



university of
 groningen

faculty of mathematics
and natural sciences

The Complex Dynamics of Newton's Method

Bachelor Project Mathematics

June 2016

Student: A.G. Wiersma

First supervisor: Dr. A.E. Sterk

Second supervisor: Prof. dr. ir. R.W.C.P. Verstappen

Abstract

Newton's method is the best known iteration method for finding a real or a complex root. When the function $f(x)$ has more than one root, which root Newton's method finds depends on the initial guess. This leads to an interesting pattern even for polynomials. We can even expand this to the complex plane by using the function $f(z)$ of a complex number z . The behavior of Newton's method for these functions leads to interesting fractal patterns. We will look at how these fractal patterns emerge for several polynomial and trigonometric functions. Furthermore we will compare the dynamics of Newton's method with the relaxed Newton's method and the Secant method.

Contents

1	Newton's Method	4
1.1	Introduction	4
1.2	How Newton's method works	4
1.3	When Newton's method fails	5
1.4	Dynamics of Newton's method	7
2	Functions with two roots	10
2.1	Function with real roots	10
2.2	Function with complex roots	13
3	Functions with more than two roots	15
3.1	Third degree polynomials	15
3.1.1	Function with only real roots	15
3.1.2	Function with complex roots	21
3.2	Fourth degree polynomials	23
3.3	Fifth degree polynomials	24
3.4	Trigonometric functions	24
4	Variations of Newton's method	27
4.1	Relaxed Newton's method	27
4.2	Secant method	32
4.2.1	Newton secant	34
4.2.2	First initial point depends on the second	34
5	Conclusion	36
6	References	37
7	Appendix: Matlab code	38

1 Newton's Method

1.1 Introduction

Newton's method is the best known iteration method for finding a real or a complex root of a differentiable function. It is also called the Newton-Raphson method named after Isaac Newton and Joseph Raphson. When the function $f(x)$ has more than one root, the root found by Newton's method depends on the initial guess. This behavior of which root is found leads to an interesting pattern. By expanding the initial guess to include numbers in the complex plane a fractal pattern will emerge.

In the rest of this section we will look at how Newton's method works, when it fails and see how we can determine which initial point leads to which root. Then in section 2 we will expand the method to the complex plane for a function with two roots. We will look at functions with more than two roots in section 3. And in section 4 we will look at two variations of Newton's method and see if they have similar or entirely different dynamics.

1.2 How Newton's method works

The idea behind Newton's method for finding the roots of a function $f(x)$ is as follows. The first step is finding a initial point in the domain of f , let's call this point x_0 . The second step is to determine the value of $f(x_0)$. The third step is determining at which point the tangent line at $f(x_0)$ crosses the x -axis. This point will be the next point in the iterative method. Let's call this point x_1 . We stop the iteration when we have $|x_{n+1} - x_n| < \text{tolerance}$, where the tolerance is predetermined.

As an example of how this works let's use the polynomial $f(x) = \frac{1}{2}x^3 - \frac{1}{4}$ this function can be seen in figure 1. As a starting point let's use $x_0 = \frac{1}{2}$ then $f(x_0) = -\frac{3}{16}$. We can determine the formula for when the tangent line crosses the x -axis using the equation of a tangent line, which is

$$y = f'(x_0)(x - x_0) + f(x_0).$$

The tangent line crosses the x -axis when it is equal to zero so

$$\begin{aligned} 0 &= f'(x_0)(x_1 - x_0) + f(x_0) \\ x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)}. \end{aligned}$$

So the next iteration point for this case will be

$$x_1 = \frac{1}{2} - \frac{\frac{1}{2}(\frac{1}{2})^3 - \frac{1}{4}}{\frac{3}{2}(\frac{1}{2})^2} = 1.$$

In figure 1 we can see this process graphically. Using the equation $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$ we can generate a sequence of points: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$. We define the Newton

function for f as follows: $N(x) = x - \frac{f(x)}{f'(x)}$. Newton's method works by finding the fixed points of the Newton function. These fixed points then correspond to the roots of the function $f(x)$.

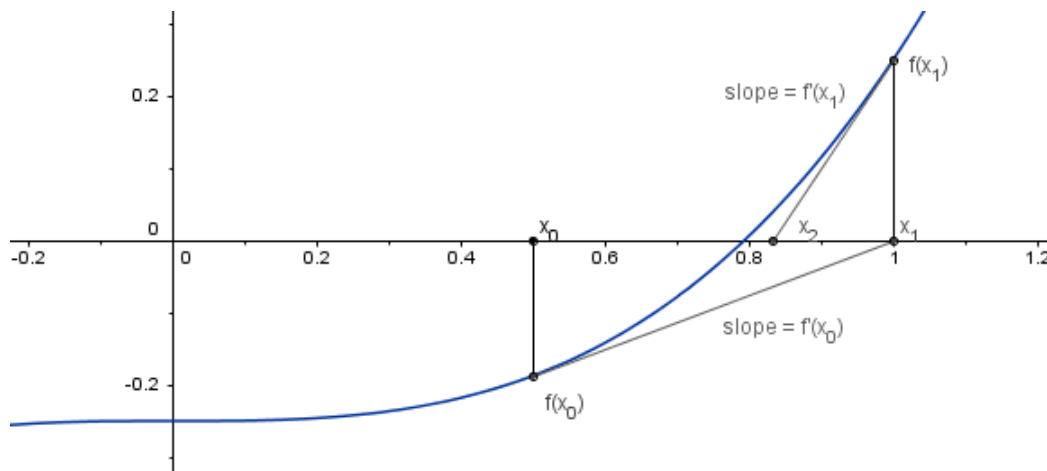


Figure 1: First two iterations of Newton's method for the function $f(x) = \frac{1}{2}x^3 - \frac{1}{4}$.

1.3 When Newton's method fails

Newton's method does not always find the roots of a polynomial. Here are the cases when Newton's method fails and why.

Critical point

If the initial point x_0 is a critical point of $f(x)$, meaning $f'(x_0) = 0$ then the tangent line is a horizontal line, which can be seen in figure 2, and therefore will never cross the x -axis. So the next iteration point doesn't exist and Newton's method will not converge to a root.

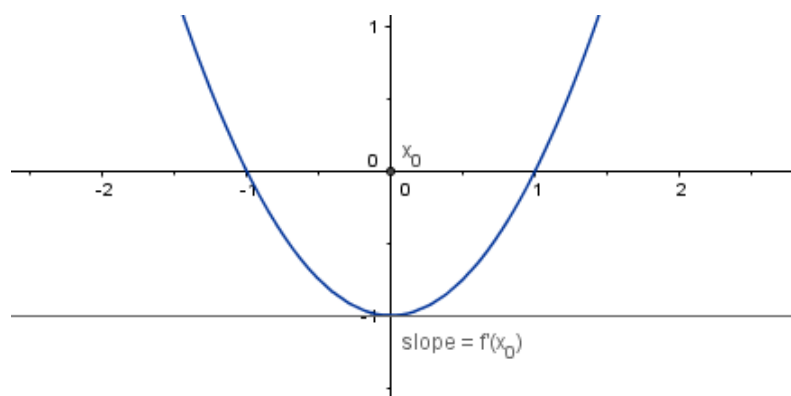


Figure 2: Newton's method applied to the function $f(x) = x^2 - 1$. The initial guess coincides with a critical point.

Periodic dynamics

It can happen that the initial point coincides with a cycle. The following example is from [2]. Consider the function $f(x) = x^3 - 5x$, which has $N(x) = x - \frac{x^3 - 5x}{3x^2 - 5}$ and choose $x_0 = 1$. Then we have

$$x_1 = N(x_0) = 1 - \frac{(1)^3 - 5(1)}{3(1)^2 - 5} = -1,$$

and

$$x_2 = N(x_1) = -1 - \frac{(-1)^3 - 5(-1)}{3(-1)^2 - 5} = 1.$$

So we have $x_2 = x_0$ this means we will get a cycle of period 2. This can be seen in figure 3.

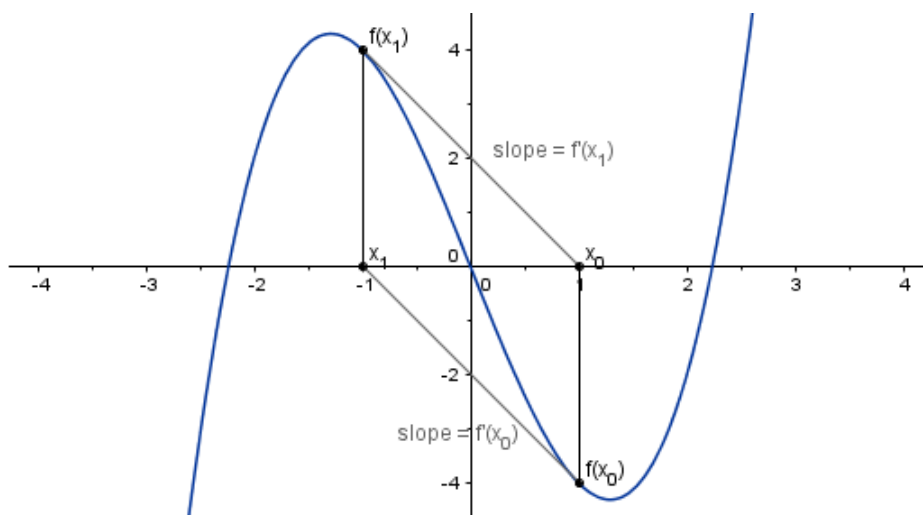


Figure 3: Newton's method for $f(x) = x^3 - 5x$ showing a 2-cycle.

The root doesn't exist

If there are no roots then Newton's method will fail. For example the function $f(x) = x^2 + 1$. As can be seen in figure 4 this function has no roots so Newton's method will not converge.

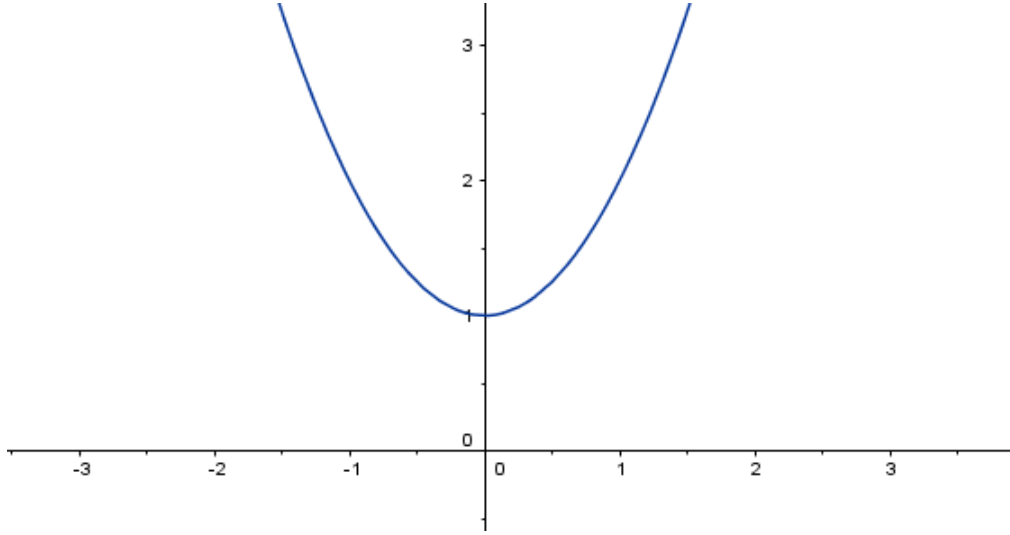


Figure 4: The function $f(x) = x^2 + 1$ has no real roots.

1.4 Dynamics of Newton's method

So now we know how Newton's method works and when it fails. However what happens when there is more than one root to find? Depending on the choice of the initial guess x_0 Newton's method will find a different root. But how do you determine which root is found by Newton's method?

Definition 1. If x_* is a root of f , the basin of attraction of x_* is the set of all numbers x_0 such that Newton's method starting at x_0 converges to x_* . In symbols

$$B(x_*) = \{x_0 | x_n = N^n(x_0) \text{ converges to } x_*\}.$$

In order to determine these basins of attractions we will first look at the Newton function $N(x) = x - \frac{f(x)}{f'(x)}$. By determining what kind of fixed points the Newton function has we can find out which initial point leads to which root. From [6] we have the following theorem:

Theorem 1. (Schroder fixed point theorem). If f is an analytic function with $f(\lambda) = \lambda$ and $|f'(\lambda)| < 1$, then there is neighborhood D of the fixed point λ on which all points converge to λ under iteration by f ; that is, for any $z \in D$, $\lim_{n \rightarrow \infty} f^n(z) = \lambda$.

Proof for real numbers [2]. Since $|f'(\lambda)| < 1$ there is a number $\alpha > 0$ such that $|f'(\lambda)| < \alpha < 1$. Therefore we can choose a number $\delta > 0$ so that $|f'(x)| < 1$ provided that $x \in I$, with $I = [\lambda - \delta, \lambda + \delta]$. Now let p be any point in I . Then by the Mean Value Theorem we have

$$\frac{|f(p) - f(\lambda)|}{|p - \lambda|} < \alpha,$$

so that

$$|f(p) - f(\lambda)| < \alpha |p - \lambda|.$$

Because λ is a fixed point so $f(\lambda) = \lambda$ we have

$$|f(p) - \lambda| < \alpha|p - \lambda|.$$

So we have $|f(p) - \lambda| < |p - \lambda|$ because $0 < \alpha < 1$. Therefore the distance between $f(p)$ and λ is smaller than the distance between p and λ . And since $f(p) \in I$ we can use the same argument again and get

$$\begin{aligned} |f^2(p) - \lambda| &= |f^2(p) - f(\lambda)| \\ &< \alpha|f(p) - f(\lambda)| \\ &< \alpha^2|p - \lambda|. \end{aligned}$$

Since $\alpha < 1$ we also have $\alpha^2 < 1$. So the points $f^2(p)$ and λ are closer together than the points $f(p)$ and λ . And we have $f^2(p) \in I$ so we can repeat the argument. By continuing this argument we find that, for $n > 0$,

$$|f^n(p) - \lambda| < \alpha^n|p - \lambda|.$$

Since $0 < \alpha < 1$, $\alpha^n \rightarrow 0$ as $n \rightarrow \infty$. Therefore $f^n(p) \rightarrow \lambda$ as $n \rightarrow \infty$. This completes the proof for real numbers. \square

The whole proof for complex numbers will not be included however the idea behind it is as follows. From [6] we have that the proof relies on the Taylor series of f expanded about the attracting fixed point λ . By choosing a point infinitely close to λ the first term of the Taylor series outweighs all the succeeding terms. So $f(z) - \lambda = f'(\lambda)(z - \lambda)$ then by inductive argument we have $f^n(z) - \lambda = (f'(\lambda))^n(z - \lambda)$. And then $f^n(z) - \lambda$ converges to zero as $n \rightarrow \infty$.

Fixed points are often classified according to the following definition:

Definition 2. Suppose that a map $f : X \rightarrow X$ is differentiable at a fixed point x_* . Then

- (i) x_* is attracting if and only if $|f'(x_*)| < 1$
- (ii) x_* is super-attracting if and only if $|f'(x_*)| = 0$
- (iii) x_* is repelling if and only if $|f'(x_*)| > 1$.

By this definition we have that all fixed points of the Newton function $N(x)$ are super attracting fixed points. Because if we let x_* be a root of $f(x)$ then $f(x_*) = 0$ and we have

$$N'(x_*) = 1 - \frac{f'(x_*)f'(x_*) - f(x_*)f''(x_*)}{f'(x_*)^2} = 1 - \frac{f'(x_*)^2}{f'(x_*)^2} = 0.$$

So we have that all fixed points of $N(x)$ are super attracting points since $|N'(x_*)| = 0$ for all fixed points x_* of $N(x)$.

Determining the basin of attraction

We can use the previous theorem and definitions to determine the basins of attraction of Newton's method. As an example consider the function $f(x) = (x - 1)(x + 1) = x^2 - 1$. This function has two roots $x = 1$ and $x = -1$. We want to find when Newton's method converges to $x = 1$ and when it converges to $x = -1$. First by looking at figure 5 we can see that $x = 0$ is a critical point and therefore Newton's method will fail at this point. Furthermore from figure 5 we can see that when $x_0 > 1$ Newton's method will converge to the root $x = 1$ since it is a super attracting fixed point. When $0 < x_0 < 1$ Newton's method will also converge to $x = 1$ because then $x_1 > 1$. So we have $(0, \infty) = B(1)$. With the same reasoning we find that $(-\infty, 0) = B(-1)$.

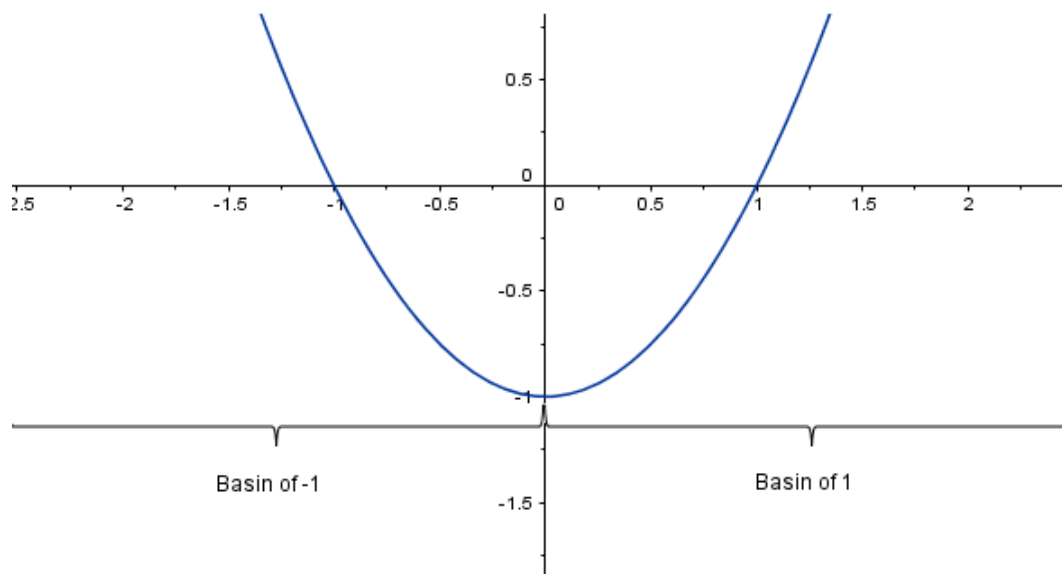


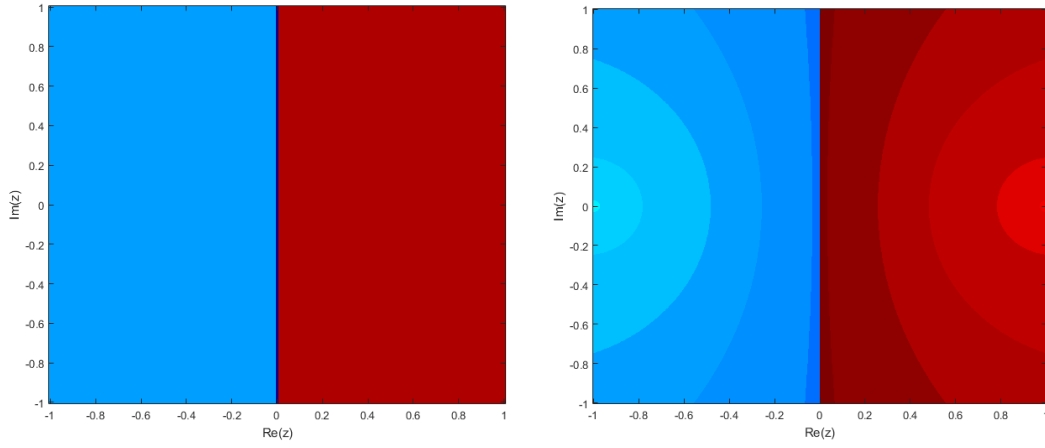
Figure 5: The basins of attraction of Newton's method for the function $f(x) = x^2 - 1$.

2 Functions with two roots

We can further analyze the behavior of Newton's method by expanding the selection of the initial point to the complex plane. We will still get a solution even when the roots of the function are real. The best way to analyze the behavior of Newton's method in the complex plane is by drawing the basins of attractions in a figure. This can be done by using Matlab, for the code see the appendix. The figure will be drawn by taking a point in the complex plane and use it as an initial point for Newton's method and then depending on which root Newton's method converges to it will be a different color. Then we repeat this action for other points in the complex plane. In this section we will go from $-i$ to i and from -1 to 1 . For this range the step size we will use will be 0.0005 . And figures that will have a smaller range will have as step size of 0.0001 . The step size determines the distance between the chosen initial points. So if we start at $-1 + i$ the next initial point will be $-0.9995 + i$.

2.1 Function with real roots

First let's consider the function $f(z) = z^2 - 1$. As we could see in section 1.4 this function has two real roots $z = 1$ and $z = -1$. From section 1.4 we found that the basins of attraction for the function $f(x) = x^2 - 1$ are $B(1) = (0, \infty)$ and $B(-1) = (-\infty, 0)$. So when we expand this function to the complex plane with the function $f(z) = z^2 - 1$ we would expect that the complex plane would be divided in two halves. Where choosing an initial point in one half Newton's method will converge to the root $z = -1$ and choosing an initial point in the other half Newton's method will converge to the root $z = 1$. And from what we can see in figure 6a this is indeed the case. The complex plane is divided in two halves by the imaginary axis. And on the imaginary axis Newton's method fails. From figure 6b we can see that the closer we get to the boundary between the basins of attraction the longer it takes for Newton's method to converge. This is always the case even for function with more than two roots, which we will see in the next section.



(a) Basins of attraction for $z = 1$ in red and $z = -1$ in blue. (b) The darker shades indicate more iterations are needed for convergence.

Figure 6: Newton basin for $f(z) = z^2 - 1$.

Behavior of Newton's method on the imaginary axis.

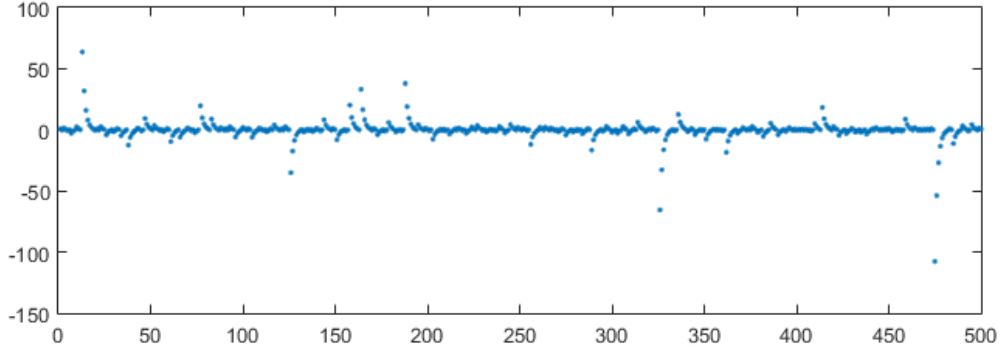
Let's take a closer look at what happens on the imaginary axis. We will do this by taking an initial point on the imaginary axis and then see how it behaves as we iterate the Newton function. As a initial point for Newton's method let's use $z_0 = 2i$. From what we can see in figure 7a the iteration points stay close to each other with some exceptions. Figure 7b gives a histogram with the frequency of the iteration points. Here you can also see that most of the iteration points stay close to zero. This behavior can be explained by rewriting $N(z)$. For $f(z) = z^2 - 1$ we have $N(z) = z - \frac{z^2 - 1}{2z}$. We are interested in the behavior of Newton's method along the imaginary axis so let's take $z = iy$, with $y \in \mathbb{R}$. Then we have

$$N(iy) = iy - \frac{-y^2 - 1}{2iy} = iy + \frac{y^2 + 1}{2iy} = iy - i \left(\frac{y^2 + 1}{2y} \right) = i \left(y - \frac{y^2 + 1}{2y} \right).$$

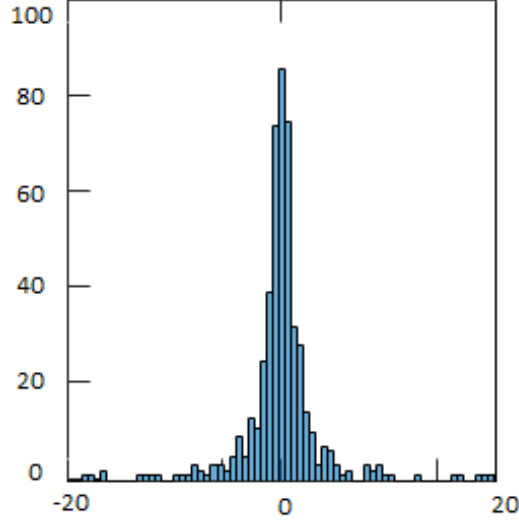
Now let's take a close look at the function $g(y) = y - \frac{y^2 + 1}{2y}$ and see if this function can explain the behavior of Newton's method for $f(z) = z^2 - 1$ along the imaginary axis. Rewriting the formula gives

$$g(y) = y - \frac{y^2 + 1}{2y} = y - \frac{y}{2} - \frac{1}{2y} = \frac{1}{2} \left(y - \frac{1}{y} \right).$$

This explains the behavior because when y is close to zero the term $\frac{1}{y}$ will be large. And for the next iteration y is large and $\frac{1}{y}$ will be small. And because of the $\frac{1}{2}$ term in front the next iteration point will be halved. This behavior then repeats itself until the iteration point will be around zero. This can see in figure 7a. As an example let's say that at some point we have $x = 0.99$ then the next iteration point will be $\frac{1}{2}(0.99 - \frac{1}{0.99}) = -0.0050505$ and the next will be $\frac{1}{2}(-0.0050505 - \frac{1}{-0.010101}) = 49.495$, and the iteration after that $\frac{1}{2}(49.495 - \frac{1}{49.495}) = 24.738$, which is about half of the previous iteration.



(a) Plot of the values that the iteration points take.



(b) Histogram with all the values that the iteration points take.

Figure 7: Behaviour of Newton's method on the imaginary axis for the function $f(z) = z^2 - 1$.

Proving that the imaginary axis is the boundary of the basins

We can even prove in this case that the imaginary axis is indeed the boundary that separates the basins of attraction [5]. We have already seen that $B(1) = \{z : \operatorname{Re}(z) > 0\}$ and $B(-1) = \{z : \operatorname{Re}(z) < 0\}$ and that for the boundary we have $\partial B(1) = \partial B(-1) =: J$, where J is the imaginary axis. Note that $N(J) = J = N^{-1}(J)$, where $N^{-1}(J)$ denotes the set of preimages of J . As we have seen earlier in this section restricting N to the imaginary axis reduces to the 1-dimensional system

$$g(y) = \frac{1}{2} \left(y - \frac{1}{y} \right).$$

By making a change of variable it will be easier to see what happens. Let T be the linear Mobius transformation

$$T(z) = \frac{z-1}{z+1}, \quad \text{with} \quad T^{-1}(u) = \frac{1+u}{1-u},$$

which we consider in \mathbb{C} . Then we obtain

$$R(u) = T \circ N \circ T^{-1}(u) = u^2.$$

Then the roots $z = 1$ and $z = -1$ of the function $f(z) = z^2 - 1$ correspond to the points $u = 0$ and $u = \infty$ under the transformation T . And the imaginary axis J corresponds to the unit circle S^1 under the transformation T . When we iterate the function $R(u) = u^2$ as we would for the Newton function we can see that the basins of attraction for the points $u = 0$ and $u = \infty$ are separated by S^1 . Since if we have $|u| < 1$ it will always converge to the point $u = 0$. And when $|u| > 1$ it will always go to infinity. The function $f(z) = z^2 - 1$ is not the only function with this behavior, every polynomial that has a degree of two will have the same type of behavior. This is because we can change the coordinates of this function and reduce it to $p_\lambda(z) = z^2 - \lambda$. And then we can easily see that for the line $J_\lambda := \{yi\sqrt{\lambda} : y \in \mathbb{R}\}$ we have that $N_\lambda(J_\lambda) = J_\lambda = N_\lambda^{-1}(J_\lambda)$, where N_λ is Newton's method applied to p_λ . Then we will have that J_λ will be the boundary between the two basins of attraction.

2.2 Function with complex roots

From section 1.3 we saw that Newton's method fails if there are no roots as was the case for the function $f(x) = x^2 + 1$. However although some functions have no real roots they do have complex roots. We can find these complex roots by expanding choice of the initial point to the complex plane. We can expand the function $f(x) = x^2 + 1$ to the complex plane by using the function $f(z) = z^2 + 1$. This function has no real roots however it does have two complex roots namely $z = i$ and $z = -i$. Since this is a polynomial of degree two we expect that the complex plane will be divided into two planes one half converging to the root $z = -i$ and the other half converging to the root $z = i$. This is the case as can be seen in figure 8.

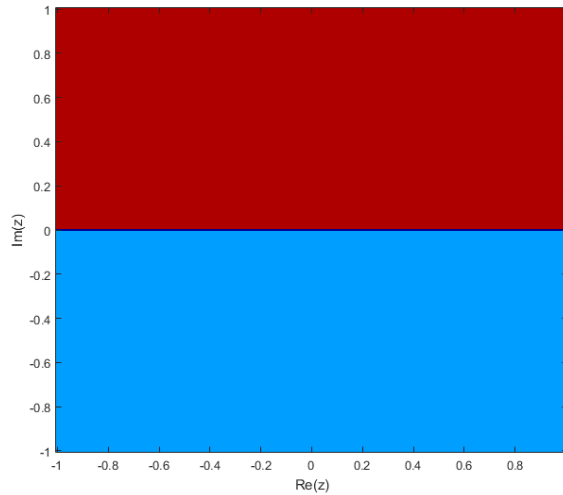


Figure 8: Newton basin for $f(z) = z^2 + 1$. The initial points converging to the root $z = i$ in red and to $z = -i$ in blue.

Behavior of Newton's method on the real axis

The behavior of Newton's method on the real axis for the function $f(z) = z^2 + 1$ is the same as the behavior of Newton's method on the imaginary axis for the function $f(z) = z^2 - 1$. This can be seen by looking at the Newton function. Since we want to look at the behavior at the real line choose $z = x + i0$. Then

$$N(x) = x - \frac{x^2 - 1}{2x} = \frac{1}{2} \left(x + \frac{1}{x} \right),$$

which gives the same type of behavior.

3 Functions with more than two roots

When there are two roots the dynamics of Newton's method is still easy to explain. As we saw in the previous section we can clearly define the boundary between the basins of attraction of the two roots. In this section we discuss what happens when there are more than two roots. Is there still a clearly defined boundary?

3.1 Third degree polynomials

3.1.1 Function with only real roots

First let's consider a function with three real roots. This example is from [1]. Consider the function $f(x) = (x-1)(x-0)(x+1) = x^3 - x$. The roots are $x = -1, x = 0$ and $x = 1$. From figure 9 we can see that there are two critical points. A simple calculation gives that $f'(x) = 0$ when $x = -\frac{1}{\sqrt{3}}$ and $x = \frac{1}{\sqrt{3}}$. So we can conclude that $(\frac{1}{\sqrt{3}}, \infty) \subset B(1)$ and $(-\infty, -\frac{1}{\sqrt{3}}) \subset B(-1)$ since $x = 1$ and $x = -1$ are super attracting fixed points because all fixed points of Newton functions are super attracting.

However determining the basin of attraction of $x = 0$ will be a bit more difficult. From figure 9 we can see that the function is symmetric therefore we expect that there will be cycle of period 2. Since it is a cycle of period 2 we need to solve $x_0 = x_2$. So what we need to solve is $x = N(N(x)) = N^2(x)$. Solving this is simplified because of the symmetry, which means we will get $N^2(x) = x$ if we have $-x = N(x)$. This happens because f is an odd function and therefore $-f(x) = f(-x)$, which means N is also an odd function so $-N(x) = N(-x)$. First we calculate the Newton function,

$$N(x) = x - \frac{x^3 - x}{3x^2 - 1} = \frac{2x^3}{3x^2 - 1}.$$

So the value for the initial point which leads to a cycle of period two is

$$\begin{aligned} -x &= \frac{2x^3}{3x^2 - 1} \\ 0 &= 5x^3 - x \\ x &= \pm \frac{1}{\sqrt{5}} \text{ or } x = 0. \end{aligned}$$

From this we conclude that $(-\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}) \subseteq B(0)$.

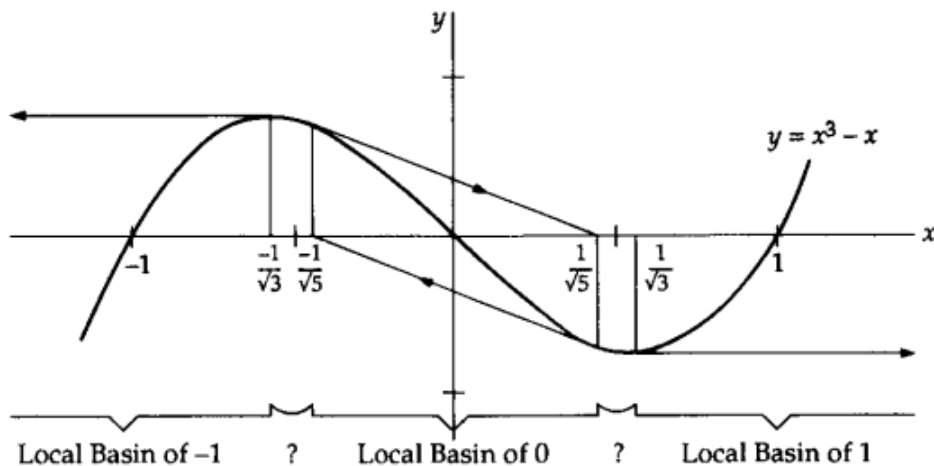


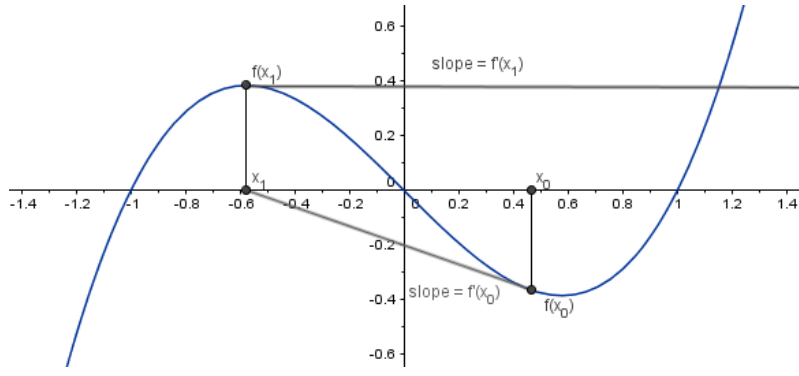
Figure 9: The basins of attraction of Newton's method for the function $f(x) = x^3 - x$. Reprinted from [1].

So all that is left is to determine what happens when we choose the initial point such that $x_0 \in \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{5}}\right)$ or when $x_0 \in \left(\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{3}}\right)$. Since the function is symmetric the interval $\left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{5}}\right)$ has the same behavior as the interval $\left(\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{3}}\right)$. In these two intervals the basins of attraction alternate between $B(-1)$ and $B(1)$ where the length of each basin gets smaller and smaller as can be seen in figure 10.

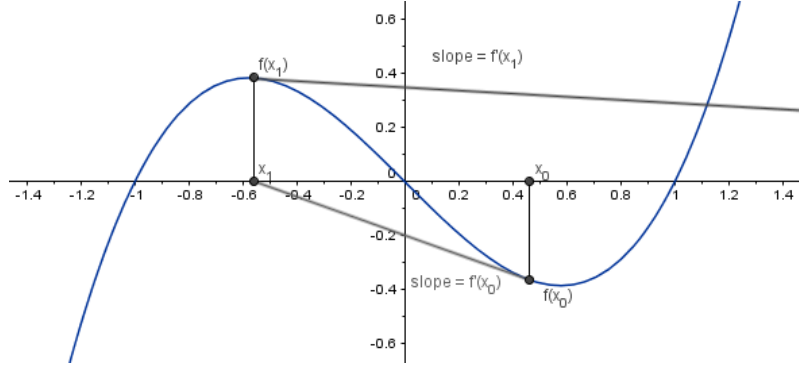


Figure 10: Basin structure in $\left(\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{3}}\right)$ for the function $f(x) = x^3 - x$ (distance is not to scale). Reprinted from [1].

We can explain this behavior by looking at what the behavior of the tangent line is in this interval. When we choose an initial point that is slightly to the left of $\frac{1}{\sqrt{3}}$ then x_1 will be a large negative number. So for this point Newton's method converges to $x = -1$. This means that the chosen initial point will be in $B(-1)$. This will continue to happen until $x_1 = -\frac{1}{\sqrt{3}}$. Then x_1 will coincide with a critical point and Newton's method will fail, which can be seen in figure 11a.



(a) The first iteration point coincides with a critical point.



(b) The second iteration point is in the basin of attraction of the root $x = 1$.

Figure 11: First two iterations of Newton's method for the function $f(x) = x^3 - x$.

Solving

$$x_1 = N(x) = \frac{1}{-\sqrt{3}},$$

gives $x = 0.465601$, so we have

$$\left(\frac{1}{\sqrt{3}}, 0.465601 \right) \subset B(-1).$$

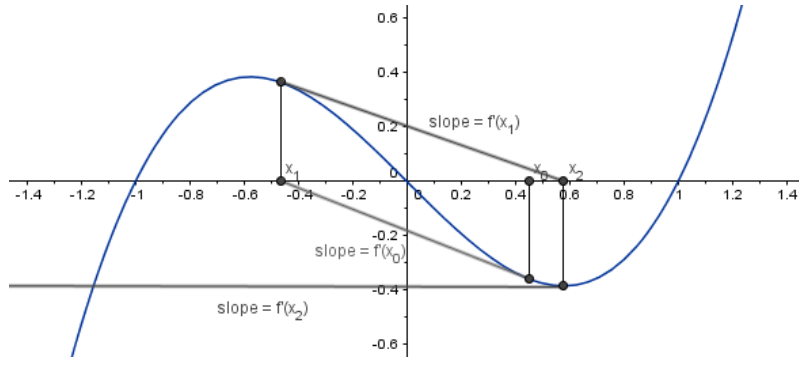
As we choose $x_0 < 0.465601$, $x_0 \in B(1)$ since x_2 will be a large positive number as seen in figure 11b. This will continue until $x_2 = \frac{1}{\sqrt{3}}$ then x_2 coincides with a critical point, which can be seen in figure 12a. Solving

$$x_2 = N(N(x)) = \frac{1}{\sqrt{3}},$$

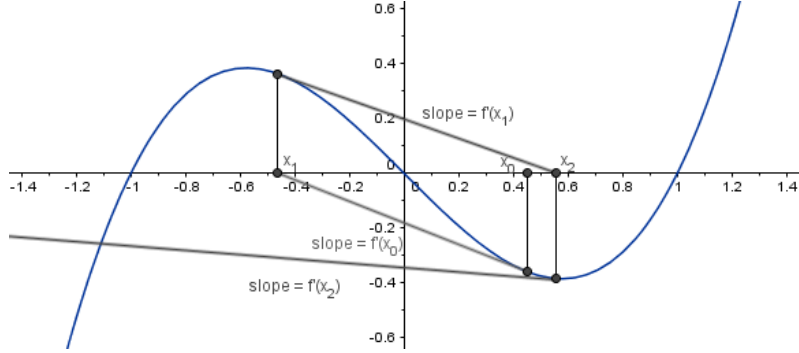
gives $x_2 = 0.4502020$, so we have

$$(0.4502020, 0.465601) \subset B(1).$$

If we choose $x_0 < 0.4502020$ then $x_0 \in B(-1)$ because x_3 will be a large negative number, which can be seen in figure 12b.



(a) The second iteration point coincides with a critical point.



(b) The second iteration point is in the basin of attraction of the root $x = 1$.

Figure 12: First two iterations of Newton's method for the function $f(x) = x^3 - x$.

Continuing this we find a sequence of numbers

$$b_0 = \frac{1}{\sqrt{3}} > b_1 \approx 0.465601 > b_2 \approx 0.4502020 > b_3 > \dots$$

such that

$$(b_i, b_{i-1}) \subset B(-1) \text{ when } i \text{ is odd,}$$

and

$$(b_i, b_{i-1}) \subset B(1) \text{ when } i \text{ is even.}$$

We can determine these numbers by successively solving the equations $N(b_i) = -b_{i-1}$. The first values can be seen in table 1.

i	b_i	$b_i - b_{i+1}$	$(b_i - b_{i-1})/(b_{i+1} - b_i)$
0	0.577350		
1	0.465601	0.111749	7.26
2	0.4502020	0.015399	6.18
3	0.4477096	0.0024924	6.03
4	0.4472962	0.0004134	6.01
5	0.44722736	0.00006884	6.00
6	0.44721589	0.00001147	6.00
7	0.44721398	0.00000191	6.00

Table 1: Lengths of intervals and ratios of lengths of successive intervals. Values from [1].

So the intervals $\left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{5}}\right)$ and $\left(\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{3}}\right)$ are divided into alternating basins of attraction, which keep getting smaller and smaller. From table 1 we can also see that the ratio between the lengths of the basins seem to approach 6. This is indeed the case and we can prove this by first seeing that b_i approaches the value $\frac{1}{\sqrt{5}}$. And we have that the value of the derivative at that point is equal to -6 since

$$N'(x) = \frac{6x^2(x^2 - 1)}{(3x^2 - 1)^2}$$

$$N'\left(\frac{1}{\sqrt{5}}\right) = -6.$$

From the definition of the derivative we have that if b_i and b_{i+1} are any two points close to $\frac{1}{\sqrt{5}}$ then

$$\frac{N(b_{i+1}) - N(b_i)}{b_{i+1} - b_i} \approx -6.$$

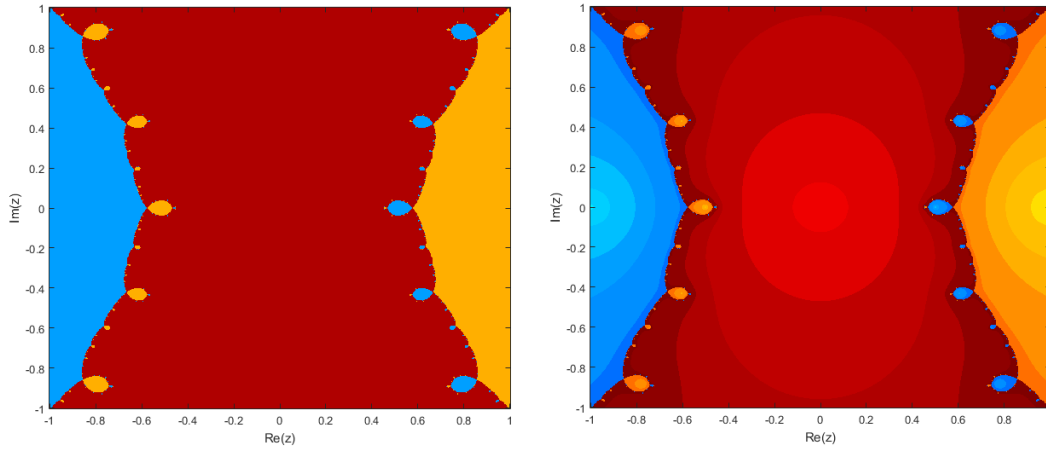
And since $N(b_i) = -b_{i-1}$ we get

$$\frac{b_i - b_{i-1}}{b_{i+1} - b_i} \approx 6.$$

This is why the ratios of the lengths of the basins approaches 6.

Complex plane

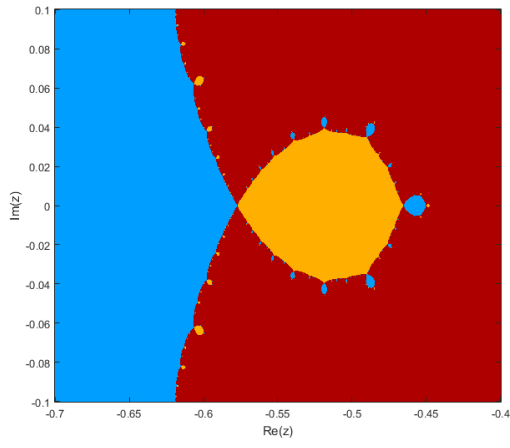
From section 2 we found that if there are two roots then the complex plane is equally divided into two regions. So with this knowledge we expect that something similar happens with the function $f(z) = z^3 - z$. That the complex plane will be divided in three regions where iterates of initial points in each region converge to one of the three roots. However as we can see in figure 13a this does not happen.



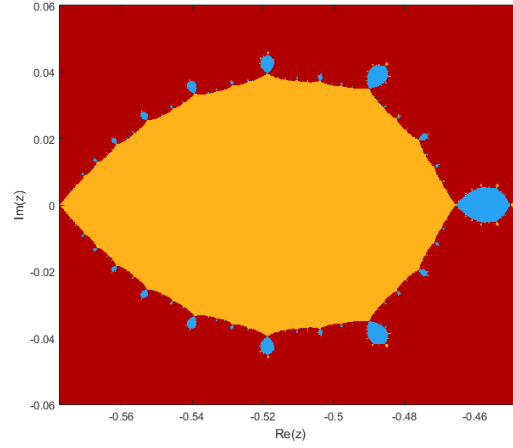
(a) Basins of attraction for the roots $z = -1$ in blue, $z = 0$ in red and $z = 1$ in orange. (b) Basins of attraction with time of convergence, where darker shades indicate more iterations.

Figure 13: Newton basin for $f(z) = z^3 - z$.

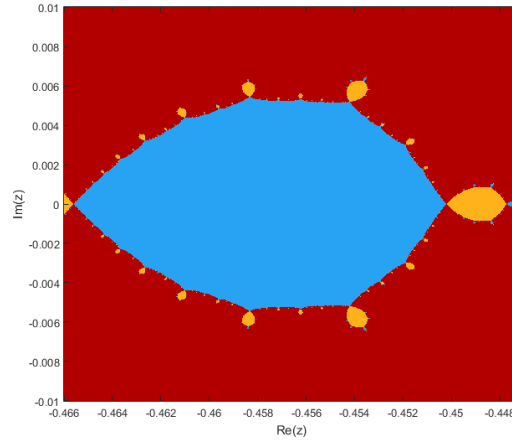
Or rather not entirely. Roughly you can say that there are indeed three regions however along the border Newton's method behaves differently. The cause for this behavior is the symmetry in the function. When only using real number as initial points the basins of attraction were split into three regions. However at the border of these regions, which are the intervals $(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{5}})$ and $(\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{3}})$, Newton's method behaved differently. We found that in these intervals the basins of attraction alternated between $B(-1)$ and $B(1)$ indefinitely. From figure 14 we can see this happening as well in the complex plane. In figure 13b we can see how many iteration it takes for Newton's method to convergence to that particular root. The lighter the color means that there are less iteration needed. From this we can see that it takes more iterations for Newton's method to converge to each root when the initial guess is close a boundary.



(a) Closeup of figure 13.



(b) Closeup of figure 14a



(c) Closeup of figure 14b

Figure 14: Closeup of the region $(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{5}})$, where you can see a fractal pattern emerging.

3.1.2 Function with complex roots

Now let's see what the behavior is for the function $f(z) = z^3 + 1$. This function has three roots $z = -1$, $z = \frac{1}{2} + i\left(\frac{\sqrt{3}}{2}\right)$ and $z = \frac{1}{2} - i\left(\frac{\sqrt{3}}{2}\right)$. The basins of attraction can be seen in figure 15.

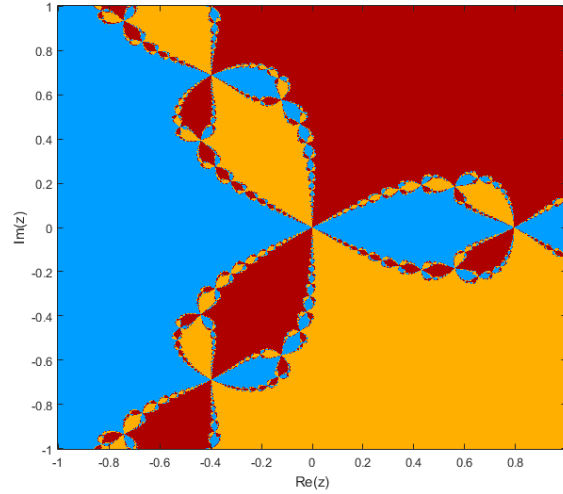


Figure 15: Newton basin for $f(z) = z^3 + 1$.

By taking a closeup we can see in figure 16 that the fractal pattern continues.

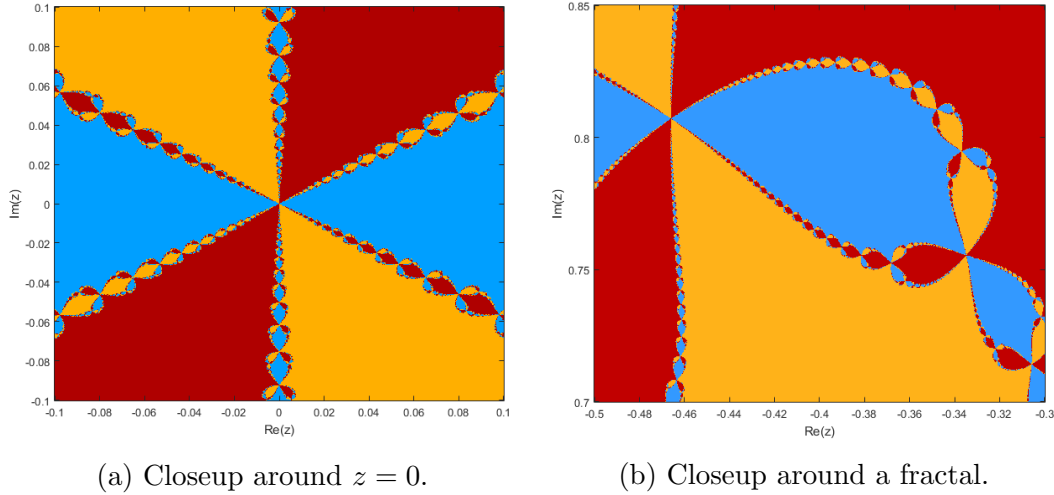


Figure 16: Closeups of figure 15.

We can explain this behavior by looking back closely at what happened in section 2. There we had that the boundaries of $B(1)$ and $B(-1)$ were equal since the boundary for both was the imaginary axis. From [5] we have that this is the case here as well. However since there are three roots in this case we get that each point where two of the basins meet the third does as well. So each point is a tri-border area. We can illustrate this with the following problem. You are asked to color a square completely with three colors. And are given the following condition: when two colors meet all of them meet. So then it is impossible to draw a clear boundary between those colors as you cannot draw any straight lines. The solution for this coloring problem can be seen in figure 15. This is what causes the fractal pattern to emerge.

3.2 Fourth degree polynomials

Now let's look at a function that has four real roots and see if this has similar dynamics as a function with three real roots. The polynomial $f(z) = z^4 - 5z^2 + 4$ has four roots namely $z = -2$, $z = -1$, $z = 1$ and $z = 2$. In figure 17 we can see that the left and the right boundary look similar to the boundaries of the function $f(z) = z^3 - z$. Only now there is an extra boundary along the imaginary axis which has similar behavior as the other boundaries. For all of the boundaries we get alternating basins of attraction which keep getting smaller and this repeats itself indefinitely.

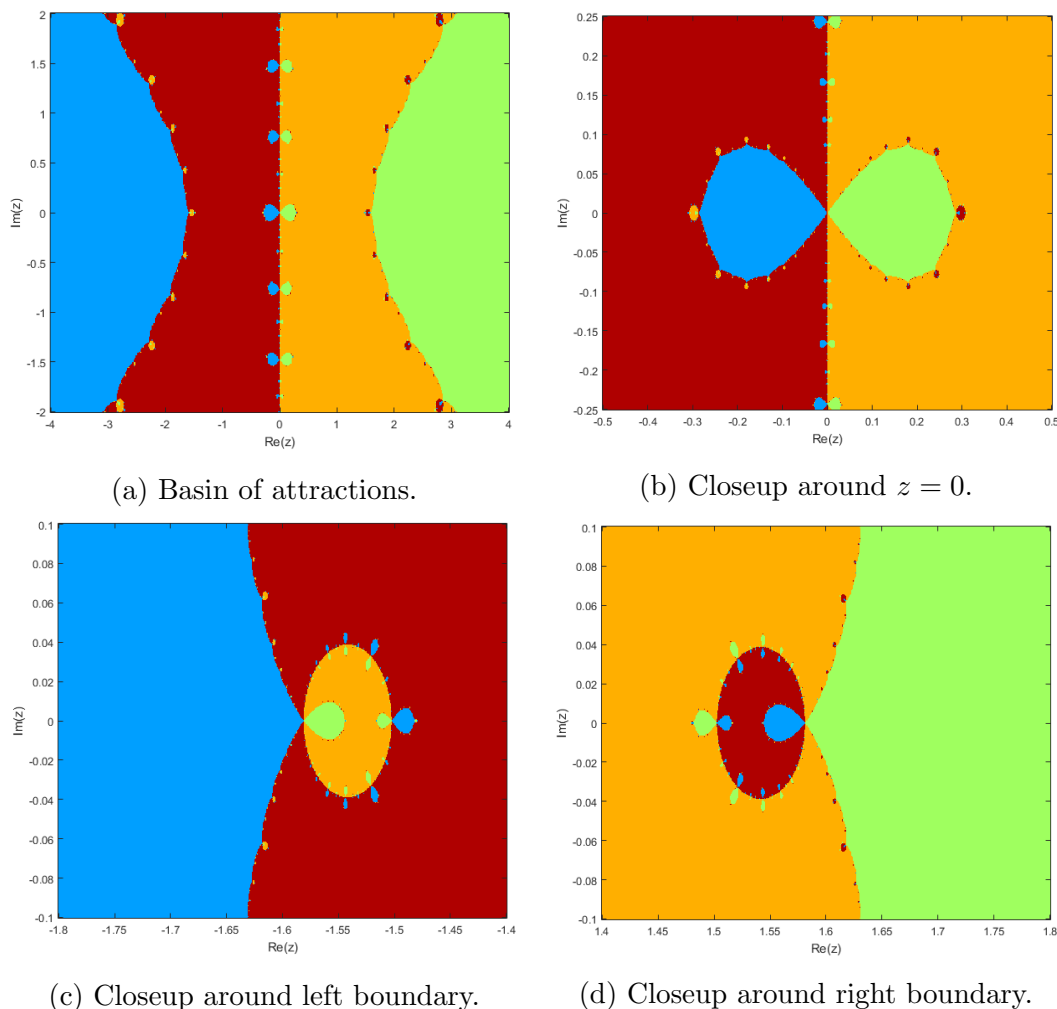


Figure 17: Newton basin for $f(z) = z^4 - 5z^2 + 4$.

Complex roots

The function $f(z) = z^4 + 1$, which has complex roots has the same pattern as a cubic function with complex roots the only difference is that it is split in four as can be seen in figure 18.

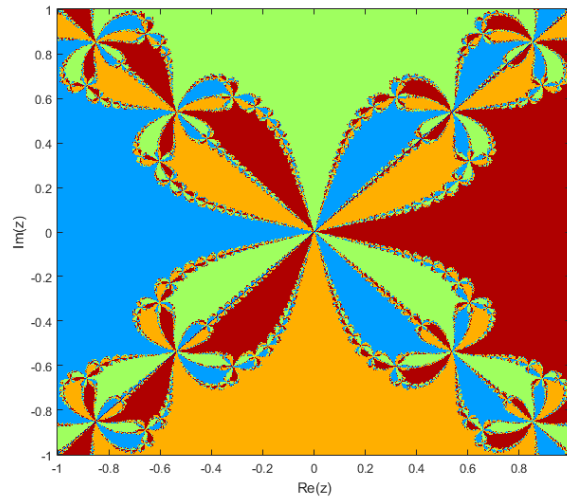


Figure 18: Newton basin for $f(z) = z^4 + 1$.

3.3 Fifth degree polynomials

For the function $f(z) = z^5 + 1$ we get the same pattern as we saw from the functions $f(z) = z^3 + 1$ and $f(z) = z^4 + 1$. Only now it is divided into 5 regions. This can be seen in figure 19.

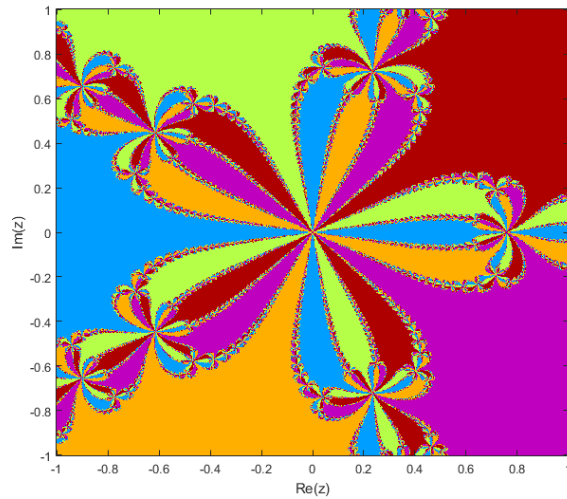
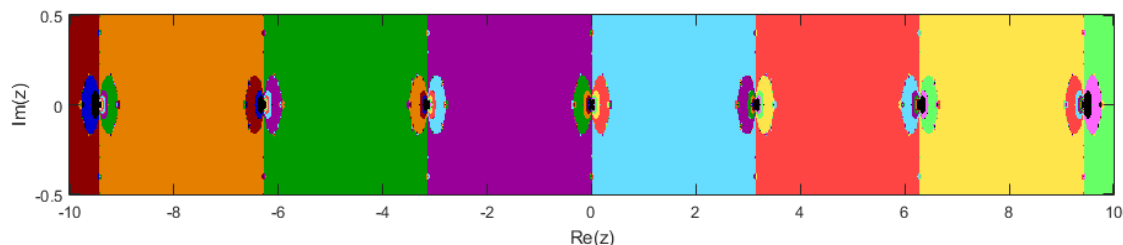


Figure 19: Newton basin for $f(z) = z^5 + 1$.

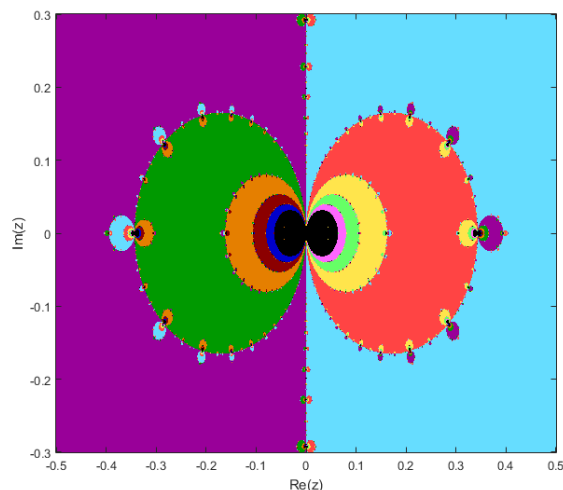
3.4 Trigonometric functions

Another group of functions with interesting behavior when using Newton's method to determine the roots are trigonometric functions. Let's consider the function $f(z) = \cos z$ it has infinitely many roots. The behavior can be seen in figure 20. For this figure we only use the following roots: $z = \pi(n - \frac{1}{2})$ with $n = -4, -3, -2, -1, 0, 1, 2, 3$. From what we can see in figure 20b at the boundary we

have some bigger circles and within those circles are even smaller circles. This continues indefinitely where the circles keep getting smaller and smaller. In figure 20b you can see a black circle, which indicates that Newton's method does not converge. This is not the case Newton's method does converge only it converges to the roots that are further away from $z = 0$. And because this figure was drawn using 10 roots we can only see when Newton's method converges to these roots. Furthermore along the boundary of each of the circles there are more smaller circles and this also continues indefinitely.



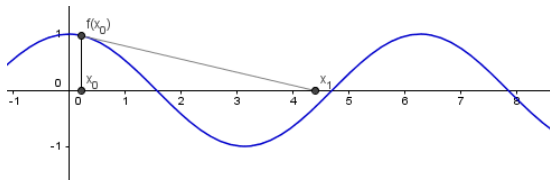
(a) Basin of attraction.



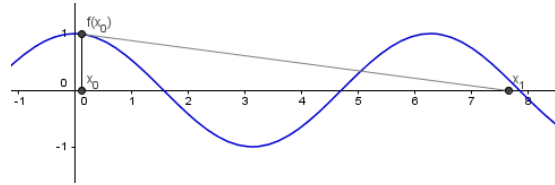
(b) Closeup around $z = 0$.

Figure 20: Newton basin for $f(z) = \cos z$.

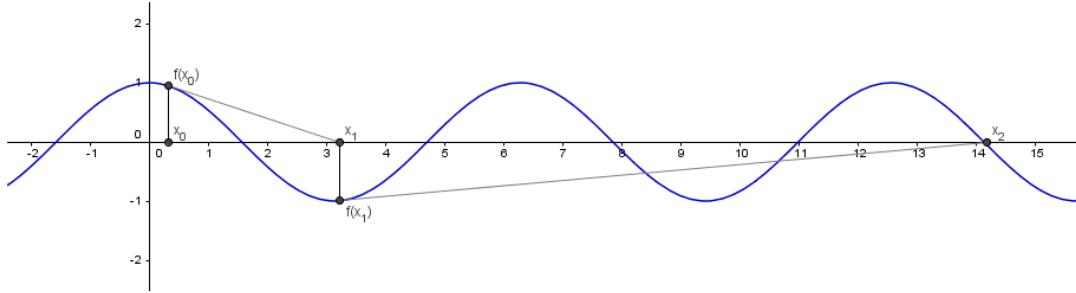
We can explain this behavior by looking at what happens on the real line. Let's look at the boundary between the roots $z = -\frac{\pi}{2}$ and $z = \frac{\pi}{2}$ as seen in figure 20b. The boundary is at $z = 0$, which corresponds to the value of a critical point of the function $f(z) = \cos z$. When we choose an positive initial point close to this critical point we get a large positive value as the next iteration point. And this iteration point is in the basin of attraction of the root that it is close to. Depending on the tangent line at that point we can end up in the basin of attraction of any root, which can be seen in figure 21a and in figure 21b.



(a) First iteration point of Newton's method is in $B(\frac{3\pi}{2})$.



(b) First iteration point of Newton's method is in $B(\frac{5\pi}{2})$.



(c) Second iteration point of Newton's method is in $B(\frac{9\pi}{2})$.

Figure 21: Newton iteration for $f(z) = \cos(z)$.

This explains the bigger circles in figure 20b. However instead of immediately ending up in the basin of attraction of a root we could also end up at a critical point at this point Newton's method fails. Or since the function is symmetric we could end up in a cycle. These points are the boundary between the circles as seen in figure 20b. Or we could end up close to the critical point, which means the next iteration point could end up in any of the basin of attraction of any root as seen in figure 21c. This explains the smaller circles on the boundary of the bigger circles. Of course this behavior repeats itself so if we would zoom in on those circles we would find even smaller circles at the boundary of these circles.

4 Variations of Newton's method

Over the years a lot of variations of Newton's method have been made. The relaxed Newton's method is one of these variations. It improves the convergence of Newton's method for roots that have an order higher than one. To a convergence that is quadratic as Newton's method has a liner convergence for roots that have an order higher than one [7]. Furthermore we will look at the secant method, which can be considered a variation of Newton's method if we use Newton's method to determine the second initial point.

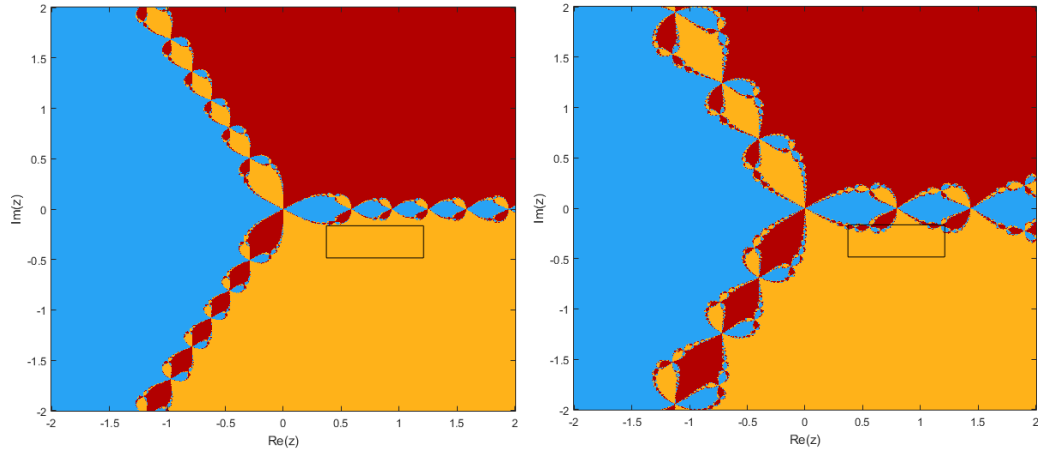
4.1 Relaxed Newton's method

For the relaxed Newton's method to have a better convergence than Newton's method the order of the roots have to be known. If it is known that the function only has a roots of order m then the relaxed Newton's method has a quadratic convergence. Then we can apply Newton's method to $\sqrt[m]{f(z)}$ in order to obtain

$$N_m(z) = z - \frac{mf(z)}{f'(z)}.$$

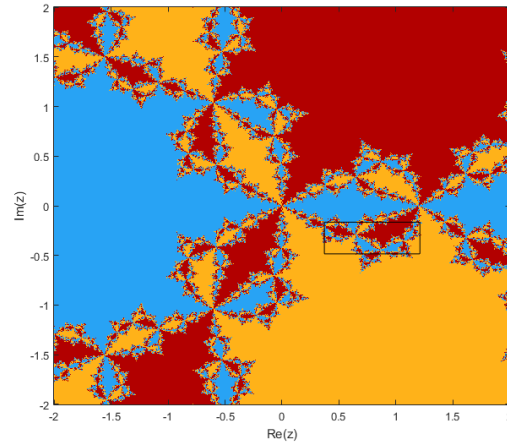
Function with roots of order 1

The roots for the functions that have been used in the earlier sections all had roots of order one. So in those cases the relaxed Newton's method will be the same as Newton's method. However it is still interesting to see what happens to the dynamics when we choose an m that is not equal to the order of the root. In figure 22 we can see choosing a different m leads to different fractal pattern. Furthermore if we choose an initial point anywhere in the square in figure 22a we will always converge to the same root. Choosing an initial point in that same area for $m = 1$ and especially for $m = 1.9$ we can converge to any root. So by choosing a lower value of m we can make the behavior of the relaxed Newton's method less erratic around the boundary.



(a) Relaxed Newton's method with $m = 0.5$.

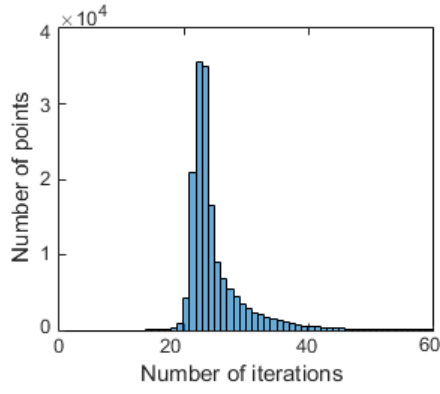
(b) Relaxed Newton's method with $m = 1$ (Newton's method).



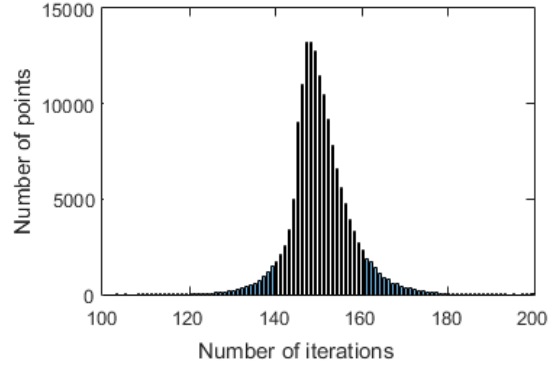
(c) Relaxed Newton's method with $m = 1.9$.

Figure 22: Basins of attraction for $f(z) = z^3 + 1$.

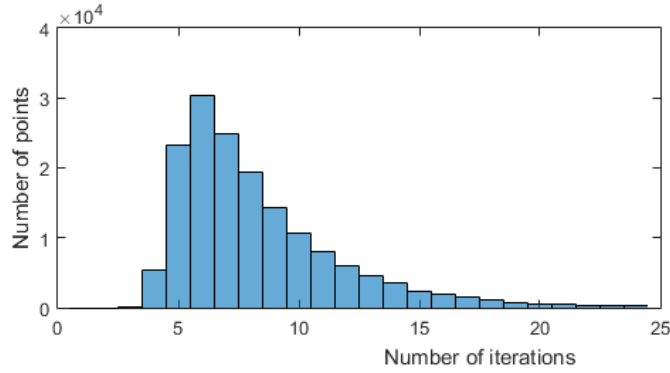
When we look at the number of iterations it takes for the Relaxed Newton's method and Newton's method to converge, which can be seen in figure 23. We can see that when $m = 1.9$ that the number of iterations have increased a lot in comparison with Newton's method. When looking at figure 22a we can see that this corresponds to the more complicated fractal. And choosing $m = 0.5$ the amount of iterations needed for convergence are less than using $m = 1.9$ however it is still more than using $m = 1$.



(a) Relaxed Newton's method with $m = 0.5$.



(b) Relaxed Newton's method with $m = 1.9$.

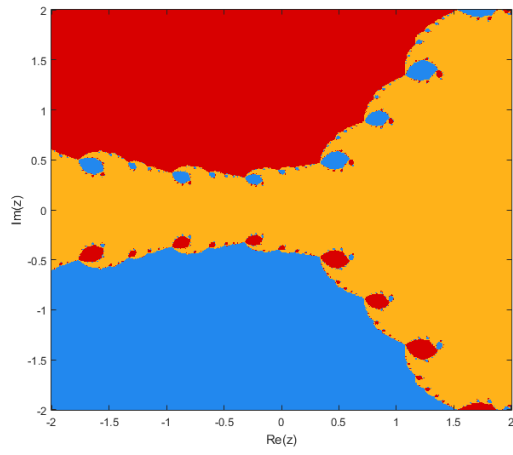


(c) Newton's method ($m = 1$).

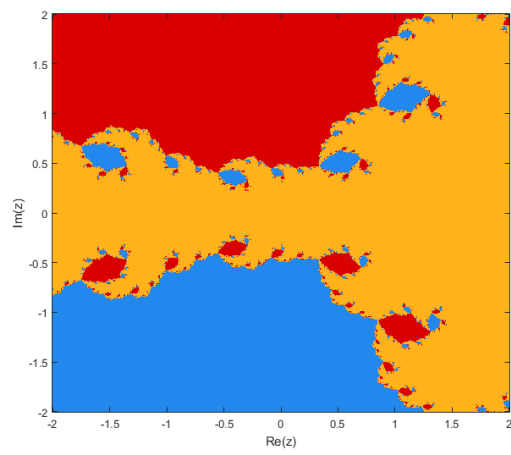
Figure 23: Number of iterations needed for the basin of $f(z) = z^3 + 1$.

Function with roots of order 2

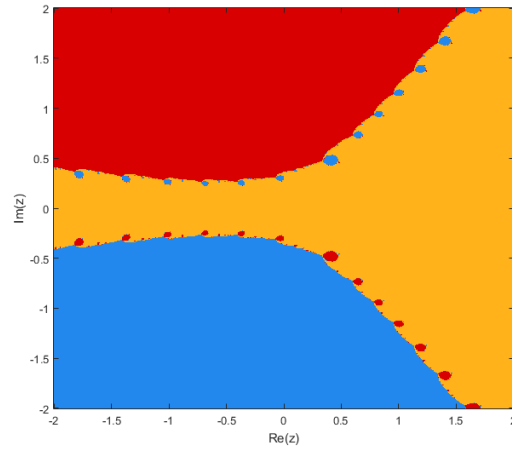
The function $f(z) = (z - i)^2(z + i)^2(z - 1)^2$ has three roots of order two. In figure 24 we can see that choosing a different m leads to different fractal patterns. And as with the case for a function with roots of order one if we increase the value of m we get more erratic behavior around the boundary. And if we use a lower value for m the behavior is less erratic.



(a) Relaxed Newton's method with $m = 2$.



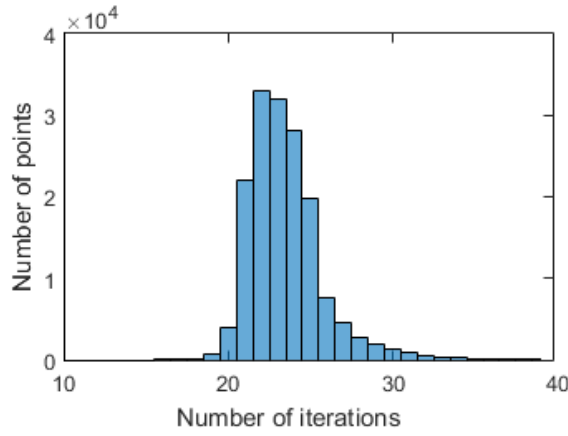
(b) Relaxed Newton's method with $m = 3$.



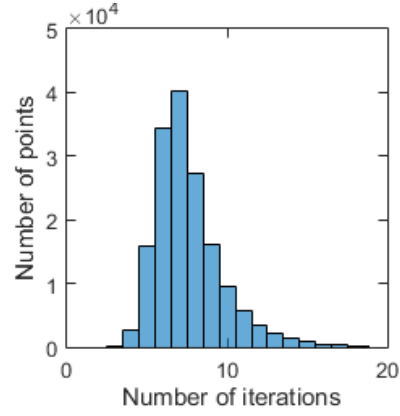
(c) Relaxed Newton's method with $m = 1$ (Newton's method).

Figure 24: Basins of attraction for $f(z) = (z - i)^2(z + i)^2(z - 1)^2$.

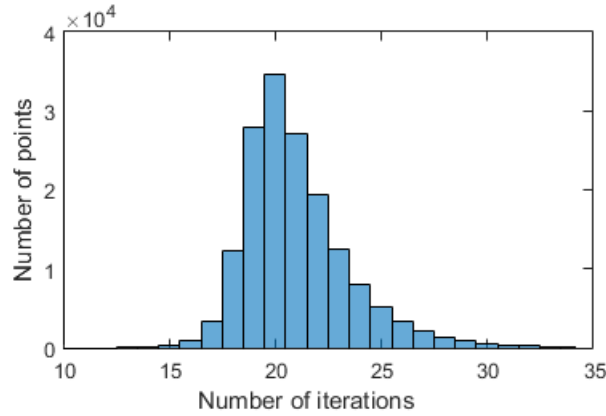
Since we have roots of order 2 the relaxed Newton's method converges faster than Newton's method, which can be seen in figure 25. However choosing a higher value for m does not increase the amount of iterations needed in comparison with a lower value of m . As was the case for the roots that had an order of 1.



(a) Newton's method ($m = 1$).



(b) Relaxed Newton's method with $m = 2$.



(c) Relaxed Newton's method with $m = 3$.

Figure 25: Number of iterations needed for the basin of $f(z) = (z-i)^2(z+i)^2(z-1)^2$.

Function with roots of different order

Now we have seen what happens when the roots have the same order. But what happens when the order is different? The polynomial $f(z) = z(z-1)(z+1)^2$ has three roots: $z = -1$, which has an order of 2, $z = 0$ and $z = 1$, which have an order of 1. When we choose $m = 2$ we can see in figure 26a that only the root with order 2 converges when using 10 iterations. We have to increase the maximum number of iterations to 5100 in order to converge to the roots that have an order of 1. This is when the tolerance=0.01, when we increase the tolerance to 0.001 we need an maximum number of iterations of 500007. So the relaxed Newton's method should only be used when the roots have the same order. Or when we are sure that the initial point is in the basin of the root that has the same value for the order as the value of m that is used.

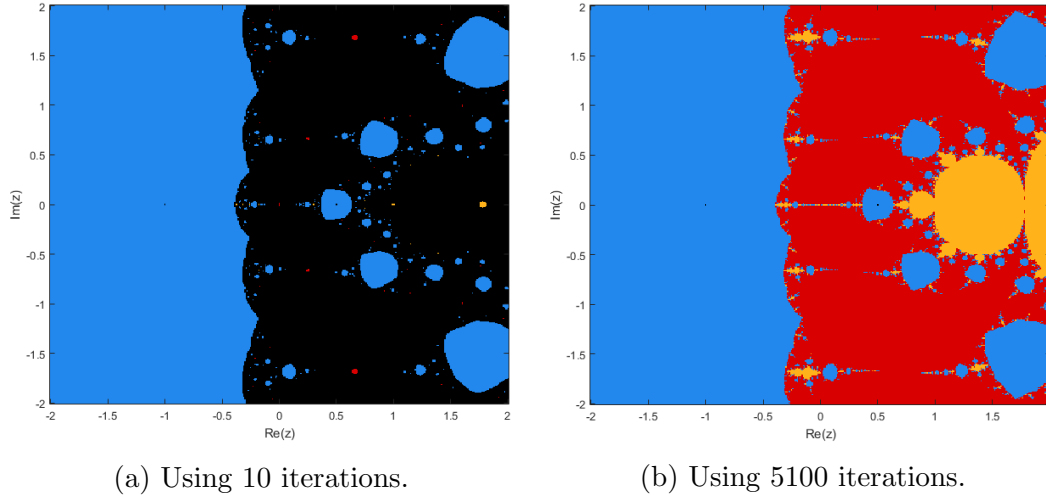


Figure 26: The relaxed Newton's method applied to $f(z) = z(z-1)(z+1)^2$, where the tolerance=0.01.

4.2 Secant method

Newton's method requires the computation of two functions, which when the derivative of the function is complicated can take more computation time. Furthermore Newton's method can fail when the derivative is undefined for some points. In that case the Secant method is a good alternative as it does not require the the derivative. The secant method works in the following way. The first step is to determine two points close to the root let's call them x_0 and x_1 . The second step is to draw a line between these two points and where this line crosses the x -axis is the next iteration point x_2 . Let m be the slope then we can find the point x_2 as follows

$$m = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{-f(x_1)}{x_2 - x_1}.$$

Rewriting this in form of x_2 then gives

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}.$$

Then we use the second step again only this time using the points x_1 and x_2 as the initial points. The iteration stops when $|x_n - x_{n+1}| < \text{tolerance}$, where the tolerance is predetermined. See figure 27 for a visualization of the first few iterations.

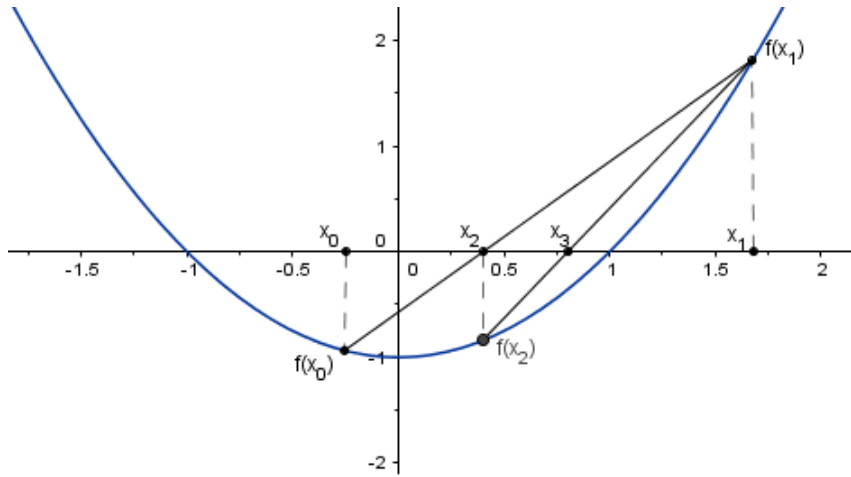


Figure 27: First few iterations of the Secant method for the function $f(x) = x^2 - 1$

The Secant method can be written as

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}.$$

Since the Secant method has two initial points we cannot use the same method to draw the behavior of the Secant method in the complex plane because then we would get a four dimensional figure. However we can look at the behavior of the Secant method for real numbers by letting x_0 be the values of the y -axis and x_1 the values of the x -axis. For the function $f(x) = x^2 - 1$ we can see what happens in figure 28.

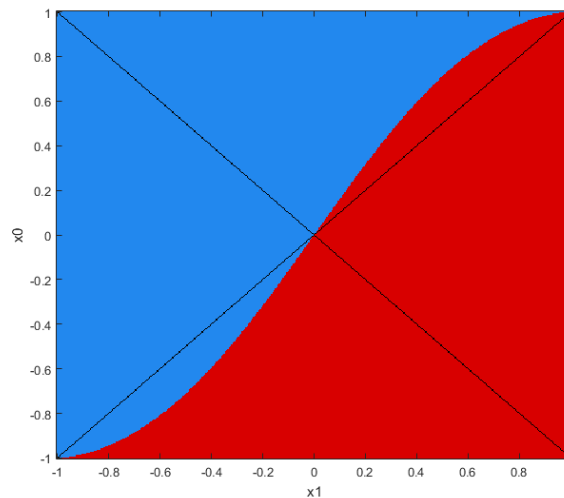


Figure 28: Secant basin for $f(x) = x^2 - 1$, where black indicates that the method does not converge to a root.

We can see that the two lines $y = x$ and $-y = -x$ do not converge, which is because in those cases the initial values x_0 and x_1 are equal and in that case we can't draw a line between those two points.

4.2.1 Newton secant

Although we cannot determine the behavior of the Secant method in the complex plane with all possible initial points x_0 and x_1 . We can still see what happens in the complex plane by only using the initial value x_0 for drawing the figure and then determine x_1 with Newton's method. And then continuing with the Secant method in order to determine the rest of the iterations points. The Secant basin for the function $f(z) = z^3 + 1$ can be seen in figure 29a. Comparing this to the Newton basin in figure 29b we can see that the place of the boundaries are the same only the fractal patterns are different. Also with the Secant method we do not converge to a root in some areas. As we seem to only increase the area of the boundary and have a lot of areas where the secant method fails it is not useful to use the secant method for this function.

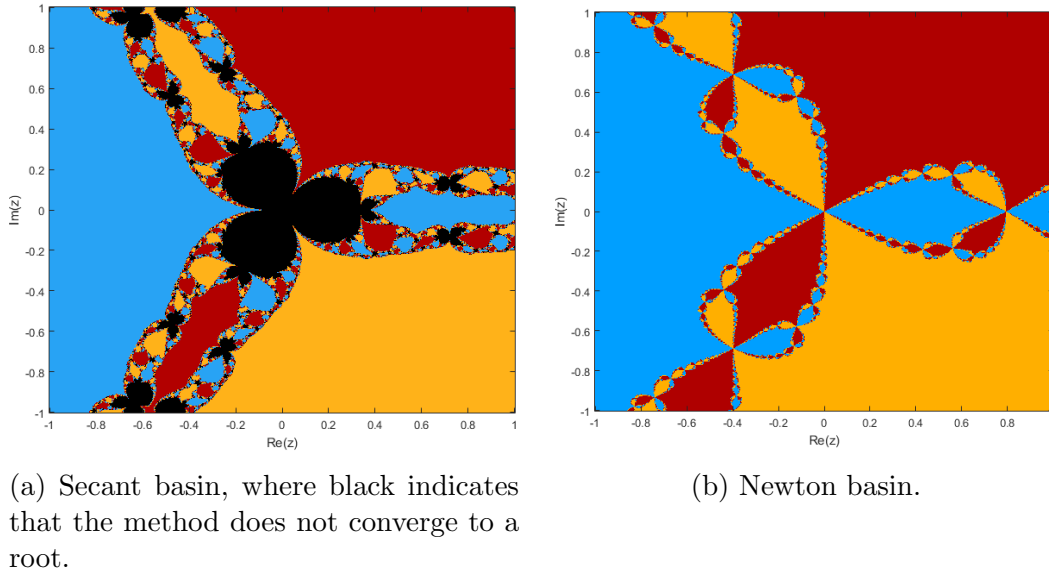
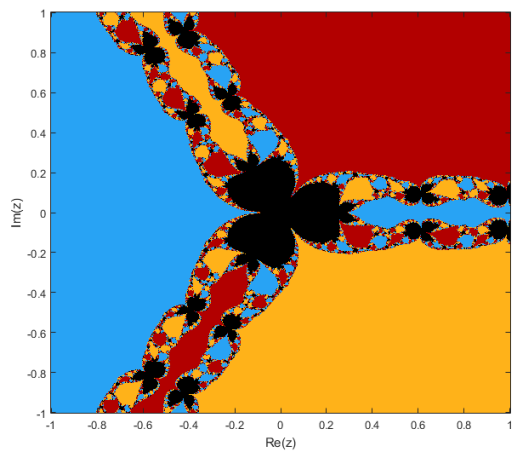


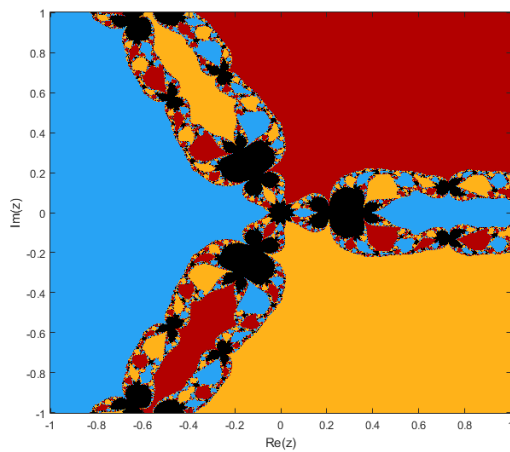
Figure 29: Basins of attractions for $f(z) = z^3 + 1$.

4.2.2 First initial point depends on the second

Another possibility, which is from [4], is by plotting only the x_1 value with $x_1 = a + bi$ where a represents the real axis and b the imaginary axis. And then letting the value of the first initial point depend on the the second. So let $x_0 = a + ad + (b + bd)i$ where $d \in \mathbb{Z} \setminus \{0\}$. In figure 30 we can see that the behavior depends on the value of d and the behavior is similar as when we used Newton's method to determine the other initial point. However by choosing a d with a larger value we can make the areas where the secant method does not converge smaller. This means that if the distance between the points x_0 and x_1 is larger we get a better chance that the secant method converges to a root.



(a) Secant method with $d = 0.5$.



(b) Secant method with $d = 10$.

Figure 30: Secant basin for $f(z) = z^3 + 1$.

5 Conclusion

So we found that if a polynomial has two roots we can still clearly define the boundary between the basins of attraction. However when we increase the number of roots this is impossible because of the fractal patterns that emerge. This is because of the property that the boundaries of the basins of attraction are the same [5]. For three roots this means that for any point where two of the basins of the roots meet the third basin does as well. For trigonometric functions the boundaries are not clearly defined either. This happened because they have an infinite amount of roots and because of repeating structure of the function. Furthermore we found that the relaxed Newton's method and the Secant method have different dynamics than Newton's method. Although the boundaries between the basin of attraction are around the same position the fractal patterns at these boundaries are different. And by choosing a lower value of m when using the relaxed Newton's method we can make the area of the boundary smaller. Finally the Secant method had large areas for when the method does not converge to a root.

6 References

- [1] Philip D. Straffin JR. *Newtons Method and Fractal Patterns*. UMAP Modules: Tools for Teaching 1991.
- [2] R. L. Devaney. *A First Course in Chaotic Dynamical Systems*. Westview Press, 1992.
- [3] W. J. Gilbert. *Generalizations of Newton's method*. 2001.
- [4] M. Szyszkowicz. *Computer art generated by the method of secants in the complex plane*. Computers & graphics, volume 14, issue 3-5, 1990.
- [5] H. O. Peitgen, D. Saupe, and F. V. Haesler *Cayley's Problem and Julia Sets*. The Mathematical Intelligencer, Volume 6, Issue 2, 1984.
- [6] D. S. Alexander, F. Iavernaro, A. Rosa. *Early Days in Complex Dynamics*. American Mathematical Soc., 2012.
- [7] F.V. Haeseler and H.O. Peitgen, *Newton's method and complex dynamical systems*, Acta Appl. Math. Volume 13, 3-58, 1998.

7 Appendix: Matlab code

Main.m

```
clearvars

% Range of the figure
imD = -1;
imU = 1;
reL = -1;
reR = 1;
stepSize = 0.01;

% Functions
f = @(z) z^3+1;
df = @(z) 3*z^2;

% Roots
root1 = -1;
root2 = (-1)^(1/3);
root3 = -(-1)^(2/3);
root4 = -1.5*pi;
root5 = -0.5*pi;

tol = 0.0000001; % The desired tolerance
maxIt = 100; % Maximum number of iterations
method = 0; % 0: Draw the figure with the Relaxed Newton's method
            % 1: Draw the figure with Newton's method
            % 2: Draw the figure with the Secant method (x1 determined
            % with Newton)
            % 3: Draw the figure with the Secant method (x0 determined
            % with x0= a+ad+i(b+bd)
m = 0.5; % Value of m for the Relaxed Newton's method
d = 1; % Value of d for the Secant method

% Determines the roots with one of the root finding methods for all of
% the initial points.
[C,I] = Figure( f, df, tol,maxIt,reL,reR,imD,imU,stepSize, root1,...
    root2,root3,root4,root5,0,method,m,d);

% Creates the figure
x = [reL reR];
y = [imD imU];
ax = gca;
load('MyColormaps','mycmap')
colormap(ax,mycmap)
clims = [1 64];
imagesc(x,y,C,clims)
colorbar
ylabel('Im(z)')
xlabel('Re(z)')
set(gca,'YDir','normal')
```

Figure.m

```
function [ C,I ] = Figure( f, df, tol,maxIt,reL,reR,imD,imU,stepSize,...
    root1, root2, root3,root4,root5,withIter,method,m,d)
%Determines all the roots for all the initial points using one of the
%root finding method.
% INPUT
% f          function of rootfinding problem
% df         the function name or function handle to the derivative of f
% tol        the desired tolerance
% maxIt      maximum number of iterations
% reL
% reR
% imD
% imU
% stepSize   the step size between the initial points
% root1-root5 All the roots of the function
% withIter
% method     0: Draw the figure with the Relaxed Newton's method.
%            1: Draw the figure with Newton's method.
%            2: Draw the figure with the Secant method (x1 determined with
%            Newton).
%            3: Draw the figure with the Secant method (x0 determined with
%            x0= a+ad+i(b+bd)).
% m          Value of m for the Relaxed Newton's method.
% d          Value of d for the Secant method.
% OUTPUT
% C          Array with the color codes for which initial points end up
%            with which root .
% I          Array with the amount of iterations needed for convergence to
%            the roots.
k=0;
l=0;
for b = [imD:+stepSize:imU]
    k=k+1;
    j=0;
    for a = [reL:+stepSize:reR]
        j = j+1;
        if method == 3;
            [rootCalc,flag,convHist,iter] = secant2(f,a,b,d,tol,maxIt);
        end
        if method == 2;
            [rootCalc,flag,convHist,iter] = secantN(f,df,a+i*b,tol,maxIt);
        end
        if method == 1
            [rootCalc,flag, convHist,iter] = newton(f,df,a+i*b,tol,maxIt);
        end
        if method == 0
            [rootCalc,flag, convHist,iter] = newtonRelaxed(f,df,a+i*b,...
                tol,maxIt,m);
        end
        l=l+1;
        I(l)=iter;
    end
end
```

```

    %Determines which root has been calculated
    if abs(rootCalc - root1) <= tol
        c = 6;

    else if abs(rootCalc - root2) <= tol
        c = 26;

    else if abs(rootCalc - root3) <= tol
        c = 36;

    else if abs(rootCalc - root4) <= tol
        c = 46;

    else if abs(rootCalc - root5) <= tol
        c = 56;
    else
        c = 0;
    end
end
end
end
end

if withIter == 1 % If used depending on the amount of iterations
    % needed for convergence the color will be darker shade if
    % more iterations are needed.
    if iter < 10
        c = c+4 ;
    end
    if iter < 15 && iter >=10
        c = c+3 ;
    end
    if iter < 20 && iter >=15
        c = c+2 ;
    end
    if iter < 25 && iter >=20
        c = c+1;
    end
    if iter < 30 && iter >=25
        c = c;
    end
    if iter >=30
        c = c-1 ;
    end
end
C(k,j)=c;
end
end

```


newton.m

Code for Newton's method.

```
function [root, flag, convHist,k] = newton(f, df, x0, tol,...
    maxIt)
% Determines the root for the initial guess x0
% INPUT
% f          function of rootfinding problem
% df         the function name or function handle to the derivative of f
% x0         initial guess
% tol        the desired tolerance
% maxIt      maximum number of iterations
% OUTPUT
% root       root of f
% flag       if 0: attained desired tolerance
%            if 1: reached maxIt nr of iterations
% convHist   convergence history
% k          amount of iterations needed
flag=1;
x=x0;
convHist = zeros(maxIt, 1);

for k=1:maxIt
    % create new x
    xNew=x-(f(x)/df(x));

    % update solution
    update = xNew - x;
    x = xNew;

    % compute error estimate
    convHist(k) = abs(update);

    % check convergence
    if convHist(k) < tol
        flag = 0;
        break
    end
end
root = x;
```

newtonRelaxed.m

Code for the relaxed Newton method.

```
function [root, flag, convHist,k] = newtonRelaxed(f, df, x0, tol,...
    maxIt,m)
% Determines the root for the initial guess x0
% INPUT
% f          function of rootfinding problem
% df         the function name or function handle to the derivative of f
% x0         initial guess
% tol        the desired tolerance
% maxIt      maximum number of iterations
% m          Value of m
% OUTPUT
% root       root of f
% flag       if 0: attained desired tolerance
%            if 1: reached maxIt nr of iterations
% convHist   convergence history
% k          amount of iterations needed
flag=1;
x=x0;
convHist = zeros(maxIt, 1);

for k=1:maxIt
    % create new x
    xNew=x-m*(f(x)/df(x));

    % update solution
    update = xNew - x;
    x = xNew;

    % compute error estimate
    convHist(k) = abs(update);

    % check convergence
    if convHist(k) < tol
        flag = 0;
        break
    end
end
root = x;
```

secantN.m

Code for the Secant method using Newton's method to determine x_1 .

```
function [root,flag,convHist,k] = secantN(f,df,x0,tol,maxIt)
% Determines the root for the initial guess where x1 is determined using
% Newton's method.
% INPUT
% f          function of rootfinding problem
% df          the function name or function handle to the derivative of f
% x0          initial guess
% tol         the desired tolerance
% maxIt       maximum number of iterations
% OUTPUT
% root        root of f
% flag        if 0: attained desired tolerance
%             if 1: reached maxIt nr of iterations
% convHist    convergence history
% k           amount of iterations needed
flag=1;
x1 = newton(f, df, x0, tol,1);
x2 = (x0*f(x1) - x1*f(x0))/(f(x1) - f(x0));

for k=1:maxIt

    x0 = x1;
    x1 = x2;
    x2 = (x0*f(x1) - x1*f(x0))/(f(x1) - f(x0));
    update = f(x2);

    % compute error estimate
    convHist(k) = abs(update);

    % check convergence
    if convHist(k) < tol
        flag = 0;
        break
    end
end
root = x2;
end
```

secant.m

Code for the Secant method where x_0 is determined by x_1 .

```
function [root,flag,convHist,k] = secant(f,a,b,d,tol,maxIt)
% Determines the root for the initial guess where x1 = a+bi and
% x0 = a+ad+i(b+bd).
% INPUT
% f          function of rootfinding problem
% df         the function name or function handle to the derivative of f
% x0         initial guess
% tol        the desired tolerance
% maxIt      maximum number of iterations
% m          Value of m
% OUTPUT
% root       root of f
% flag       if 0: attained desired tolerance
%            if 1: reached maxIt nr of iterations
% convHist   convergence history
% k          amount of iterations needed
flag=1;
x0 = a+d*a+(b+b*d)*i;
x1 = a+b*i;
x2 = (x0*f(x1) - x1*f(x0))/(f(x1) - f(x0));

for k=1:maxIt

    x0 = x1;
    x1 = x2;
    x2 = (x0*f(x1) - x1*f(x0))/(f(x1) - f(x0));
    update = f(x2);

    % compute error estimate
    convHist(k) = abs(update);

    % check convergence
    if convHist(k) < tol
        flag = 0;
        break
    end
end
root = x2;
end
```