f(z) = z^2$.

Complex-valued functions of complex variables are usually simply referred to as complex functions. Only complex functions which

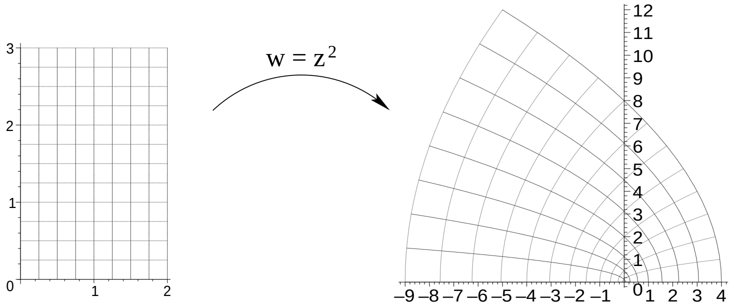

Figure 1.1: The Mapping $w = z^2$

exhibit certain characteristics will be considered here. First the functions must be differentiable at most points in the complex plane.

Definition 1 *Suppose a complex function f is defined at a point z_0 and at all points close to z_0 in the complex plane. Then f is* ***differentiable*** *at z_0 if $\lim_{z \to z_0} \frac{f(z)-f(z_0)}{z-z_0}$ exists and is finite. In that case the limit is denoted by $f'(z_0)$.*

A stronger condition is that the functions be analytic at most points in the complex plane.

Definition 2 *A complex function is* ***analytic*** *at a point z_0 if there is an $\epsilon > 0$ such that $f^{'}(z)$ exists for all z in the ϵ-neighborhood of z_0, i.e., for all z with $|z - z_0| < \epsilon$. A funtion that is analytic at every point in a region R is called* ***analytic on*** *R.*

An even stronger condition than analytic at most points is that the functions be entire.

Definition 3 *A complex function is called* ***entire*** *if it is analytic on the whole complex plane.*

Most functions considered in this paper are analytic at all but a finite number of points.

We will be using a numerical method, Newton's Method, to determine the roots, or zeros, of the complex functions and to plot colored images.

Definition 4 *The complex number r is a* ***zero (or root)*** *of the function $f(z)$ if it is a solution to the equation $f(r) = 0$.*

Several of the functions investigated have singularities which result in rather interesting images.

Definition 5 *A point α is called a* ***singular point (or singularity)*** *of f if f is not analytic at α. The singular point α is called* ***isolated*** *if, in addition, there is some neighborhood of α throughout which f is analytic except at the point itself.*

Definition 6 *If $z = a$ is an isolated singularity of f and if $\lim_{z\to a}|f(z)| = \infty$, then a is a* ***pole*** *of f.*

Chapter 2

Complex Dynamical Systems

Many aspects of dynamical systems can be visualized very nicely when only dealing with the reals and real-valued functions. This visualization is much more difficult, if not impossible, when dealing with complex functions. Although this paper focuses on complex dynamics, an introduction to real dynamics is conducive to understanding the concepts involved in both the real and the complex cases. In the following sections, concepts and definitions will be demonstrated using real dynamics where applicable.

1 Orbits

A sequence $\{z_k\}$ can be generated from a function $f(z)$ and a single complex value z_0 by iteration. Starting with $z_0 = z_0$, the sequence of iterates is generated as follows: $z_1 = f(z_0), z_2 = f(z_1), z_3 = f(z_2), \ldots.$

The points $\{z_k\}_{k=0}^{\infty}$ are the iterates of f generated by z_0. This set of iterates is also called the orbit of z_0 generated by f. The starting value z_0 is often called the initial seed, or just the seed. $f(z) = z$. The behavior of these orbits as $k \to \infty$ leads to very interesting dynamics.

If the orbits are bounded they may never converge. If they do converge, they may converge either to a single functional value (a z for which $f(z) = z$ called a fixed point) or to a repeating periodic cycle of values. Otherwise they diverge to ∞.

To illustrate these orbits, consider the real function $g(x) = x^3$. The fixed points occur where $g(x) = x$, therefore the points $-1, 0$, and 1 are the fixed points. Whether orbits converge to or diverge from these points depends on the initial seed value. Figure 2.1 shows that the orbit of a seed x_0, with $0 < x_0 < 1$, will converge to the fixed point $(0, 0)$. We therefore call the point $x = 0$ an attracting fixed point.

However, if the seed is x_0 such that $1 < x_0$, as in Figure 2.2, then the orbit diverges from the fixed point $(1, 1)$. In this case $x = 1$ is called a repelling fixed point.

Consider the family of quadratic functions of the form $f(z) = z^2 + c$. For $c = 2$ and $z_0 = 1 + i$, the sequence of iterates is:

$z_0 = 1 + i$,

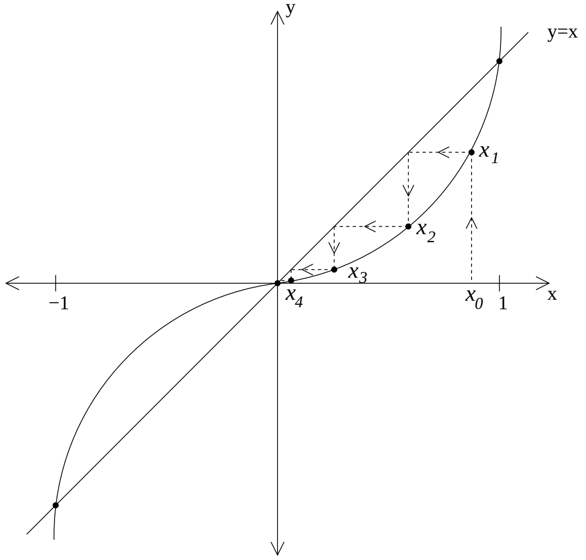

Figure 2.1: Convergent Orbit for $g(x) = x^3$

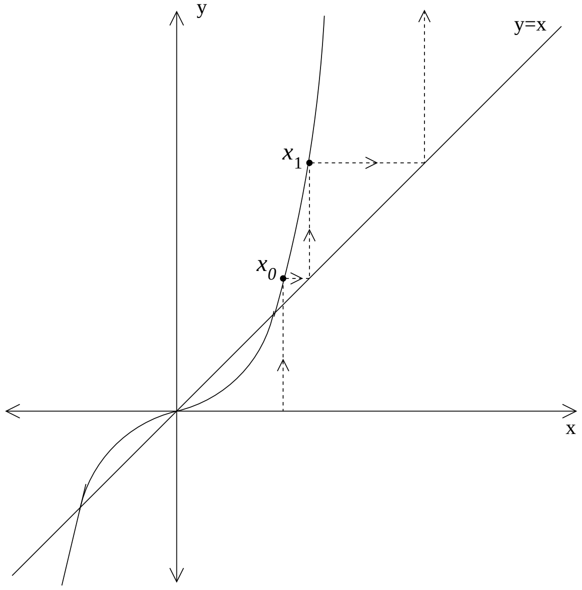

Figure 2.2: Divergent Orbit for $g(x) = x^3$

$z_1 = (1+i)^2 + 2 = 2 + 2i,$

$z_2 = (2+2i)^2 + 2 = 2 + 8i,$

$z_3 = (2+8i)^2 + 2 = -58 + 32i,$

$z_4 = (-58+32i)^2 + 2 = 2342 - 3712i, ...$

which is a diverging sequence.

For $c = -\frac{1}{4}$ and $z_0 = 0$, however, the beginning of the sequence of iterates (to four decimals) is

$z_0 = 0$

$z_1 = 0.25$

$z_2 = 0.3125$

$z_3 = 0.3477$

$z_4 = 0.3709$

which converges to the fixed point $z = 0.5$.

For $c = -1$ and $z_0 = 1$, the iterates converge to a 2-cycle: $z_0 = 1, z_1 = 0, z_2 = -1, z_3 = 0, z_4 = -1, ...$ When the orbits converge to a periodic cycle, the cycle is called an n-cycle, where n is the number of distinct values in the cycle.

This 2-cycle can be visualized in the real case with $g(x) = x^2 - 1$ and a seed of $x_0 = 1$. Just as with the complex function above, the sequence of iterates immediately converges to the 2-cycle with $x_1 = 0$, $x_2 = -1$ as the two points in the cycle (see Figure 2.3).

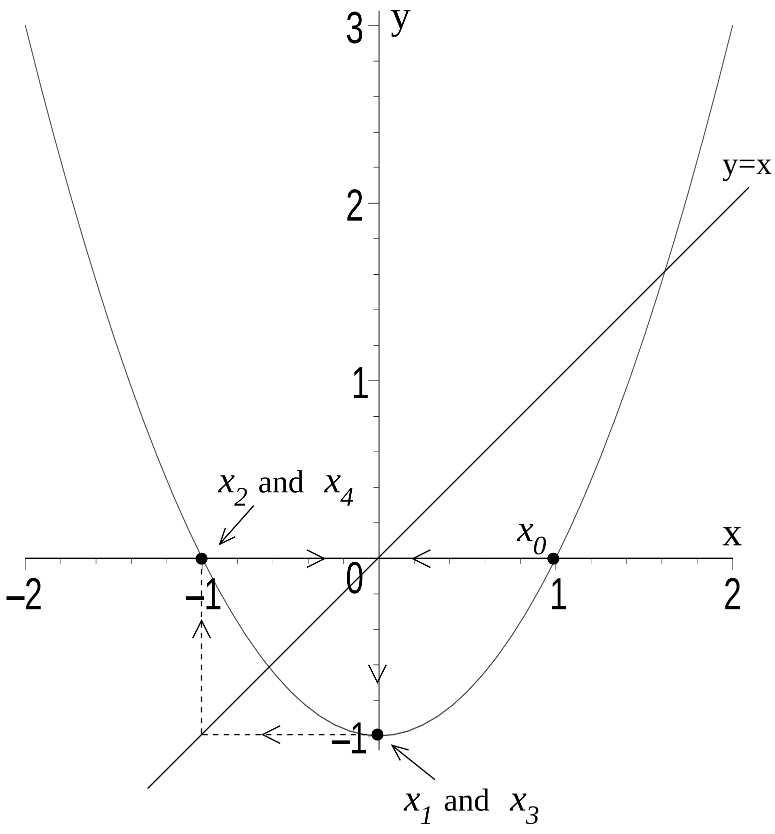

Figure 2.3: A 2-cycle for $x^2 - 1$

2 Fixed Points and Cycles

Definition 7 *The orbit of z_0 is **periodic of period** n, if $z_n = z_0$ and $z_0 \neq z_i$ for $0 < i < n$. Such an orbit is called an n-cycle and the points in the cycle are called periodic points. A point z_0 is called **preperiodic** if for $k > 0$, the k^{th} iterate of z_0 is equal to a periodic point in an n-cycle, but z_0 itself is not a point in that cycle.*

Fixed points and n-cycles can be attracting, repelling or neutral. A fixed point is attracting if there is a disk around the point such that a seed inside the disc leads to iterates that converge to the fixed point. A fixed point is repelling if no matter how close the seed is to the fixed point, iterates of that seed will diverge from the fixed point.

Definition 8 *If z_0 is a fixed point of an analytic function, f, then the number $\lambda = f'(z_0)$ is called the **multiplier** of f at z_0. The fixed point is classified dependent on λ as follows:*

*i) **attracting** if $|\lambda| < 1$,*

*ii) **superattracting** if $\lambda = 0$,*

*iii) **repelling** if $|\lambda| > 1$,*

*iv) **neutral** if $|\lambda| = 1$.*

3 Julia Sets

Some interesting objects have arisen simply from the study of several families of functions, most notably the quadratic family of functions $P_c(z) = z^2 + c$. One such group of objects is composed of Julia sets.

Definition 9 *Let $f : \mathbf{C} \to \mathbf{C}$ be analytic. Let $\mathbf{K(f)}$ be the set of all seeds whose orbits are bounded. The **Julia set** of f, $\mathbf{J(f)}$, is the boundary of $\mathbf{K(f)}$.*

*The **filled Julia set** of f is the union of $\mathbf{J(f)}$ and $\mathbf{K(f)}$.*

For example, consider the mapping $P_0(z) = z^2$. For all z_0 with $|z_0| \leq 1$, the orbits either stay on the unit circle, or converge to 0, i.e. are bounded. For $|z_0| > 1$, the orbits diverge. Therefore, $K(P_0)$ is the closed unit disc, $\overline{D}_1(0)$, and the Julia set, $J(P_0)$, is the unit circle $C_1(0)$.

For both Julia sets and filled Julia sets, seeds in each of the sets will have orbits which remain in that set. The pictures presented in this paper are images of filled Julia sets for the various functions we examine.

4 Newton's Method

Newton's Method is an algorithm for finding the roots or zeros of a function $f(z)$. The algorithm uses successive iteration of the function

$$N(z) = z - \frac{f(z)}{f'(z)}$$

Convergence to a root using Newton's Method can therefore be thought of as convergence to a fixed point of $N(z)$.

Although we can not graphically represent how Newton's Method works for complex functions, one can get an idea by looking at the method for real functions. Consider the real-valued function $g(x) = \frac{1}{4}(e^x - 1)$ with root $x = 0$. We will let $x_0 = 1$ be our initial guess at the root $x = 0$. In order to determine the next point, Newton's Method follows down the line tangent to g at the point $(x_0, f(x_0))$ until it reaches the x axis. The point where the tangent line intersects the x axis is the next approximation. This process is repeated until the iterates either converge to the root of g or diverge. The iterates to four decimal places are $x_0 = 1$, $x_1 = 0.3679$, $x_2 = 0.0601$, $x_3 = 0.0018$, $x_4 = 0.0000$,... which are clearly converging to the root. Figure 2.4 illustrates Newton's Method for a real valued function converging to a root at $x = 0$.

To visualize the convergence of Newton's Method for complex functions, see Figure 2.5. The first few iterates of $N(z)$ for the

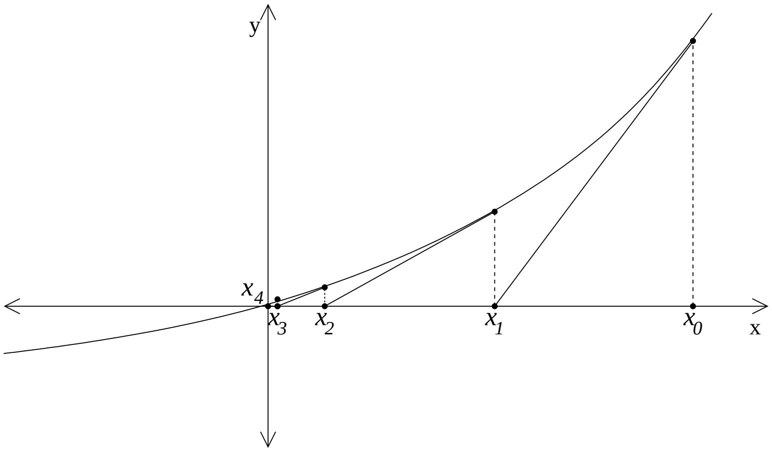

Figure 2.4: Newton's Method for $g(x) = \frac{1}{4}(e^x - 1)$

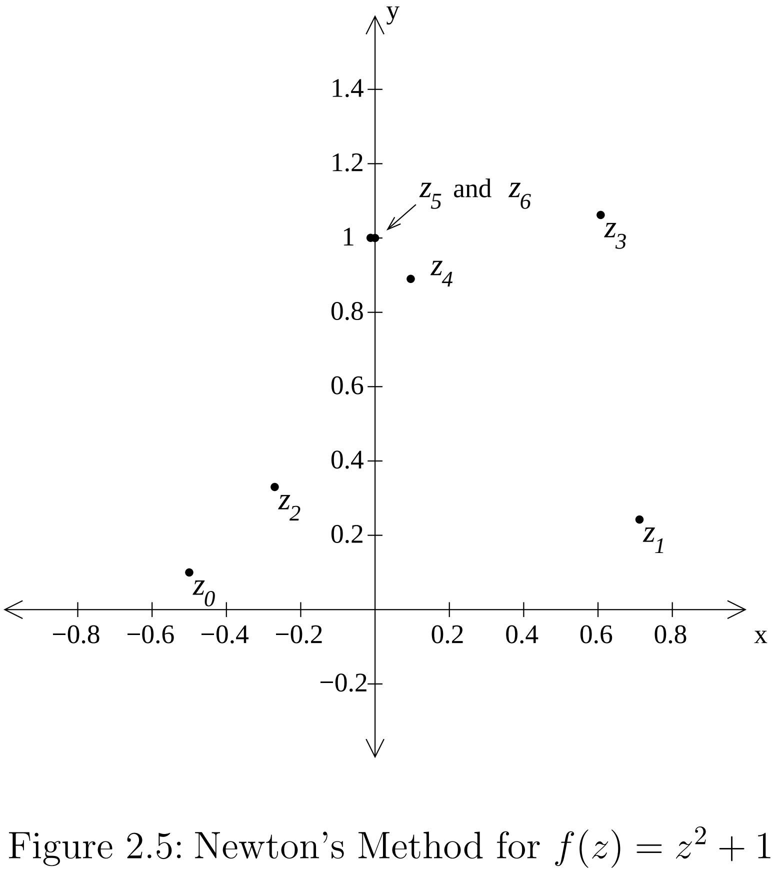

Figure 2.5: Newton's Method for $f(z) = z^2 + 1$

function $f(z) = z^2 + 1$ with initial $z_0 = -0.5 + 0.1i$ are plotted. The orbit of $z_0 = -0.5 + 0.1i$ converges to the root $z = i$.

Chapter 3

Matlab Code

The purpose of our Matlab code is to create a visual representation of regions of the complex plane which converge to the roots of complex functions under iterations of Newton's Method. The code allows the user to choose the coloring scheme for this visual representation. The first argument determines what type of coloring scheme will be used in the figure. An input value of 0 results in coloring each point according to which root its orbit converges; an input of 1 results in coloring according to rate of convergence; and an input of 2 will result in both color schemes, each in a different figure.

The code takes either user-input or default window boundaries and creates a matrix grid of complex values filling the window. The meshsize or dimension of this grid can also be either an input or default. The default window boundaries are from -1.5 to 1.5 on both the real and imaginary axes. The default meshsize is 100, which creates a 100 by 100 grid of points in the window. The user must also enter the function and its derivative into the code before running it. The matrix initially created is essentially a grid of points which will be seed values for iterations of Newton's method.

The code performs iterations of the method on each seed in the grid. Those seeds which diverge are discarded while those which converge are stored into an output matrix. The orbit of a seed is determined to converge if the distance between the current point in the orbit and the previous point in the orbit is less than the programmed radius of convergence, R. Three matrices are simultaneously updated throughout this process. The original seed values which converge are stored in one matrix, the values that they converge to (i.e., the roots) are stored in a second output matrix, and the number of iterations it took for each point to converge is stored in the third matrix. Once the computations of Newton's Method have finished, the output matrix of converged-to values (roots) is sorted to sift out the distinct roots. If values in the output matrix are within a set tolerence of each other, they are considered equal. This sorting process results in a matrix of distinct roots.

In order to color according to the roots, seed values which converged to the distinct roots are plotted assigning a different color for each root. The seeds which did not converge are not plotted, so they remain white in the figure. To color according to the rates of convergence, Matlab's pink colormap is used. Roots which converge the quickest are colored a dark pink/brown, while roots that take the longest to converge are colored a very pale shade of the same color.

Examples of functions and their corresponding figures will be presented in this paper. To avoid repetitive wording, the meshsize of each function can be assumed to be 500 unless otherwise mentioned. The x and y boundaries will not be written in the text as these may be seen by looking at the axes of each figure. Please note that the axes in the figure are box axes which means that they do not intersect at the origin. Many, but not all, of the functions will have images of both color schemes.

This code is written using Matlab Student Version 5.3. Some changes may need to be made for it to run properly on other versions of Matlab. Please refer to the appendix for more details about the code and about improvements that were made in its efficiency.

Chapter 4

Iteration of Rational Functions

1 The Fixed Point at Infinity

Let $r(z) = \dfrac{p(z)}{q(z)}$ where $p(z) = a_n z^n + a_{n-1} z^{n-1} + ... + a_0$ and $q(z) = b_m z^m + b_{m-1} z^{m-1} + ... + b_0$ are coprime polynomials, that is, they have no common zeros. Newton's Method applied to this rational function is itself iteration of the rational function

$$N(z) = z - \frac{r(z)}{r'(z)} = z - \frac{p(z)q(z)}{q(z)p'(z) - p(z)q'(z)}.$$

The zeros of r are the fixed points of N. Since we are interested in the zeros of $r(z)$, it is important to investigate the fixed points of $N(z)$. Of particular interest is whether or not the fixed points of N are attracting, repelling or neutral. If they are attracting, then Newton's Method will converge to the roots of $r(z)$. Again, this classification is determined by the value of the multiplier, which we have already defined for the finite fixed point z_0 as $f'(z_0)$, or in this case, $N'(z_0)$.

Some functions have ∞ as a fixed point. For example, let $f(z) = z+1$. Clearly $f(\infty) = \infty$ making ∞ a fixed point of f. The multiplier at ∞, however is not $f'(\infty) = lim_{z \to \infty} f'(z)$. The multiplier of a fixed point, z_0, of an arbitrary rational function $R(z) = \dfrac{\alpha_k z^k + \alpha_{k-1} z^{k-1} + ... + \alpha_0}{\beta_j z^j + \beta_{j-1} z^{j-1} + ... + \beta_0}$, written $m(R, z_0)$ is

$$m(R, z_0) = \begin{cases} R'(z_0), & \text{if } z_0 \neq \infty \\ \dfrac{1}{R'(z_0)}, & \text{if } z_0 = \infty \end{cases} \tag{1}$$

where

$$R'(\infty) = \begin{cases} \dfrac{\alpha_k}{\beta_j}, & \text{if } k = j+1 \\ \infty, & \text{if } k > j+1 \end{cases} \tag{2}$$

Newton's Method on the rational function r is iteration of the function $N(z) = z - \dfrac{r(z)}{r'(z)}$ which is a rational function itself when in reduced form. Assuming the leading terms do not cancel, the degree of the numerator of N is $m+n$ while the degree of the denominator of N is $m+n-1$. Therefore, we have the case where the multiplier at ∞, $m(N, \infty)$, is equal to the leading coefficient of the denominator divided by the leading coefficient of the numerator. We will keep track of the leading coefficients in the numerator and denominator of $N(z)$ in order to determine this ratio. Recall that $p(z) = a_n z^n + a_{n-1}z^{n-1} + \ldots + a_0$ and $q(z) = b_m z^m + b_{m-1}z^{m-1} + \ldots + b_0$.

$$
\begin{aligned}
(3) \qquad N(z) &= z - \frac{p(z)q(z)}{q(z)p'(z) - p(z)q'(z)} \\
&= z - \frac{a_n z^n b_m z^m + \ldots}{b_m z^m a_n n z^{n-1} + \ldots - a_n z^n b_m m z^{m-1} + \ldots} \\
&= z - \frac{a_n b_m z^{n+m} + \ldots}{a_n b_m z^{n+m-1}(n-m) + \ldots} \\
&= \frac{z[(n-m)z^{n+m-1} + \ldots] - (z^{n+m} + \ldots}{(n-m)z^{n+m-1} + \ldots} \\
&= \frac{(n-m)z^{n+m} + \ldots - z^{n+m} - \ldots}{(n-m)z^{n+m-1} + \ldots} \\
&= \frac{(n-m-1)z^{n+m} + \ldots}{(n-m)z^{n+m-1} + \ldots}
\end{aligned}
$$

From Equations 1, 2 and 3 we see that $m(N, \infty) = \frac{n-m}{n-m-1}$, which is greater than one for $n > m$ and less than one for $n < m$. This means that if ∞ is a fixed point of N, then it is repelling for $n > m$ and attracting for $n < m$. Again, this is assuming the leading terms in N do not cancel.

Simple calculation shows that $N'(z) = \dfrac{r(z)r''(z)}{[r'(z)]^2}$. It follows that the simple roots of r are superattracting, since $N'(z) = 0$ at those simple roots of r. So for any rational function Newton's Method will converge rapidly to its simple roots. As a matter of fact, in all examples of rational functions considered in this paper, the maximum number of iterations needed for convergence is less than 50. For the examples in this paper, the roots of each function are shown in the legends of the figures, rounded to two decimal places.

2 Newton's Method Applied to the Family of Functions $\mathbf{f(z) = \dfrac{az^n - b}{cz^n - d}}$

The first functions to be investigated are the family of functions $f(z) = \frac{az^n - b}{cz^n - d}$, where a, b, c, and d are complex constants. When Newton's Method is applied to this special family of functions, however, its leading terms always cancel. In these cases,

$$N(z) = z - \frac{(az^n - b)(cz^n - d)}{(bc - ad)nz^{n-1}},$$

so that the degree of the numerator is $2n$ while the degree of the denominator is $n - 1$. Thus the degree of the numerator always is greater than one more than the degree of the denominator. Since $2n > (n-1) + 1$ for all $n \geq 1$, we have the case where $N'(\infty) = \infty$ and, hence, $m(N, \infty) = \frac{1}{\infty} = 0$. Since $m(N, \infty) = 0$, ∞ is a superattracting fixed point of N. Infinity is a fixed point of Newton's Method on many of the functions in this paper; however, since it is not a root of the functions, we do not plot points whose orbits converge to it and those points remain white in the figures.

Since ∞ is a superattracting fixed point for these functions, Newton's Method will only converge to the roots of f when starting relatively close to those roots.

2.1 The $n = 1$ Case

We will start with the simplest case of n=1 and see if any generalizations can be made about the pictures generated by the code for this case. Given the function $f(z) = \frac{z-1}{z+i}$, the code produces Figure 4.1. The root $z_o = 1$ is at the center of the disk while the pole, $z = -i$ is at the very edge of the disk.

For $f(z) = \frac{3z+6-9i}{2z+4+i}$, the code produces Figure 4.2. The root $z_0 = -2 + 3i$ is at the center of the disk, and the pole, $z = -2 + \frac{1}{2}i$ is at the edge.

Given $f(z) = \frac{2z-10-6i}{z-9-3i}$, the code produces Figure 4.3. Again, the root $z_0 = 5+3i$ is at the center of the disk with the pole, $z = 9+3i$, at the edge.

Functions of this type with varying constants always seem to result in the same image. The image is that of a disk centered at the single root of $f(z)$ which is $z = \frac{b}{a}$. The radius of the disk is equal to the distance between the function's root and its pole, $z = \frac{d}{c}$. Therefore, we arrive at the following conjecture:

Conjecture: For any function of the form $f(z) = \dfrac{az-b}{cz-d}$, Newton's Method converges to the root $z = \frac{b}{a}$ for all z_0 in the disk centered

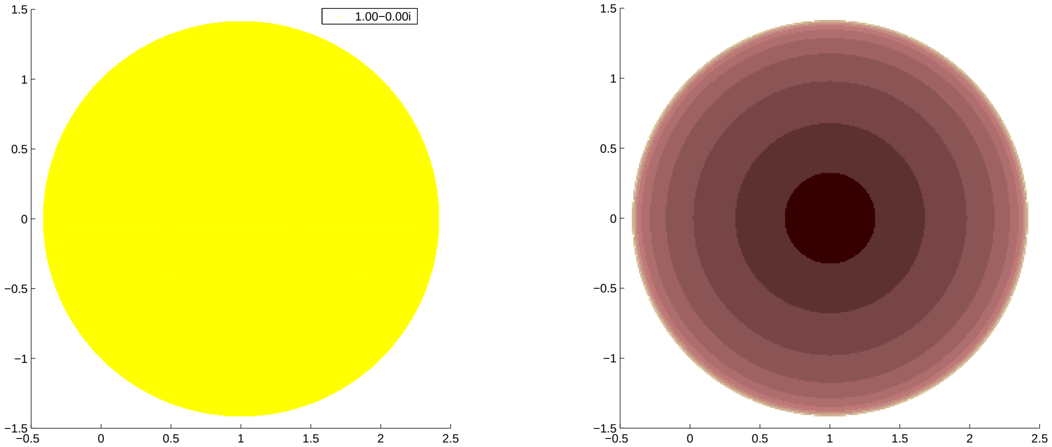

Figure 4.1: $f(z) = \frac{z-1}{z+i}$

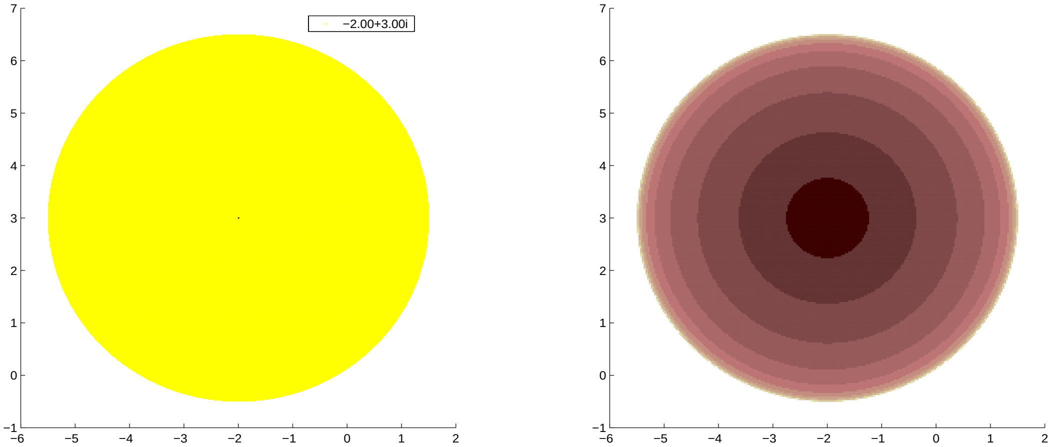

Figure 4.2: $f(z) = \frac{3z+6-9i}{2z+4+i}$

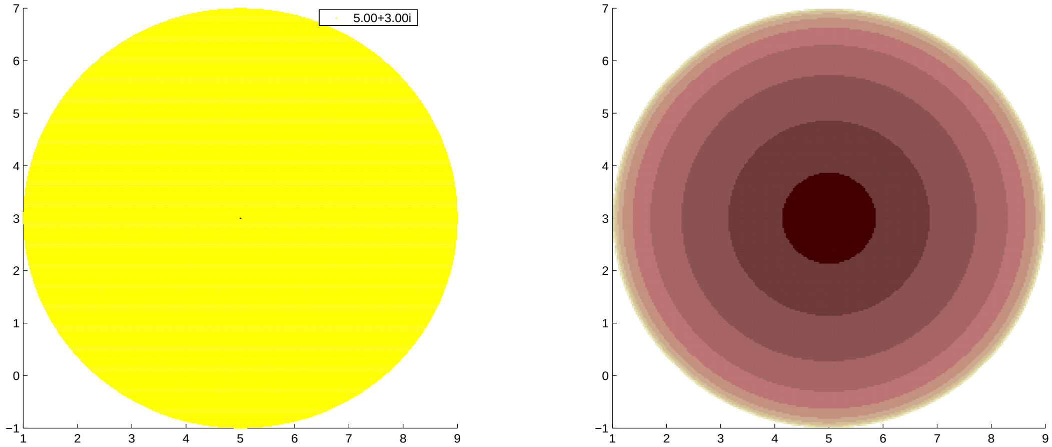

Figure 4.3: $f(z) = \frac{2z-10-6i}{z-9-3i}$

at the root with radius $r = |\frac{b}{a} - \frac{d}{c}|$. For all z_0 outside this disk, Newton's Method does not converge.

2.2 The $n = 2$ Case

Now consider the case of n=2 where $f(z) = \frac{az^2+b}{cz^2+d}$. One might expect the pictures to look like those for $n = 1$, except with one disk centered around each of the two roots. This outcome is not quite what results from running the code, however. The first function we look at is $f(z) = \frac{z^2-4}{z^2-1}$. The zeros are at ± 2, while the poles are at ± 1. We see in Figure 4.4 that there are two round-shaped regions with the roots contained in the regions, but the shapes are not perfectly circular, and the roots are not in the exact center of the regions. Also, these two regions are not the only colored regions

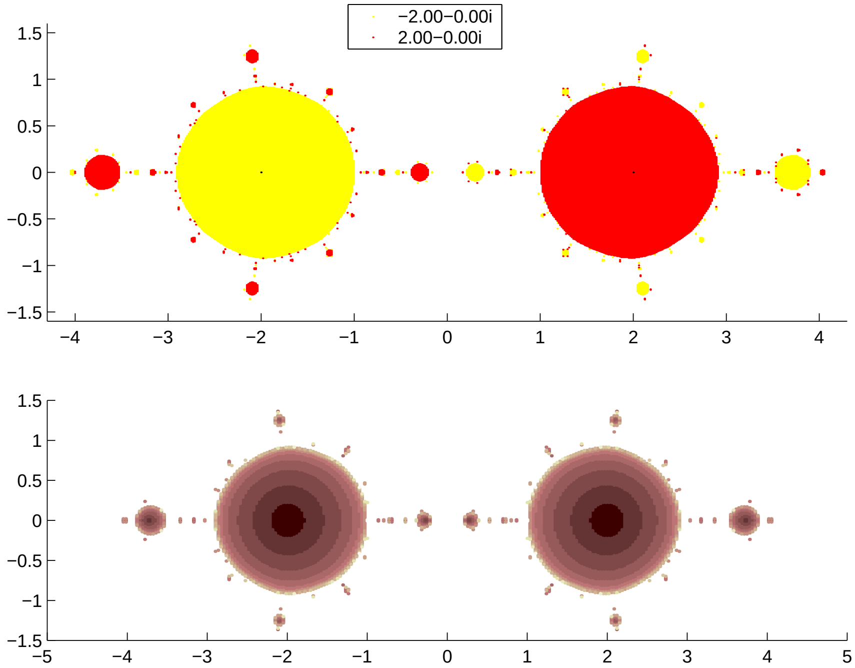

Figure 4.4: $f(z) = \frac{z^2-4}{z^2-1}$

present. Some complicated regions are clustered around the main regions. Note that the poles of the function are on the inner edges of these main regions, and that both the zeros and the poles lie along the real axis.

The picture for the function $f(z) = \frac{z^2+4}{z^2+9}$ with roots and poles on the imaginary axis is shown in Figure 4.5. Unlike Figure 4.4, however, the poles are further away from the origin than the roots. If f had been inverted, i.e. $\frac{z^2+9}{z^2+4}$, we would have a picture that looked much like Figure 4.4, but rotated 90 degrees.

Figure 4.6 is the picture for the function $f(z) = \frac{z^2-4}{z^2+9}$. The roots

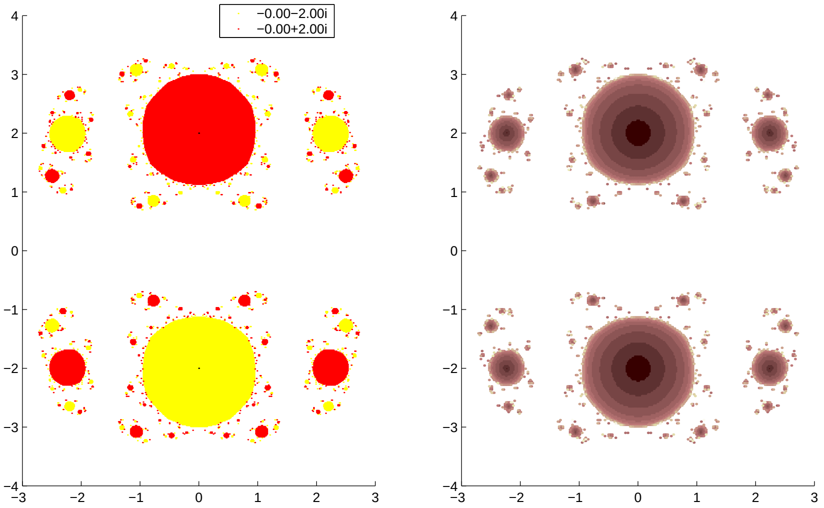

Figure 4.5: $f(z) = \frac{z^2+4}{z^2+9}$

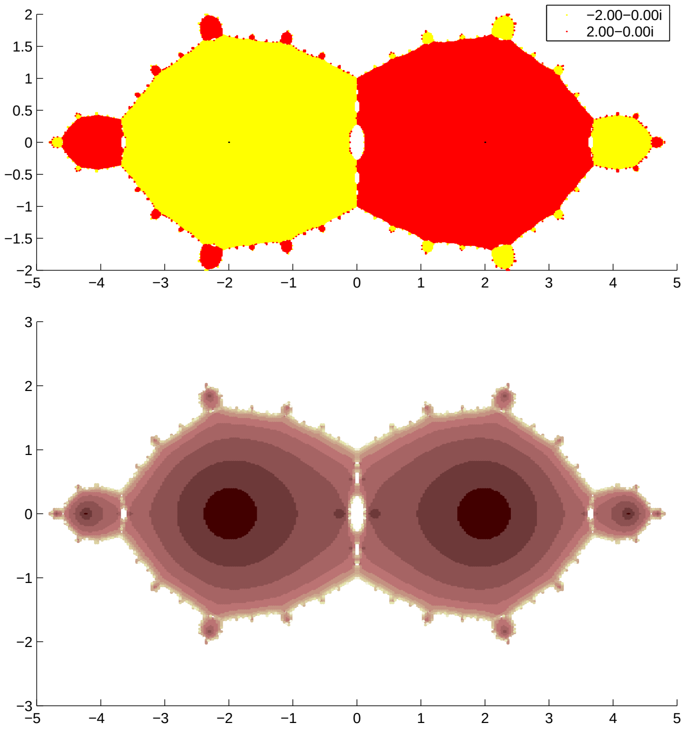

Figure 4.6: $f(z) = \frac{z^2-4}{z^2+1}$

of this function are also ± 2, but the poles, $\pm i$, lie on the imaginary axis. In this case, the two major regions meet at the poles.

A function whose roots and poles have both real and imaginary nonzero parts is $f(z) = \frac{3z^2+5i}{z^2-30i}$, for which the picture is shown in Figure 4.7. The roots of f are $0.91 - 0.91i$ and $-0.91 + 0.91i$ with poles $3.87 + 3.87i$ and $-3.87 - 3.87i$, rounded to two decimal places. Like our previous example, the roots and poles lie along perpendicular lines. In this case, however, those lines are the $y = x$ line and the $y = -x$ line rather than the real and imaginary axes. We see that the shape of Figure 4.7 is remarkably like that of Figure 4.6, only rotated 45 degrees.

So far we have tried functions which have the poles and roots on the same line and functions which have them on perpendicular lines. The function $f(z) = \frac{z^2+8i}{z^2-9.6+5.12i}$, for which the picture is shown in Figure 4.8, has roots $2 - 2i$ and $-2 + 2i$ which lie on the line $y = -x$ with poles $3.2 - 0.8i$ and $-3.2 + 0.8i$ which lie on the line $y = -\frac{1}{4}y$. This figure is not symmetrical in shape about a line the way the other examples have been. It has a rotational symmetry about the origin.

Figure 4.9 is for another function, $f(z) = \frac{z^2+8i}{z^2-12+16i}$, whose roots and poles lie on different lines that are not perpendicular.

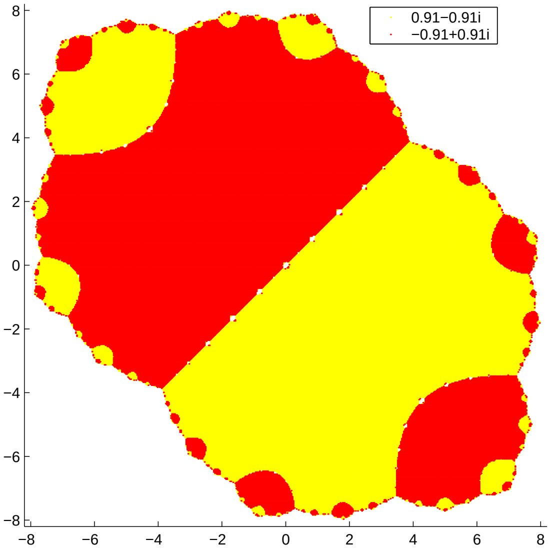

Figure 4.7: $f(z) = \frac{3z^2+5i}{z^2-30i}$

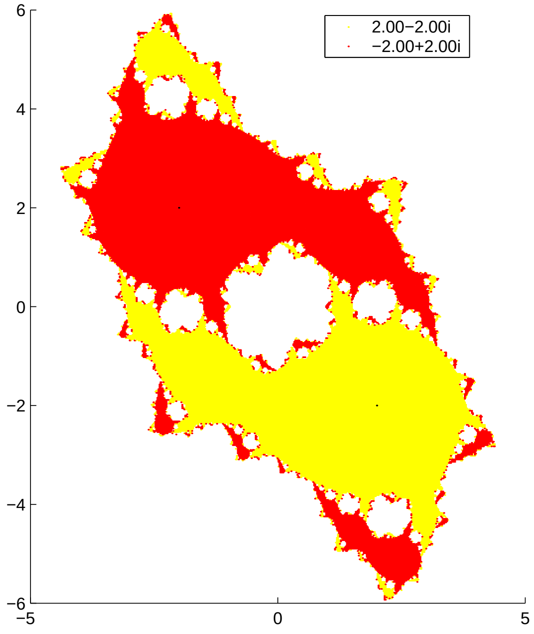

Figure 4.8: $f(z) = \frac{z^2+8i}{z^2-9.6+5.12i}$

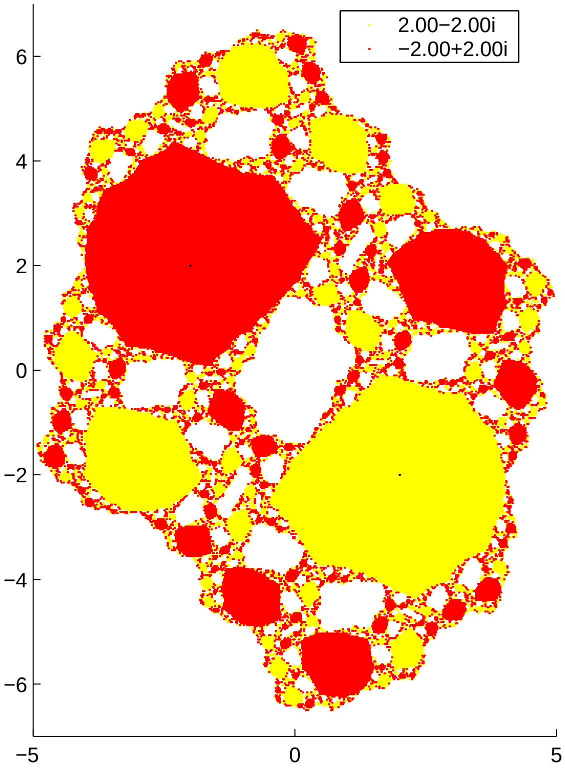

Figure 4.9: $f(z) = \frac{z^2+8i}{z^2-12+16i}$

2.3 The $n = 3$ Case

Much of what is observed in the $n = 2$ case also applies to the $n = 3$ case, and indeed, to $n > 3$. We will first look at functions with roots that are closer to the origin than the poles.

Figure 4.10 is for the function $f(z) = \frac{z^3-8}{z^3-125}$. This function has a root at 2 and a pole at 5, which both lie along the same ray emanating from the origin. The function's other two roots and poles do likewise. We see that there are very rounded major regions around each of the three roots. In between and around each of these major regions is an area with much complexity. Again, the poles are located on the very outer edges of the major regions.

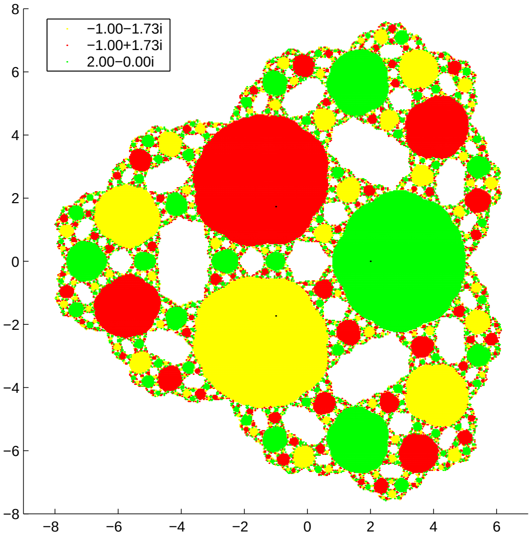

Figure 4.10: $f(z) = \frac{z^3-8}{z^3-125}$

The function $f(z) = \frac{z^3-8}{z^3+125}$, Figure 4.11, also has roots that are closer than the poles to the origin. Unlike our previous example, however, the poles lie along rays that bisect the angles formed by the rays on which the roots lie. This is analagous to the functions in Figures 4.6 and 4.7. Just like in those $n = 2$ cases, the major regions are separated by complex regions which straightly follow along the rays going to the poles. A small perturbation of a seed in the complex regions of these figures can result in a drastically different orbit that is likely to converge to a different root.

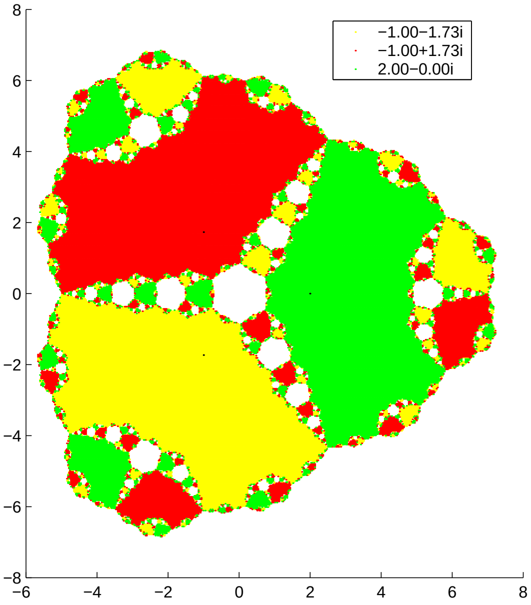

Figure 4.11: $f(z) = \frac{z^3-8}{z^3+125}$

For the next example we use $f(z) = \frac{z^3-1}{z^3+125i}$, Figure 4.12. This function has poles that are on different rays than its roots, however, the rays do not bisect each other's angles, so we will call them skew. As can be seen in the figure, the complex area separating the major regions follows a path starting near the origin and ending at the poles. Unlike our previous example, straight paths are not made. In this case it begins to follow a path perpendicularly bisecting the roots, but it then curves until it meets the poles.

Now we will consider functions that have roots which are further from the origin than are their poles. Figure 4.13 shows $f(z) = \frac{z^3-125}{z^3-8}$, which has zeros and poles both on the same rays. One of the roots is $z_0 = 5$, which is farther from the origin than the pole, $z = 2$, on the same ray. The resulting image is much like Figure 4.4 for the $n = 2$ case, with three main regions rather than two.

Let us consider a function with poles that lie along the bisectors of the angles formed by the rays on which the zeros lie. The function $f(z) = \frac{z^3+125}{z^3-8}$ shown in Figure 4.14 satisfies these conditions and also has roots which are farther from the origin than its poles. As one might begin to expect, we see an image much like Figure 4.6, but with three regions instead of two.

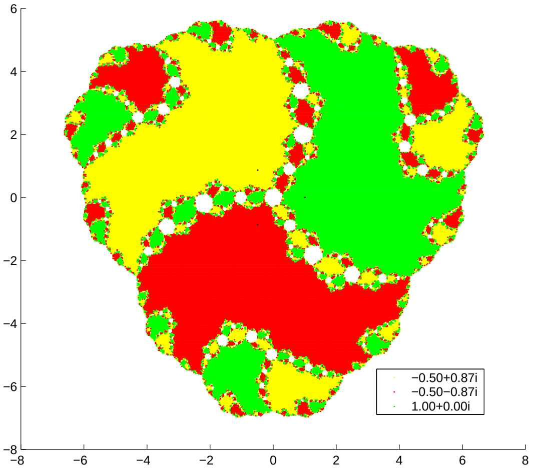

Figure 4.12: $f(z) = \frac{z^3-1}{z^3+125i}$

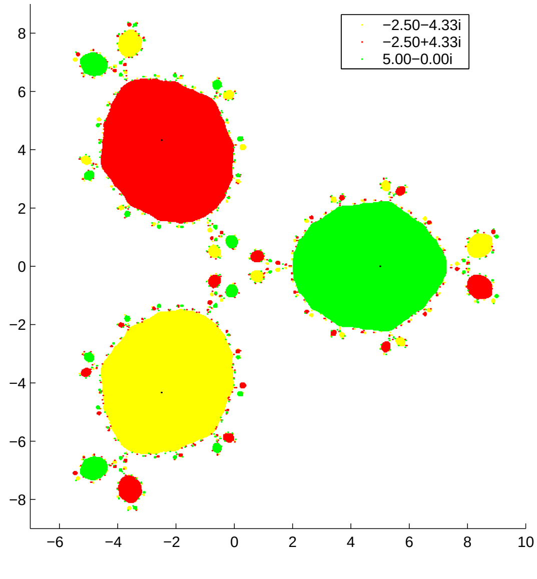

Figure 4.13: $f(z) = \frac{z^3-125}{z^3-8}$

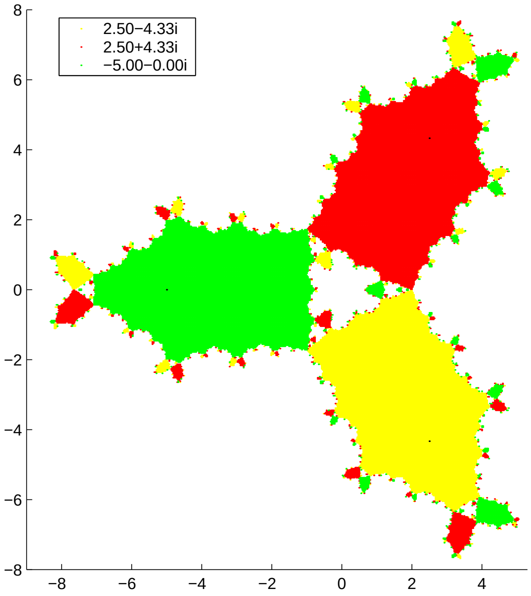

Figure 4.14: $f(z) = \frac{z^3+125}{z^3-8}$

For the family of functions $f(z) = \frac{az^n - b}{cz^n - d}$ and $n > 3$, the images follow the same pattern established here. One can think of n as the number of 'petals' in each image. Notice that seeds which are not very close to an area described by the roots and poles do not have orbits that converge to the roots. This behavior occurs because, as we mentioned at the beginning of the section, ∞ is a superattracting fixed point of Newton's Method for this family of functions.

Chapter 5

Compositions of Functions

Figure 5.1 illustrates the composite function $f(z) = log(z^2 - 1)$, which has two roots at $z = \pm\sqrt{2}$. If $p(z)$ is a polynomial of degree n, then functions of the form $f(z) = log(p(z))$ have n roots which are the solutions to the polynomial equation $p(z) = 1$.

Before looking at the composite function $f(z) = sin(log(z))$, consider the simpler function $f(z) = sin(z)$. When this function is used in the code, it produces a very scattered picture with no distinct regions. This occurs because seeds converge to a very large number of the infinite roots of $sin(z)$, however this convergence apears to be quite chaotic. The roots of $sin(z)$ are the integer multiples of π which are each the same distance from its neighbors. It seems that each root is competing with its neighbors for the orbits of seeds to converge to it.

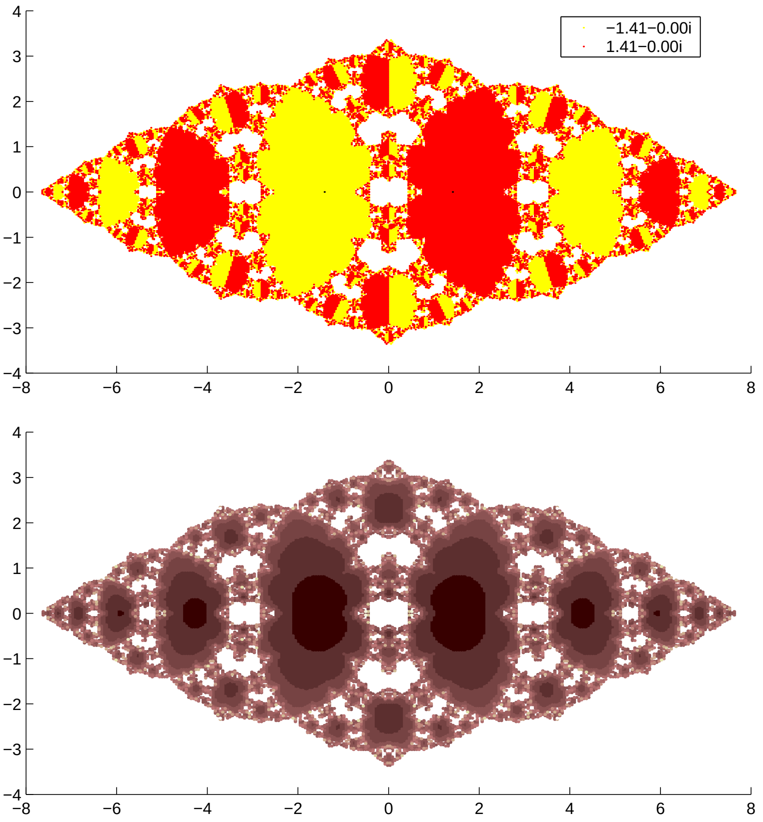

Figure 5.1: $f(z) = log(z^2 - 1)$

Figure 5.2 illustrates that $f(z) = sin(log(z))$ gives much more orderly and distinct regions of convergence than does $f(z) = sin(z)$.

Notice that at $(0,0)$ Figure 5.2 appears to have what may be a tiny replica of itself. In order to determine this, the code is run again using a window which zooms in on the area. Indeed the very same shape appears in the figure with yet again a small apparent replica of itself at $(0,0)$. Zooming out from the original window also produces the same image.

Using a finer and finer grid of points (larger meshsize) on this function results in finding more and more roots, which are not all plotted due to limitations of the code (see Appendix). The reason for this is that the roots are the solutions to the equation $sin(log(z)) = 0$, which are $z = (e^{\pi})^n$. There are infinitly many roots, and as n becomes more negative the roots approach 0. Since there is an infinite number of roots near zero, a finer mesh means more of these small roots will be 'caught' in the mesh. The reason this function has regions of convergence that are more orderly than $f(z) = sin(z)$ may be that starting from zero, each successive positive root is larger than its predecessor by the factor of e^{π}. So as we move away from the roots near zero, the large roots are spread further and further away from each other effectively 'reducing competition' among the roots.

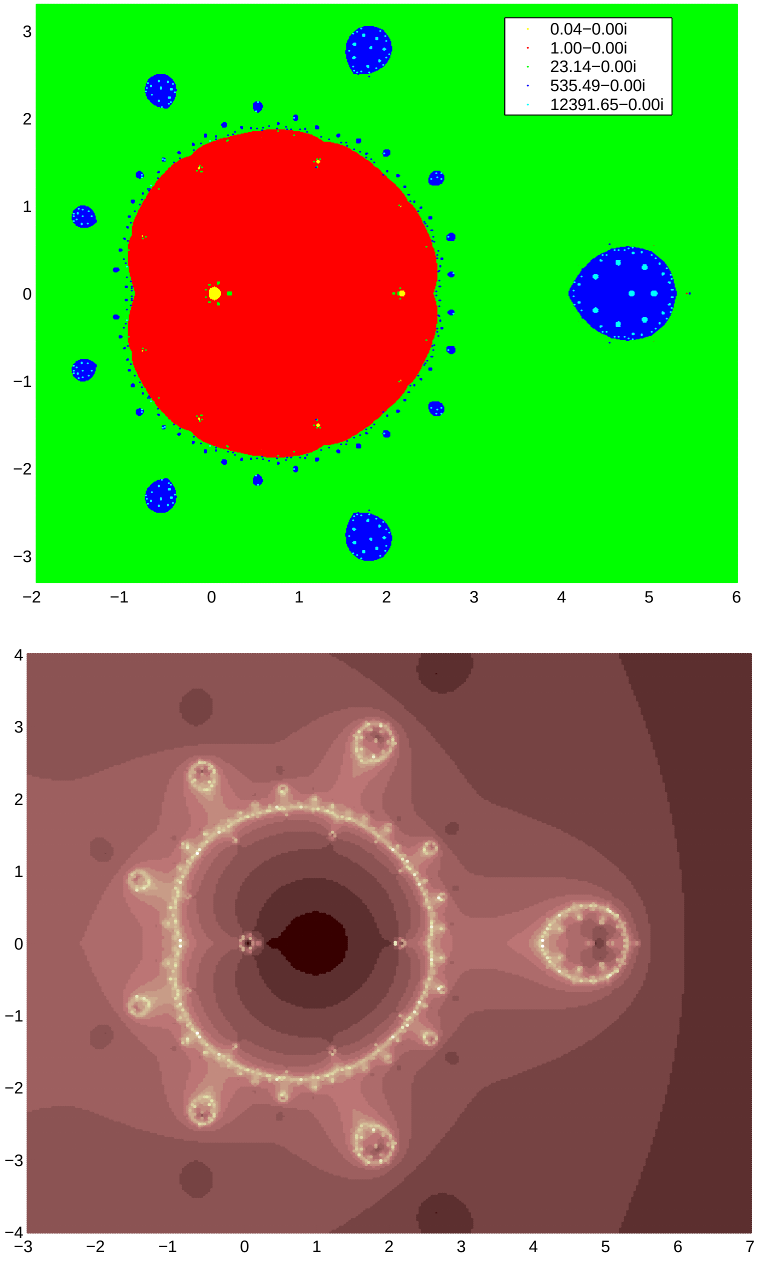

Figure 5.2: $f(z) = sin(log(z))$

Another interesting area appears in the original figure around the value $(2,0)$. To better see this region, we again zoom in producing Figure 5.3 which is a smaller mirror image of the original.

Another function with orderly regions of convergence is $f(z) = sin(z^{\frac{1}{3}})$ seen in Figure 5.4. The roots of this function are $z = (n\pi)^3$. Like the roots of $f(z) = sin(log(z))$, the roots of $f(z) = sin(z^{\frac{1}{3}})$ get farther apart as we move away from the origin. Surprisingly, as can be seen from Figure 5.4, none of the negative roots are converged to.

The scale of the image is very large, and as one zooms out even further, it appears that these nested regions continue indefinitely although we run out of colors to use in the plot. Unlike the previous example, there is not an infinite number of roots at zero, so this nested shape eventually ends upon zooming in (see Figure 5.5).

What we see upon zooming in on the single root at the origin is shown in Figure 5.6. This new pattern continues under repeated zooming in until reaching the limit of Matlab's accuracy of 16 significant decimal digits.

Several other composite trigonometric functions that have infinitly many roots demonstrate ineresting dynamics, such as $sin(log(sin(z)))$, $sin(\pi e^z)$ and $sin(log(ze^z))$. Unfortunately it is difficult to plot the seeds according to which root is converged to. How does one color

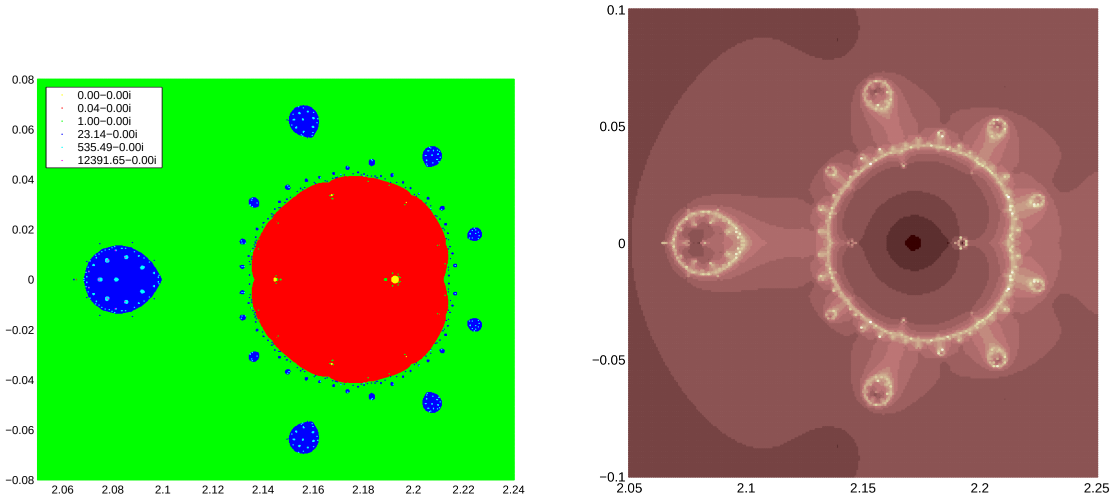

Figure 5.3: Zoomed in picture for $f(z) = sin(log(z))$

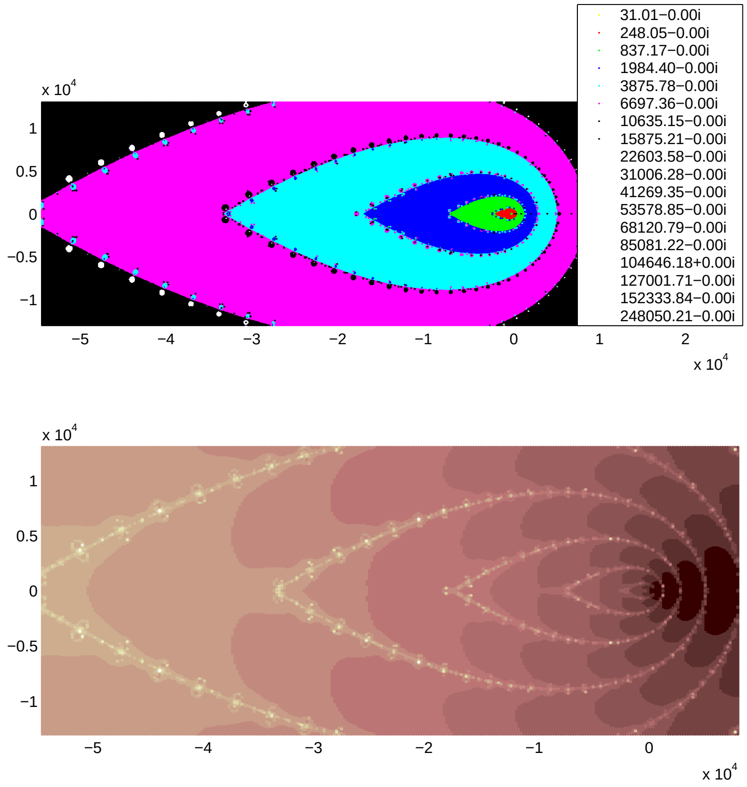

Figure 5.4: Zoomed out picture for $f(z) = sin(z^{\frac{1}{3}})$

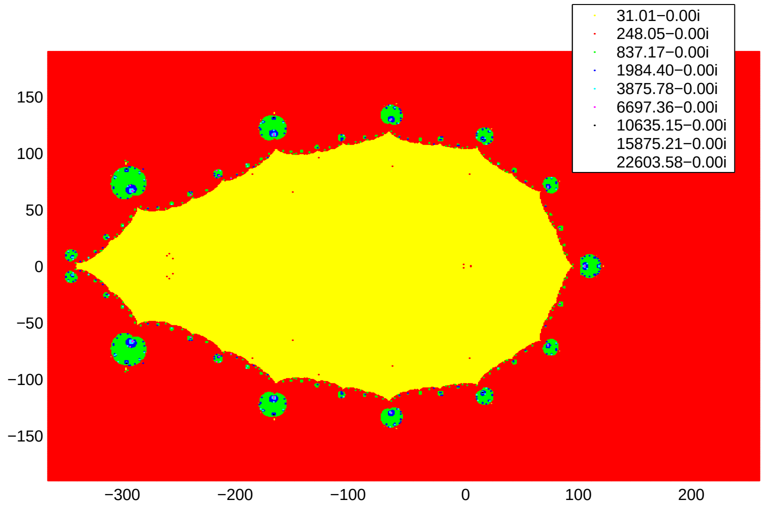

Figure 5.5: $f(z) = sin(z^{\frac{1}{3}})$

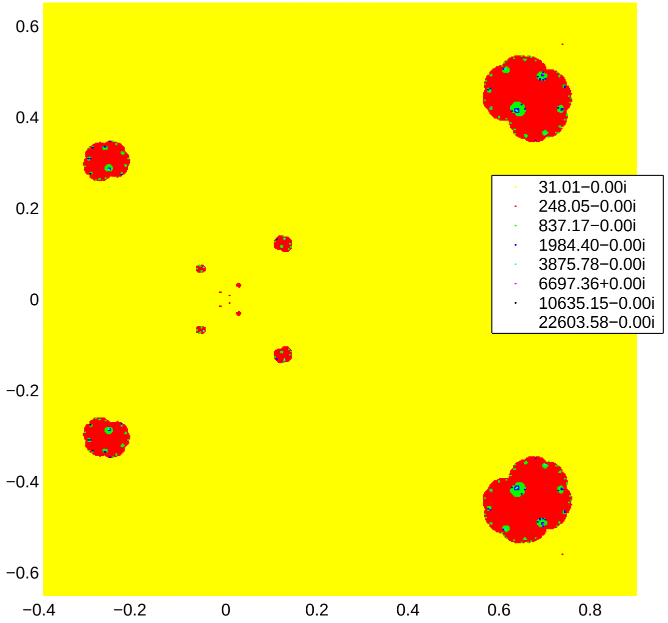

Figure 5.6: Zoomed in picture for $f(z) = sin(z^{\frac{1}{3}})$

regions using infinitely many different colors? Trying to plot these functions using the first coloring scheme is not very fruitful. Only a few convergent seeds are actually plotted resulting in a scattered and mostly white image. It is possible to create nice figures by plotting according to the rate of convergence. Unfortunately this approach gives limited feedback as one can not tell what the orbits are converging to. Figures 5.7 and 5.8 are rate of convergence plots for $sin(\pi e^z)$ and $sin(log(sin(z)))$, respectively.

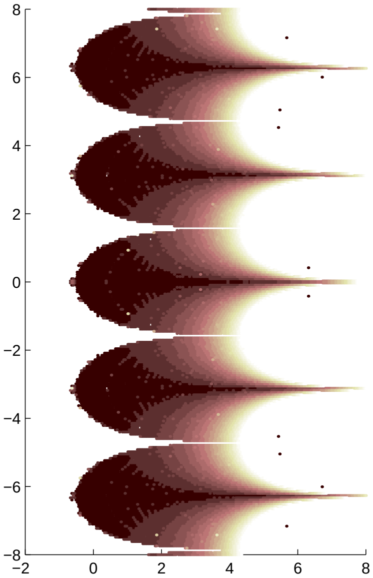

Figure 5.7: $f(z) = sin(\pi e^z)$

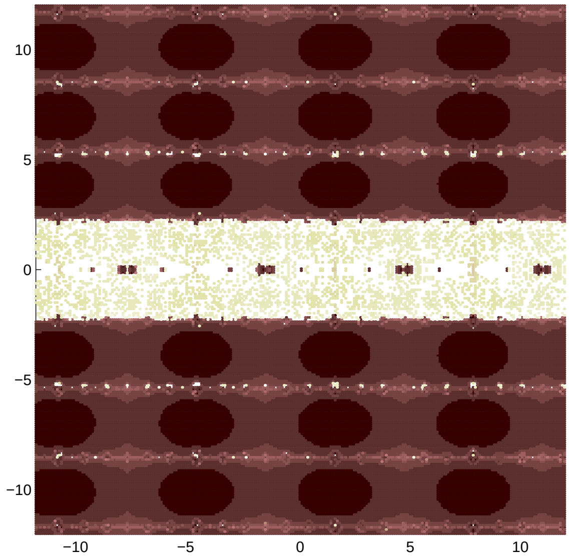

Figure 5.8: $f(z) = sin(log(sin(z)))$

Chapter 6

Closing Remarks

The behavior of functions of the type $f(z) = \frac{az^n-b}{cz^n-d}$ and of a few composite functions is investigated in this paper. The major regions of convergence of Newton's Method to the roots of the functions have been shown to be somewhat predictable given the type of function, the nature of its roots, and in some cases its poles.

The images produced by our Matlab code for the function $f(z)$ are filled Julia sets for the function $N(z) = z - \dfrac{f(z)}{f'(z)}$. The maximum number of iterations is never reached for the functions investigated here. Therefore, for all of our examples the orbits of the seeds either converge to a root or diverge to infinity, but they do not remain bounded without converging to a root. Hence, the code is plotting all of the seeds in the window that have bounded orbits, which is by definition a filled Julia set.

This paper has dealt with applying Newton's Method to a specific type of rational function where the degrees of the numerator and denominator are the same. Investigation of other types of rational functions would be quite interesting and may possibly lead to furthur generalizations. Recall that keeping track of the leading terms of N gives us

$$N(z) = \frac{(n-m-1)z^{n+m} + \ldots}{(n-m)z^{n+m-1} + \ldots}.$$

Of special interest, therefore, is the case where $n = m+1$. In such cases, the coefficient of the leading term in the numerator of N is zero making the degree of the numerator and denominator of N the same and eliminating ∞ as a fixed point of N. Other cases which would be interesting to explore are the cases where $n < m$ and $n > m+1$.

Bibliography

Bibliography

[1] Alan F. Beardon. *Iteration of Rational Functions* . Springer-Verlag, 1991.

[2] Paul Blanchard. The Dynamics of Newton's Method. *Proceedings of Symposia in Applied Mathematics*, 49:139, 1994.

[3] Bodil Branner. The Mandelbrot Set. *Proceedings of Symposia in Applied Mathematics*, 39:75, 1989.

[4] Anne M. Burns. Plotting the Escape - An Animation of Parabolic Bifurcations in the Mandelbrot Set. *Mathematics Magazine*, 75:104, 2002.

[5] Ruel V. Churchill, James W. Brown, and Roger F. Verhey. *Complex Variables and Applications.* Mcgraw-Hill Book Company, 1948.

[6] Robert L. Devaney. The Dynamics of Simple Maps. *Proceedings of Symposia in Applied Mathematics*, 39:1, 1989.

[7] Desmond J. Higham and Nicholas J. Higham. *Matlab Guide* . SIAM, 2000.

[8] Linda Keen. Julia Sets. *Proceedings of Symposia in Applied Mathematics*, 39:57, 1989.

[9] John H. Mathews and Russell W. Howell. *Complex Analysis for Mathematics and Engineering.* Jones and Bartlett Publishers, 2001.

Appendix

1 Code

```
% Call in one of the following ways:
   % compNewt;      to use default values
   % compNewt(Routine);
   % compNewt(Routine,M);
   % compNewt(Routine,M,a,b,c,d);
   % unspecified values will take on default values
   function [dr] = compNewt(Route,M,a,b,c,d)
   if (nargin =0 & nargin =1 & nargin =2 & nargin =6)
      'Error in number of input arguments'
      return
   end
   if (nargin < 3)
      a = -1.5; b = 1.5;
      c = -1.5; d = 1.5;
   end
   if (nargin < 2)
      M=100;
   end
   if (nargin < 1)
      Route = 0;
```

```
end
if (Route∼=0 & Route∼=1 & Route∼=2)
   ’The first input must be 0, 1, or 2’
   return
end
if M < 2
   ’The second input must be greater than 1’
   return
end
R = .0000001;
Nit = 500;
TOL = 1.e-8;
ct = 0;
ba = b - a;
dc = d - c;
xi = a + ba*[0:(M-1)]/(M-1);
yj = c + dc*[0:(M-1)]/(M-1);
[X,Y] = meshgrid(xi,yj);
Z0 = X+i*Y;
Z = Z0;
NZ0=reshape(Z0,1,(M2̂));
NZ = NZ0;
```

```
Zout = [];
Zloc = [];
Znit = [];
for n = 1:Nit                 % Iteration of Newton’s Method
   Iterate = [];
   [f,df] = newtf(NZ);
   ind = find(abs(df) < TOL);
   NZ(ind)=[]; NZ0(ind)=[];
   f(ind)=[]; df(ind)=[];
   Zold = NZ;
   NZ = NZ - f./df;
   ind = find(abs(NZ-Zold) < R);
   Zout = [Zout, NZ(ind)];
   Zloc = [Zloc, NZ0(ind)];
   Iterate([1:(M+1)2̂]) = n;
   Znit = [Znit, Iterate(ind)];
   NZ(ind) = []; NZ0(ind)=[];
end
dr=[];   B=[];   C=[];
B=sort(Zout);
dr=B(1);
sB = size(B);
```

```
for i = 2:sB(2)             % Searches through Zout to find
    ct=0;                  % the distinct roots, placing
    sdr=size(dr);            % them in dr.
    for j = 1:sdr(2)
        if (abs(B(i)-dr(j)) < .001)
            ct=ct+1;
        end
    end
    if ct==0
        dr = [dr,B(i)];
    end
end
switch Route
case 0                              % colors figure according to the root
converged to.
    picture = firstplot(dr,sdr,Zloc,Zout)
case 1                              % Colors figure according to rate of
convergence
    picture = secondplot(Znit,Zloc)
case 2                              % Does both figure plots.
    firstpicture = firstplot(dr,sdr,Zloc,Zout)
    secondpicture = secondplot(Znit,Zloc)
```

```
end
% Subfunction function picture = firstplot(dr,sdr,Zloc,Zout)
picture = ’plot color-coded by root’;
figure(1);
hold on
if sdr(2) > 0
   z1 = Zloc(find(abs(Zout-dr(1)) < .001));
   plot(z1,’r.’);
end
if sdr(2) > 1
   z2 = Zloc(find(abs(Zout-dr(2)) < .001));
   plot(z2,’y.’);
end
if sdr(2) > 2
   z3 = Zloc(find(abs(Zout-dr(3)) < .001));
   plot(z3,’b.’);
end
if sdr(2) > 3
   z4 = Zloc(find(abs(Zout-dr(4))<.001));
   plot(z4,’g.’);
end
if sdr(2) > 4
```

```
   z5 = Zloc(find(abs(Zout-dr(5)) < .001));
   plot(z5,’c.’);
end
if sdr(2) > 5
   z6 = Zloc(find(abs(Zout-dr(6)) < .001));
   plot(z6,’m.’);
end
if sdr(2) > 6
   z7 = Zloc(find(abs(Zout-dr(7)) < .001));
   plot(z7,’k.’);
end
hold off
h=findobj;
get(h,’type’);
set(h(3),’DataAspectRatio’,[1 1 1]);        % Forces scale of
                                          % the figure’s axes
                                          % to be the same.
drleg=dr.’;
leg = num2str(drleg,’%0.2f’);
legend(leg,0);                             % Creates legend of roots
                                          % rounded to 2 dec pts.
hold off
```

```
sh=size(h);
for m = 4:sh(1)
    set(h(m),’Marker’,’.’,’MarkerSize’,3)
end
% Subfunction
function picture = secondplot(Znit,Zloc)
picture = ’plot color-coded by rate of convergence’;
figure(2);
hold on
inc=[0];
put=[];
minit=Znit(1);
maxit=Znit(end);
ni=min((maxit-minit),20)
for i=1:ni
    put=minit+round((maxit-minit)/ni*i);
    inc = [inc, put];
    put=[];
end
mymap=pink(ni);          % Creates colormap of appropriate size.
for i=1:ni
    p = Zloc(find(Znit > inc(i) & Znit < (inc(i+1)+1)));
    scatter(real(p),imag(p),35,mymap(i,:),’.’)
end
h=findobj;
get(h,’type’);
set(h(3),’DataAspectRatio’,[1 1 1]);
function [f,df] = newtf(z)
%———— given z, finds f(z) & f’(z)
f = z.3̂-1;
df = 3*z.2̂;
```

2 Input Arguments

The user may input zero, one, two or six arguments when calling the function compNewt. An error message will appear and there is a break in the code if the user inputs the incorrect number of arguments. The first argument, *Route*, must take on a value of 0, 1, or 2, which determines what type of plot is performed. The second argument, *M*, determines the number of points that will be in the rectangular window of interest. This window is determined by the next 4 arguments, a, b, c, and d. The values a and b determine the real-axis boundaries in the complex plane while c and d determine the imaginary-axis boundaries. If the user enters an incorrect value for either *Route* or *M*, an error message will appear and there is a break in the code.

3 Coding Newton's Method

Some parameters which are built in to the code, but which can be changed are *R*, *Nit*, and *TOL*. For the examples in this paper, R=1×10^-8, *Nit*=500, and TOL=1×10^-8. *R* is used to determine whether or not Newton's Method has converged. The current iterate of Newton's Method is determined to converge if it is within a distance of *R* from the previous iterate of Newton's Method. The maximum number of iterations of Newton's Method that will be perfomed is *Nit*. This is to prevent the **for** loop which performs Newton's Method from running indefinitely. The value of *TOL* is a tolerance which is used to 'throw out' seed values which result in division by zero in Newton's Method. If the derivative of the function, *df*, is less than than *TOL*, i.e., close to zero, then Newton's method will diverge. These points are removed from the output matrix, so they will not be plotted. Those seeds whose orbits converge are stored in the output matrix, *Zout*, and the number of iterations it took the orbit to converge is stored in the matrix *Znit*.

The first matrix of seeds produced by the code is an M by M square matrix. In order to perform the arithmetic operations of Newton's Method in Matlab, the matrix needs to be in vector form. In the first version of the code, this was done with two nested **for** loops which placed the elements of the square matrix one by one into a 1 by M^2 vector matrix. It was later discovered that instead of element by element, this could be done column by column, which very significantly increased the efficiency of the code. The discovery of the built-in Matlab function, **reshape**, completely eliminated the need of **for** loops to do the job and significantly increased efficiency once again.

4 Plotting the Figures

The output matrix, *Zout*, is sorted and searched to find the distinct roots to which orbits have converged. These are stored in the matrix *dr*.

The first plot subfunction uses the size of *dr* in **if** statements to determine which plots to do. If the size of *dr* is > 0, then the seeds which converged to the first element in *dr* are plotted. If the size of *dr* is > 1, then the seeds which converged to the second element in *dr* are plotted. This continues until we run out of colors. A problem with this approach is that if there are more than seven roots, only the first seven are plotted.

The second plot subfunction creates a **colormap** whose size is determined to be the minimum of either (*maxit-minit*) or the number 20. The number (*maxit-minit*) is the largest number of iterations needed minus the least number of iterations needed for convergence. This will be the number of shades of the **colormap** (in this case **pink**) that will be plotted. Then in each pass of a **for** loop, **scatter** is used to plot the seeds which converged within a certain number of iterations.

Scatter is not ideal for this type of plot. It creates a separate object in Matlab for each and every point that is plotted. This approach uses a very large amount of memory which causes the code to run slowly.

Vita

Jennifer Joan Corte was born on October 16, 1975 in Lubbock, Texas. She grew up in Buffalo, NY, graduating from Nichols School in May 1993. She received a Bachelor of Arts degree with a double major in Mathematics and Psychology from Tusculum College in May 1998. In August 2003 she will receive a Master of Science degree in Mathematics from the University of Tennessee, Knoxville.