LOG(M) FINAL REPORT

RYAN VIG

December 8, 2023

1. Introduction

Definition 1.1. (Root-finding algorithm) A *root finding algorithm* is an algorithm used to find the roots of a function, often seeking to get as close as possible to roots by executing the same set of steps repeatedly. [1].

Our project focuses on explorations of Newton's method, a simple, but effective, root finding algorithm. Given an initial guess x_0, Newton's method will iteratively produce a sequence of values that aim to converge to a root of a function [2]. Geometrically speaking, the subsequent value x_{n+1} is found by constructing the tangent line of f at x_n and setting x_{n+1} to the x-intercept of this tangent line [5]. An example of this process is illustrated below in .

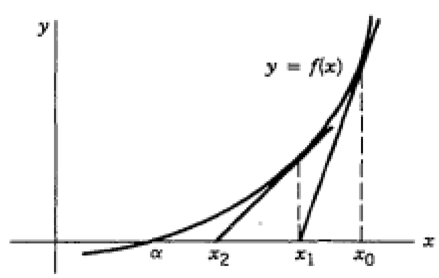


Figure 1. Newton's method on a real-valued function [2]

This process leads to the *Newton iteration formula* given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n \geq 0.$$

Date: December 8, 2023.

In this report, we explore a few of the many applications of Newton's method. Of particular focus will be *Newton fractals*, which are generated when using Newton's method on complex numbers and often create amazing images!

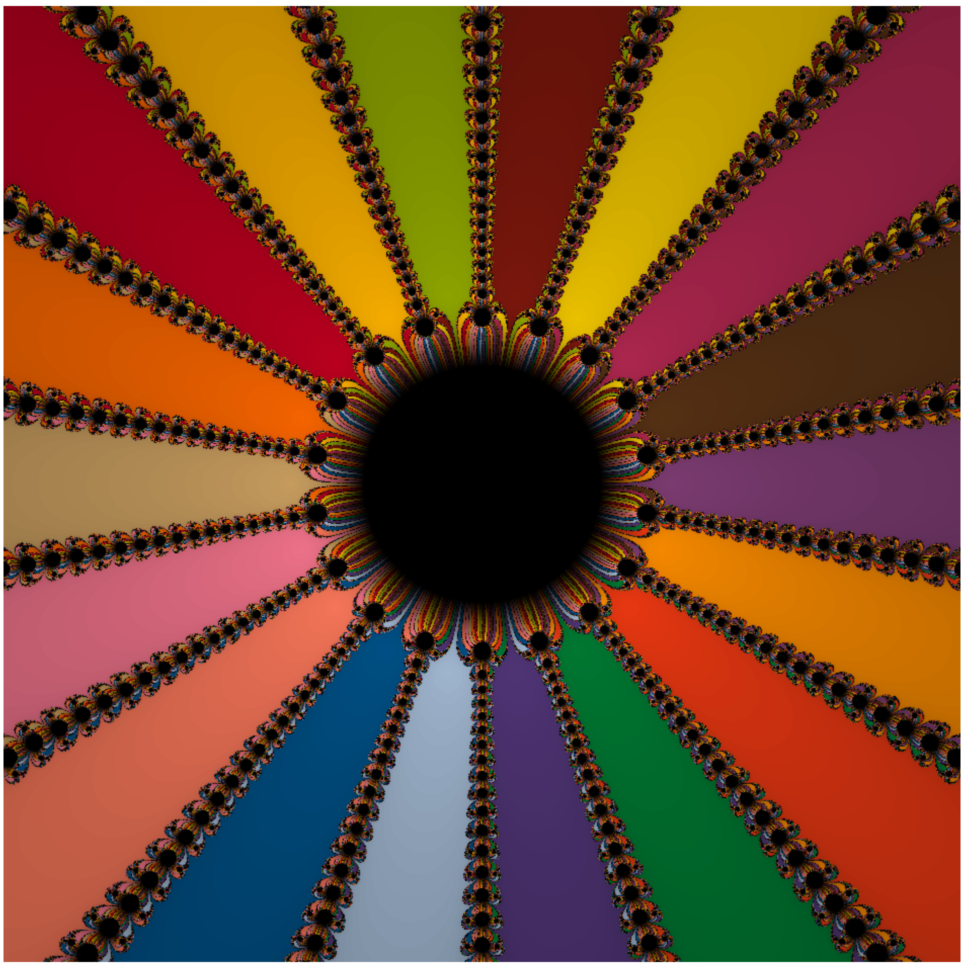


FIGURE 2. Newton fractal of $z^{18} - 1$.

Afterwards, we will take a look at the application of Newton's method to sets with a finite number of elements, which creates interesting structures we call *Newton graphs*.

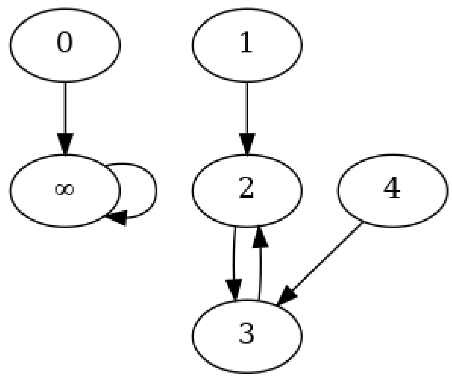


FIGURE 3. Newton graph of $x^2 - 3$ on the set of integers modulo 5.

Before continuing, I'd like to acknowledge that some of the topics discussed here are fairly abstract and may be difficult to understand at first glance. If the notion of a complex number, quaternion, or a finite set of elements is confusing to you, I encourage you to take a look anyways! Many of the finer details of these topics are still lost on me, and I will do my best to explain ideas in broad strokes and provide plentiful amounts of pretty pictures to keep things interesting and (hopefully) understandable.

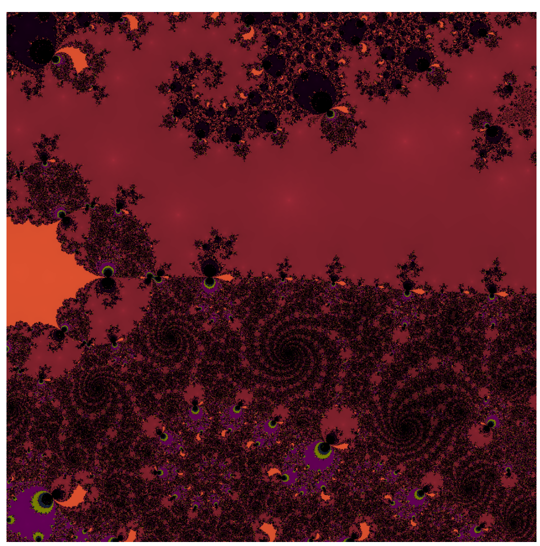


FIGURE 4. Neat-looking Newton fractal I won't pretend to understand.

2. Newton Fractals

2.1. Introduction. Newton fractals typically deal with using Newton's method on functions with complex number inputs. As a reminder, complex numbers are numbers of the form $a + bi$, where a is the "real" part, and b is the "imaginary" part (which is why it's the term next to i). Newton fractals are typically drawn in the *complex plane*, a coordinate system where the x-axis represents the "real" part of numbers, and the y-axis represents the "imaginary" part of numbers. To create the Newton fractals shown in this paper, we run Newton's method for each pixel in the image. Where the pixel's coordinates lie in the complex plane determines the "starting guess" for Newton's method, and the result of running Newton's method from that starting point determines the color of that pixel. The more iterations it takes to get close to a root, the darker the pixel will be shaded. (As a quick aside, we will be using z as our function variable in this section rather than x, as z is typically the variable name of choice when working with complex numbers)

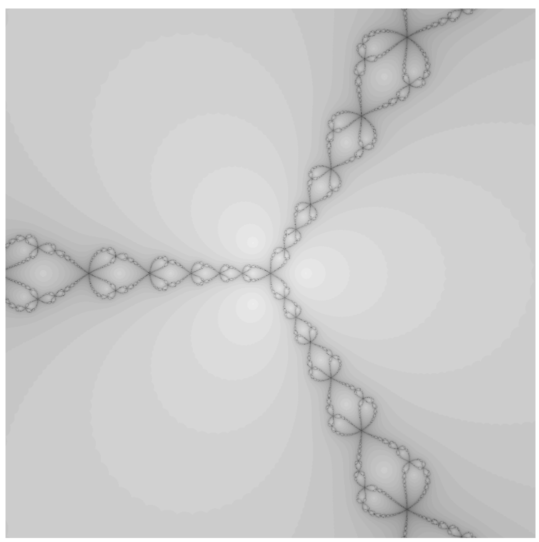


Figure 5. Newton fractal of $z^3 - 1$, shaded by iterations required to converge.

Notice in the above figure that there are 3 points that almost look like the primary "light sources" of the picture. It turns out that these points are where each of the 3 roots of the function are! This follows the intuition that being closer to a root implies faster convergence, but notice that this is not universally true. Even in this picture, the "bulbs" extended from the center appear to break this convention and have their own shading pattern. These bulbs are also what we consider to be the "fractal" part of this image (more on that later), so it's worth considering how the shading of this area and it's fractal nature may be related.

Besides shading, the other factor we typically consider when coloring Newton fractals is hue. In our case (and in most cases), the hue of each pixel corresponds to *which* root of the function we ended up converging to. For instance, the function $z^3 - 1$ has 3 roots, so we might color all the points converging to "root 1" as red, all the points converging to "root 2" as green, and all the points converging to "root 3" as blue. A plot of what that might look like for a few points in the complex plane in shown below.

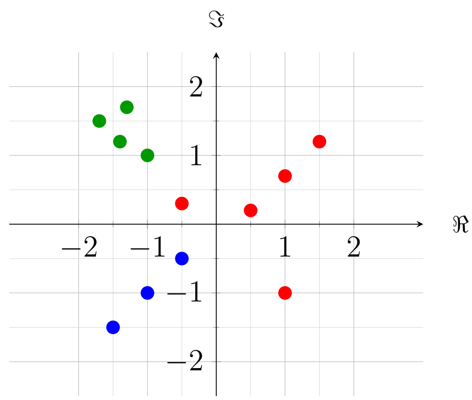


FIGURE 6. Plot of a few points in the complex plane, colored by the root converged to using Newton's method with the function $z^3 - 1$.

Putting our two ideas of shading and hue together, we achieve pictures like the one we see below.

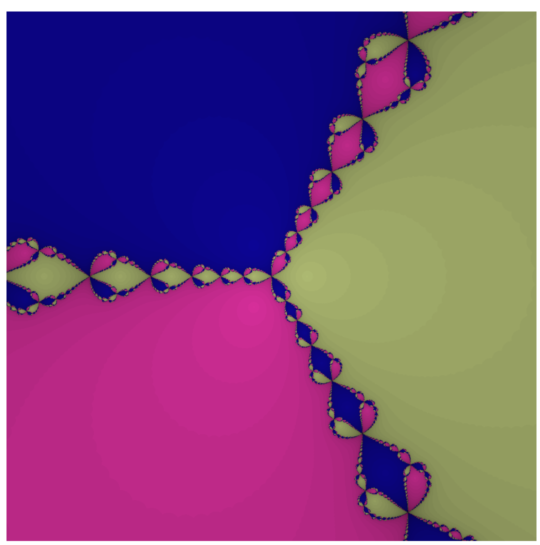


FIGURE 7. Newton fractal of $z^3 - 1$, colored by root reached, shaded by iterations required to converge.

You might be wondering at this point, "What exactly is a fractal? I've heard snowflakes are like fractals, and I think it has something to do with their patterns, but I'm not sure exactly what it means for something to be a fractal." While there isn't any one entirely agreed upon definition of what a fractal is, we generally consider fractals to be geometric structures that have a detailed structure regardless of how much you zoom in, and often are "self-similar", meaning they have patterns that tend to repeat themselves as you zoom in or out [3].

When we perform Newton's method in the complex plane and draw a picture of the result, there are often two main regions the resulting image. One region is the areas in which every point converges to the same root and is thus colored the same, which more formally may be called the "Fatou set". The other region is the areas in which the coloring is very chaotic and non-uniform looking due to the points converging to different roots, which more formally can be called the "Julia set". When we talk about Newton fractals, we are actually referring specifically to the "Julia set" of Newton's method in the complex plane, which are the areas containing the complex patterns with fractal properties that make our pictures so fun to look at. The figure below may be of some use in illustrating this.

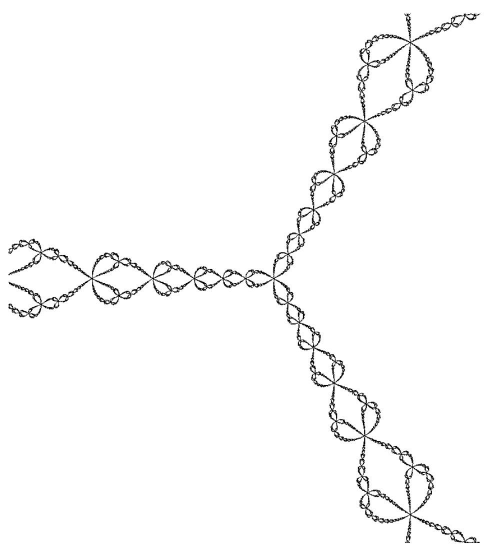


FIGURE 8. Boundary of Newton's method on $z^3 - 1$.

In the image above, we see the result of Newton's method in the complex plane on the same function we used previously, this time simply tracing the "boundary" of the image rather than coloring as we did previously. The area within this trace makes up the boundary or the Julia set of Newton's method on this function that we call the Newton fractal.

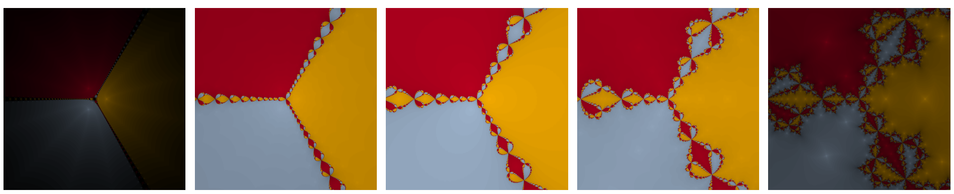
2.2. **Generalized Newton fractals.** Now we'll take a look at what can happen to these fractals when adding some modifications to Newton's method. Recall the typical iteration of Newton's method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Using a constant value a, we can get a generalization Newton's iteration given by

$$x_{n+1} = x_n - a\frac{f(x_n)}{f'(x_n)},$$

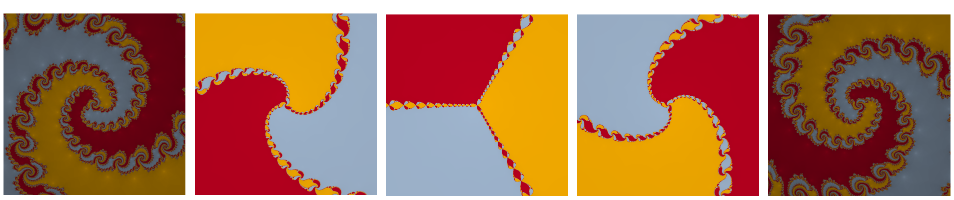
which effectively is just "scaling" the distance we travel between each step of our algorithm. This results in some interesting changes to our pictures! As a decreases, the fractal patterns tend to "smooth" for many functions, and similarly become more complex and "spiky" as a increases [6]. An example of this is shown in the figure below. The caption of each image indicates its respective a value. Another interesting pattern occurs when we give



0.2 0.6 1 1.4 1.8

FIGURE 9. Newton fractals of $z^3 - 1$ captioned by their a value ($a = 1$ in the "normal" case).

a a nonzero imaginary part, which tends to have a "swirling" effect as the imaginary part gets bigger. Shown below is an example of this effect with the caption of each image indicating the imaginary part of its a value. Notice that a negative imaginary part results in a similar effect to the positive imaginary part, but rotates in the opposite direction. For all the images below, the real part of a is fixed at 0.5.



$-0.8i$ $-0.4i$ $0i$ $0.4i$ $0.8i$

FIGURE 10. Newton fractals of $z^3 - 1$ with a values from $0.5 - 0.8i$ to $0.5 + 0.8i$.

If you'd really like to get crazy, you can combine these ideas by using a large real *and* imaginary part for a.

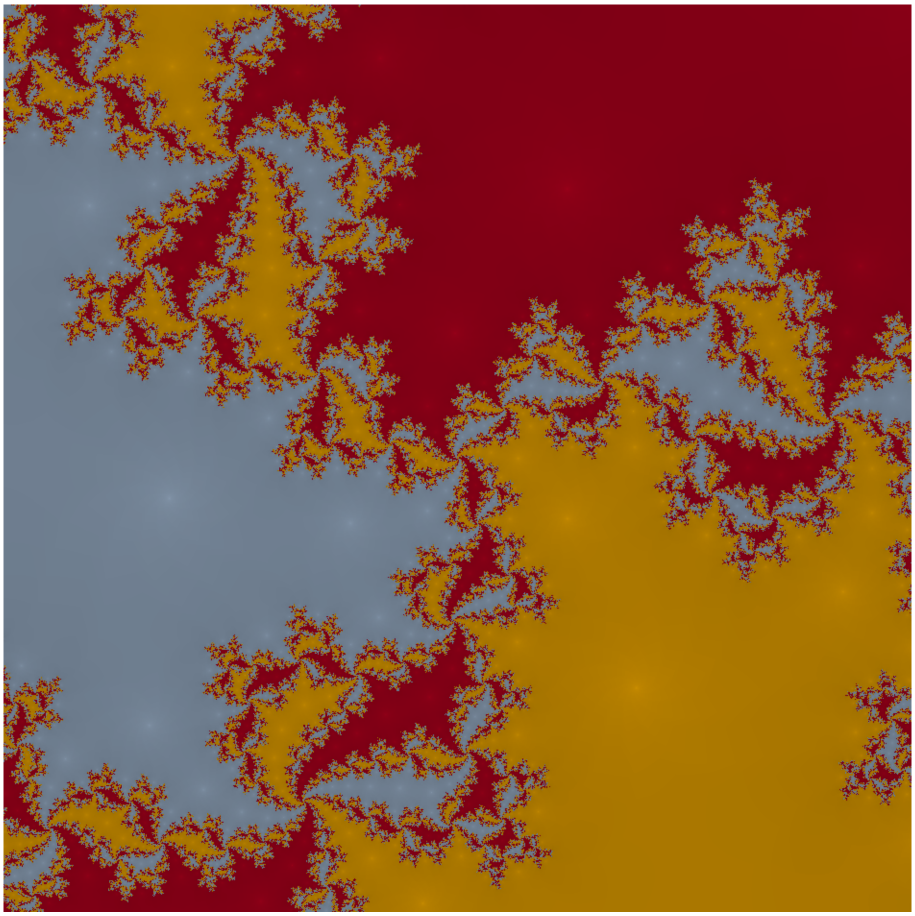


FIGURE 11. Newton fractal of $z^3 - 1$ with $a = 1.8 + 0.5i$

Notice that as our value of a grows increasingly far from its "default" value 1, our resulting images tend to be shaded darker. This is no accident! The darker shading of this images suggests that it takes longer for Newton's method to converge for most points when modifying the distance travelled between each iteration. It turns out that on most functions, Newton's method works very well when $a = 1$ and converges quickly for most initial guesses. By modifying the value of a, we typically cause some "overshooting" or "undershooting" as we iterate that slows down our rate of convergence.

2.3. Attractors in Newton fractals. Another interesting phenomenon can be seen in the effects of *attractors* on Newton fractals.

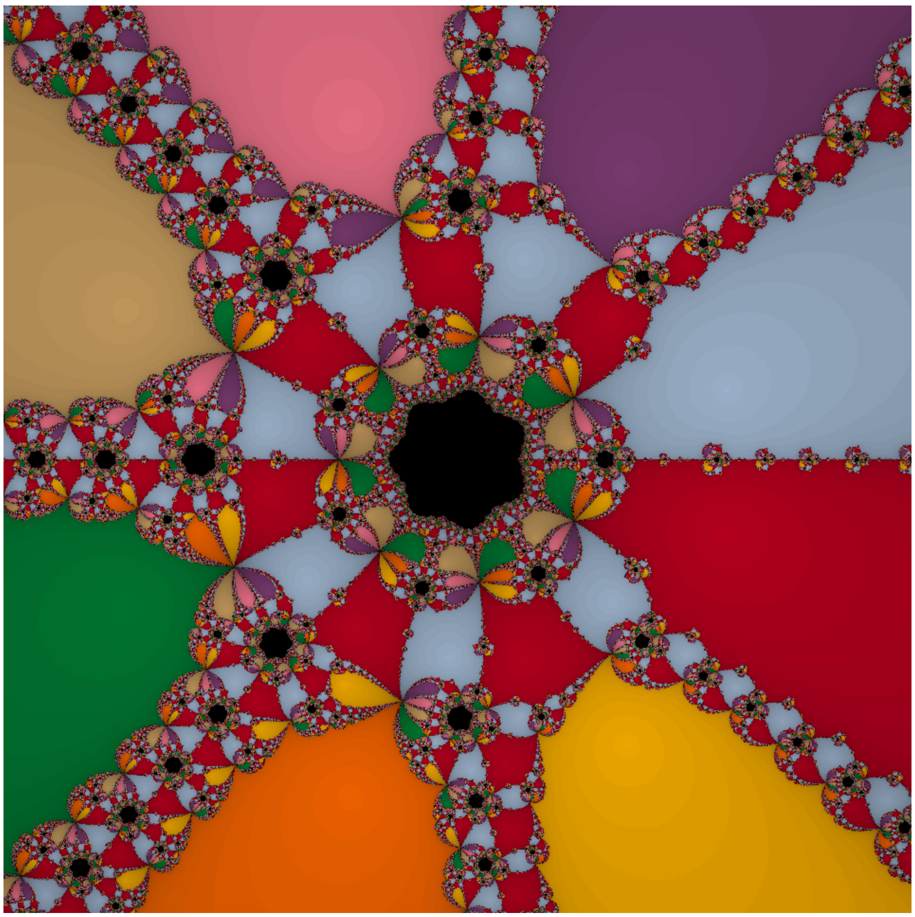


FIGURE 12. Newton fractal of $z^8 - 2z + 2$

Informally, an attractor is a set of points that many other points will "tend towards". The examples we'll look at here are a type of attractor called *periodic orbits*, which are sets of points that act about as their name implies, that is, orbiting periodically [4].

Consider the Newton fractal for the function $z^3 - 2z + 2$.

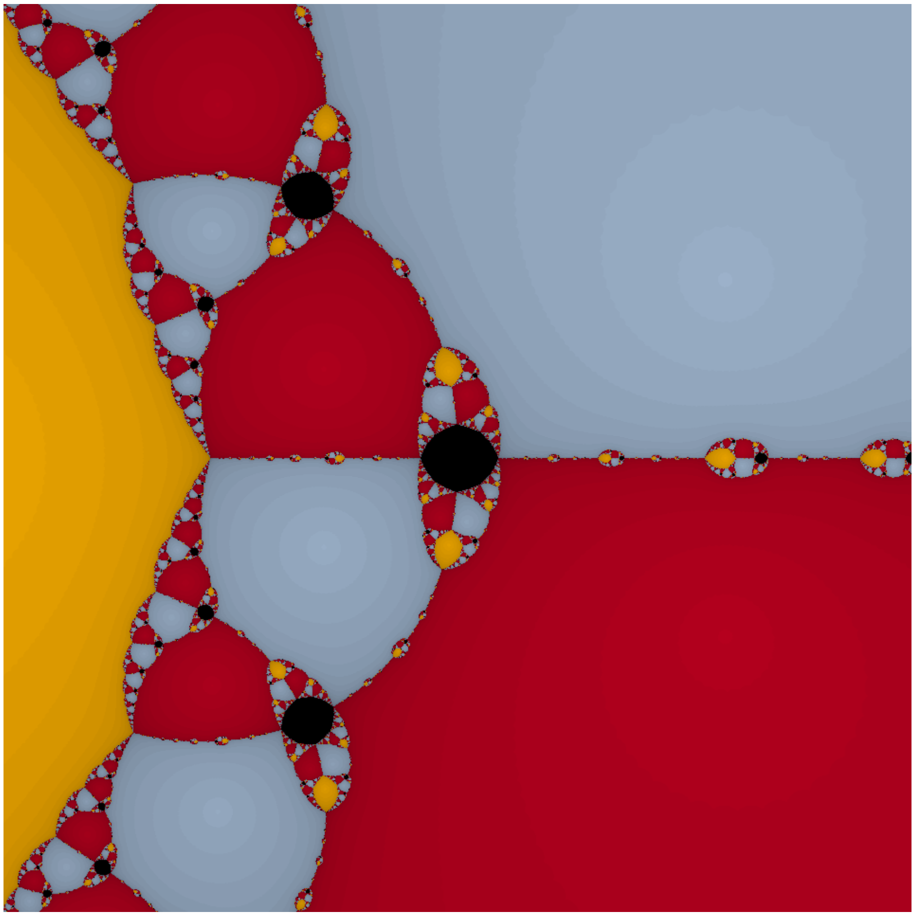


FIGURE 13. Newton fractal of $z^3 - 2z + 2$

Notice the many "black holes" spread throughout the figure, the biggest of which is around 0 (the center of the figure). What's special about Newton's method's on this function is that given the input 0, Newton's method will output 1, and given the input 1, Newton's method will output 0. This means that if Newton's method reaches 0 or 1, it will bounce back and forth between them forever, never converging to a root. Interestingly, inputs near 0 will have outputs near 1 (and vice-versa) and will eventually converge to the cycle between 0 and 1, so other starting points that end up near 0 or 1 will also eventually enter this cycle. Since any point that is "attracted" to this cycle (periodic orbit) never make it to a root of the function, all such points will be colored black in our drawings of the Newton fractals, resulting in the black holes we see in these figures.

It's worth noting that the cycles these points attract to can be more than 2 points as well. The figure below is an example of a Newton fractal containing a cycle between 19 points!

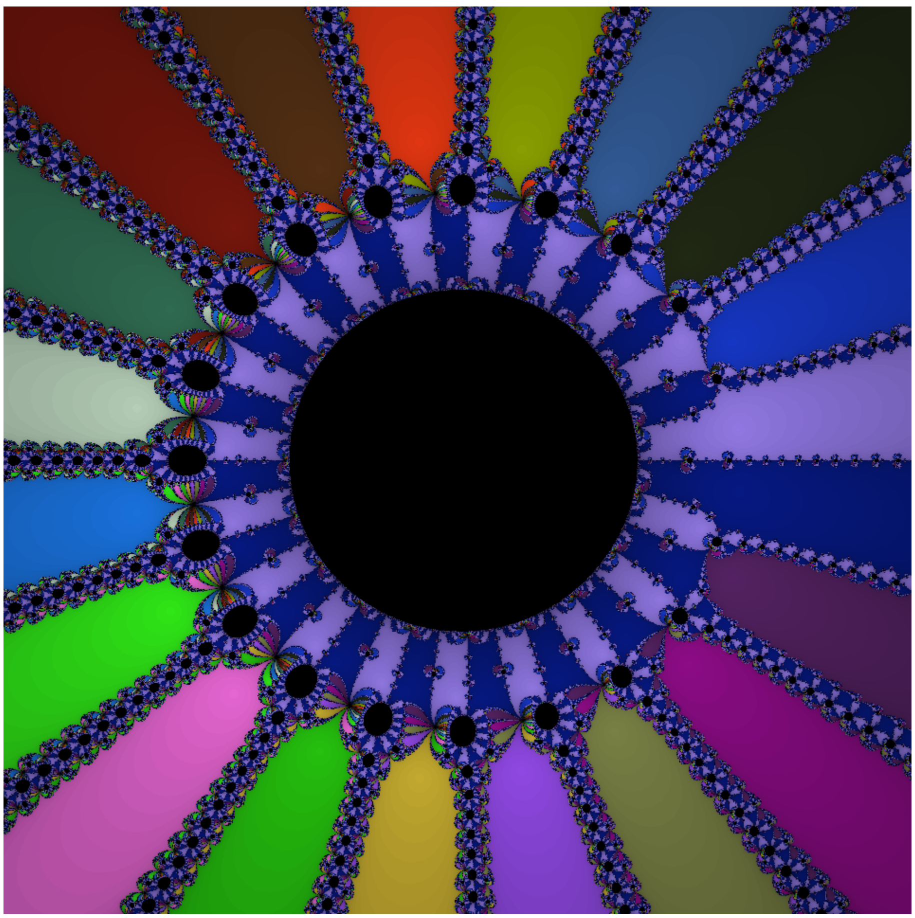


FIGURE 14. Newton fractal of $z^{20} - 2z + 2$

Functions with attracting cycles like this when performing Newton's method provide interesting examples of where Newton's method can fail entirely to converge to a root for certain initial guesses.

2.4. **Additional resources.** If you've made it this far, thanks for taking a look! This is the end of the section on Newton fractals, but if you're still interested, feel free to check out the fractal visualizer that made the examples in this section possible at this link: Newton fractal visualizer. With this visualizer, you can specify the function whose Newton fractal you'd like to explore and customize the parameters used by Newton's method and the appearance of the image. This visualizer is usable on both desktop and mobile devices, though some features are currently limited on the mobile version for formatting purposes. Make sure to use z as your function variable instead of x!

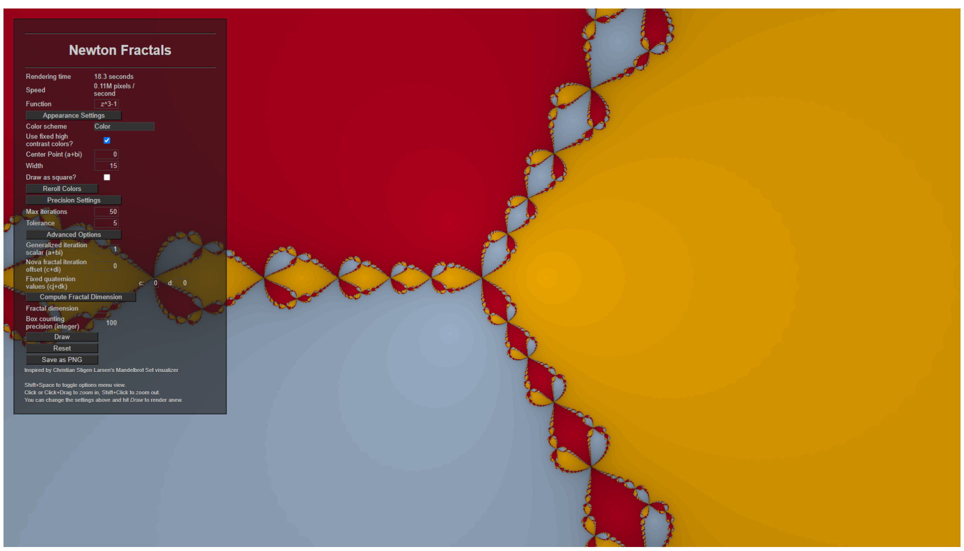


FIGURE 15. Newton fractal visualizer (example is $z^3 - 1$)

3. Newton Graphs

3.1. **Introduction.** We now turn our attention to the application of Newton's method to an entirely different domain that produces interesting structures we call *Newton graphs*.

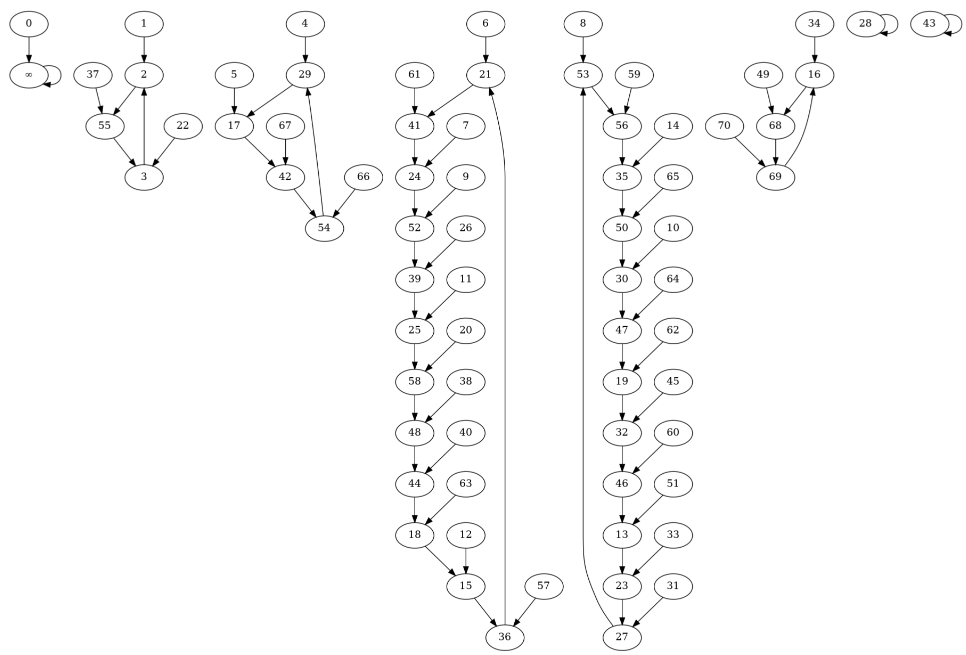


FIGURE 16. Newton graph of $x^2 - 3$ on the set of integers modulo 71

Similarly to Newton fractals, Newton graphs are a figure formed by applying Newton's method to a set of numbers and visualizing the results. However, unlike Newton fractals, Newton graphs deal with *fields* that have a *finite* number of elements, as compared to fields like the complex numbers (what Newton fractals use) that have (uncountably!) infinitely many elements. What a field is is a bit of an abstract concept, but generally fields can be thought of as any set whose basic operations (addition, subtraction, multiplication, and division) behave in the same ways they do for real numbers. In our case, we'll focus on the integers modulo p where p is any prime number, which a commonly studied type of finite field. The integers modulo p can be thought of as the integers, but where each element's value is determined by its remainder when divided by p. Thus, the only "distinct" values of this set are $0, 1, ..., p-1$ since any remainder plus any multiple of p is considered to be the same element in this set. This property is what makes this set "finite", since it only contains p unique elements. Since these sets are finite, we can consider the result of *every* possible input for Newton's method on a given function! Let's first consider the function $x^3 + x + 1$ on the integers modulo 2. The only remainders when dividing integers by 2 are 0 and 1, so these are the only elements in our set. Using our iteration formula for Newton's method with 0 as our input gives us

$$0 - \frac{0^3 + 0 + 1}{3(0)^2 + 1} = 0 - 1 = -1 = 1,$$

noting that -1 is the same as 1 modulo 2 (has the same remainder when divided by 2). Then for the input 0, we have received the output 1. To represent this visually, we can represent the elements 0 and 1 as nodes, and draw an arrow from the node 0 to the node 1 to show that 0 goes to 1 when iterating Newton's method with this function. Now let's see what happens

$$0 \longrightarrow 1$$

when we have 1 as our input.

$$1 - \frac{1^3 + 1 + 1}{3(1)^2 + 1} = 1 - \frac{3}{4} = 1 - \frac{1}{0} =?.$$

Here we ended up dividing by zero, uh oh! So, we need to decide how we're going to deal with division by zero, which we often consider to be undefined. In our case, we say that this output *diverges to infinity*, so our input 1 will yield the output ∞. This means that we need to add the infinity element to our set, so the set we're dealing with is really the integers modulo 2, plus infinity. Now, we can have infinity as another node, and draw an arrow from 1 to infinity since our input 1 yielded infinity as its output.

$$1 \longrightarrow \infty$$

Given the nature of our iteration formula, we know that infinity must always go to itself, so after drawing that arrow, we achieve our completed Newton graph for $x^3 + 1$ on the integers modulo 2 (plus infinity).

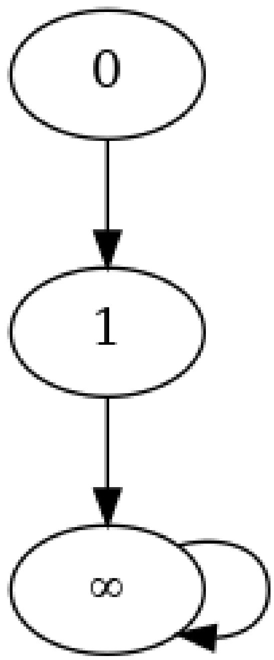


FIGURE 17. Newton graph of x^3+x+1 on the set of integers modulo 2

3.2. **Some common properties of Newton graphs.** With Newton graphs, we can identify where Newton's method converges by seeing which nodes lead to a node that goes to itself, as the iteration formula implies that a node n going to itself is a root since $f(n)$ must be 0. In some cases like our above example, we never converge to a root since we always end up diverging to infinity! In other cases, we might have cycles preventing convergence similar to the cycles we saw with Newton fractals.

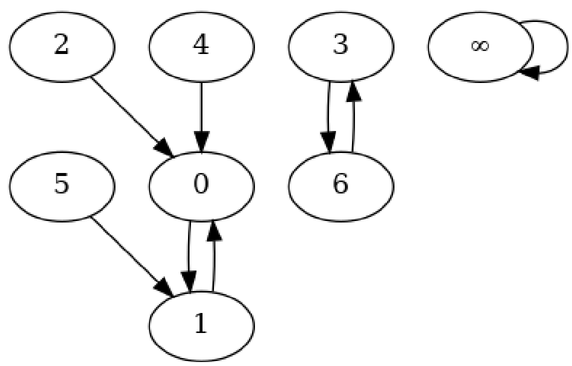


FIGURE 18. Newton graph of x^3-2x+2 on the set of integers modulo 7 (notice the cycle between 0 and 1 we also saw when using complex numbers!)

We also see cases where everything converges and Newton's method "succeeds" for all values (unfortunately for Newton's method, this is rarely the case with finite fields).

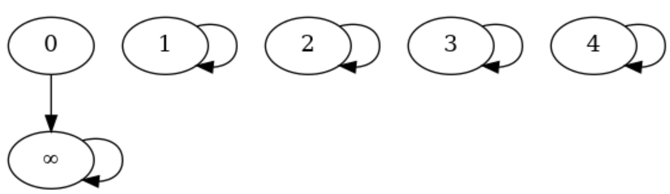


FIGURE 19. Newton graph of $x^4 - 1$ on the set of integers modulo 5

As we grow the number of elements in our set, we see intricate structures form that prompt interesting questions about the patterns that emerge when performing Newton's method in this domain.

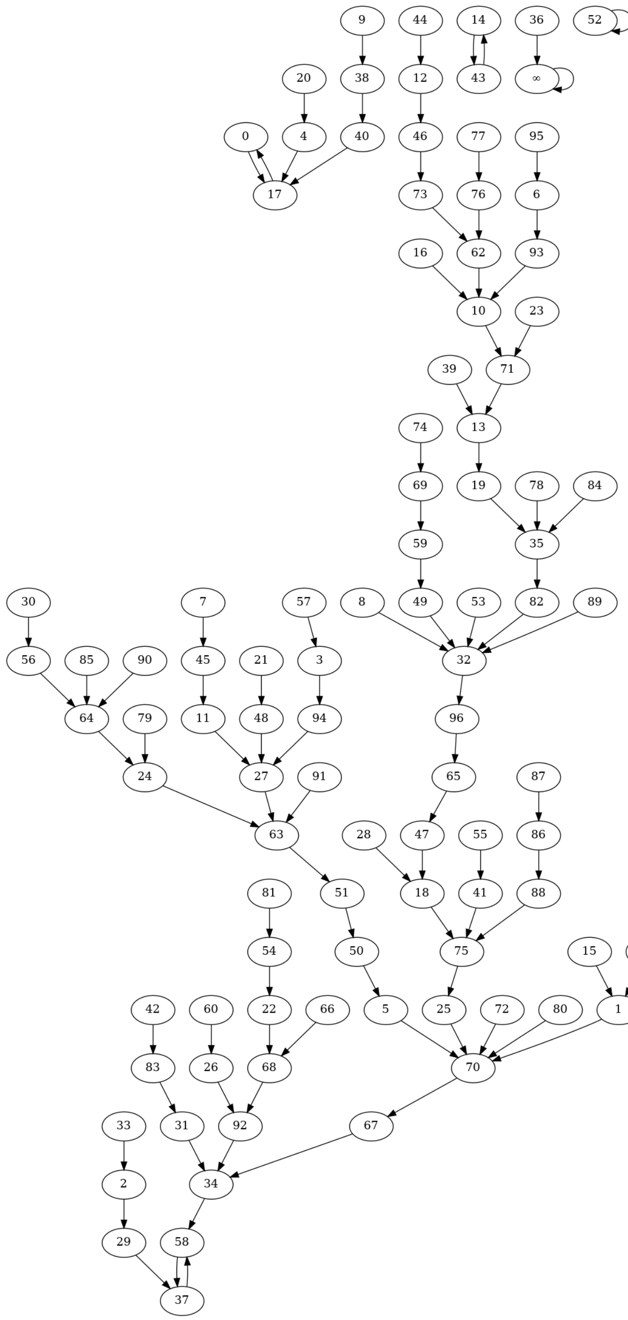


FIGURE 20. Newton graph of $x^5 + 2x^4 - 6x + 5$ on the set of integers modulo 97

3.3. **Additional resources.** Thanks for taking a look at this section on Newton graphs! If you're interested in exploring further or learning more, feel free to check out our interactive Newton graph visualizer at this link: Newton graph visualizer. With this visualizer, you can specify a polynomial and a positive integer n to explore the Newton graph of that polynomial on the integers modulo n. You can zoom in and drag the nodes too if you'd like to take a closer look or reposition some nodes! This visualizer is usable on both desktop and mobile devices.

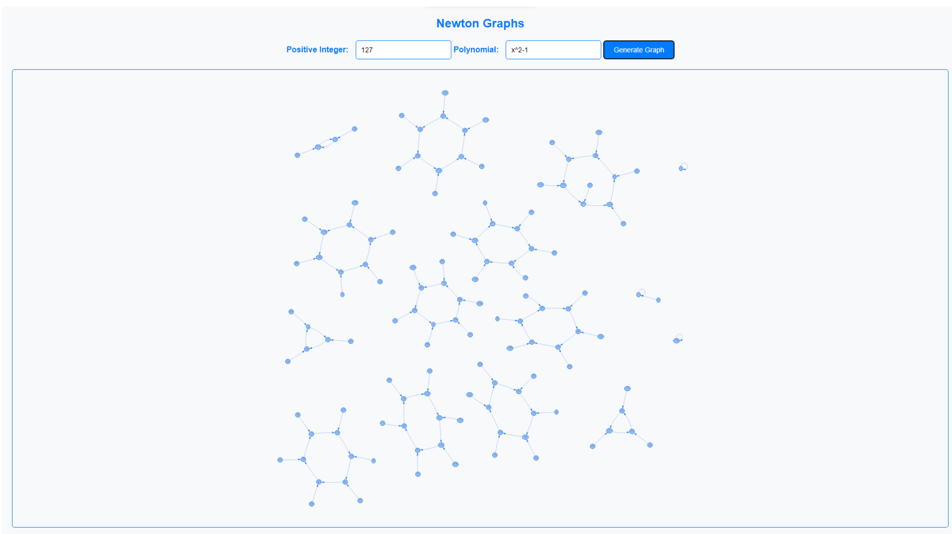


FIGURE 21. Newton graph visualizer (example is $x^2 - 1$ on the integers modulo 127)

If you're interested in the Newton graphs of more advanced structures and are open to taking a look into some advanced maths (that I barely understand myself), we also have a visualizer for a specific type of field extension that outputs the Newton graph of $\mathbb{F}_{p^n}$ for a specified prime p and degree n for a specified polynomial at this link Newton graph field extension visualizer. These field extensions are enabled by something called Conway polynomials, and while the maths involved with this are not necessarily for the faint of heart, it can still be fun to see the structures created in these graphs (try incrementing the degree a few times!).

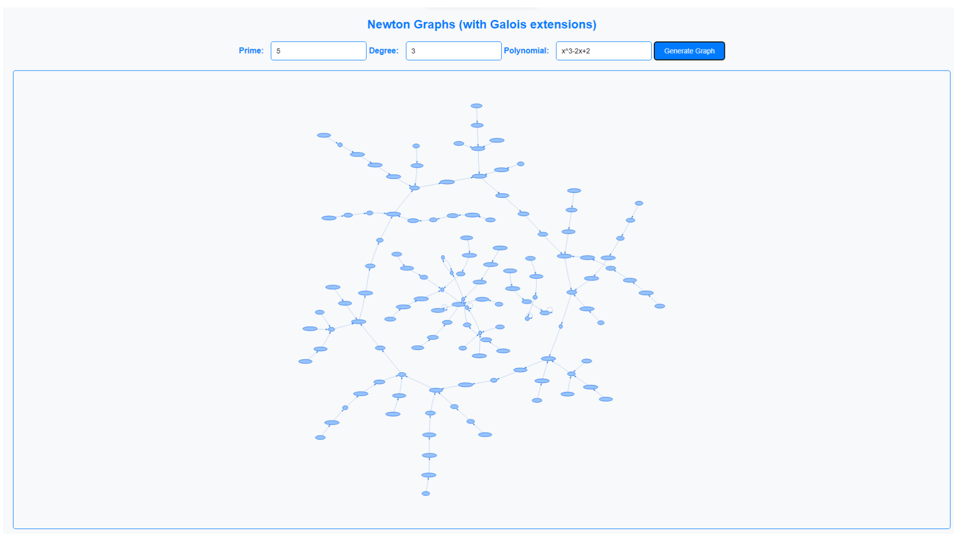


FIGURE 22. Newton graph with field extensions visualizer (example is $x^3 - 2x + 2$ on $\mathbb{F}_{5^3}$)

References

[1] U.M. Ascher and C. Greif. *Nonlinear Equations in One Variable*, pages 39–64. Computational Science and Engineering. Society for Industrial and Applied Mathematics, 2011.

[2] Kendall E. Atkinson. *Newton's Method*, page 58–65. John Wiley amp; Sons, Inc., 1989.

[3] Kenneth Falconer. Fractal geometry: Mathematical foundations and applications. 2003.

[4] J. W. Milnor. Attractor. *Scholarpedia*, 1(11):1815, 2006. revision #186525.

[5] H. O. Peitgen, D. Saupe, and F. v. Haeseler. Cayley's problem and julia sets. *The Mathematical Intelligencer*, 6(2):11–20, 1984.

[6] Simon Tatham. Fractals derived from newton-raphson iteration, 2017.