Basic complex dynamics

A computational approach

Basic complex dynamics

A computational approach

Mark McClure
University of North Carolina at Asheville

March 4, 2019

Preface

This is an introductory text on the basics of complex dynamics in one dimension. The intended audience includes undergraduate students of mathematics or other technical disciplines with a strong background in calculus and some exposure (or willingness to explore) more advanced topics.

Obviously, some knowledge of the basics of complex variables is necessary to study complex dynamics but one does *not* need to fully digest the complete body of knowledge of complex variables in order to follow the topics in this text. One just needs a basic understanding of the algebra and geometry of the complex plane, together with an understanding of how polynomial functions work. I recommend studying chapters 1 and 2 of A First Course in Complex Analysis by Mathias Beck et. al. When moving beyond the dynamics of polynomials, some information from chapter 3 would also be useful - in particular, section 3.1 on Möbius transformations.

Topics in the text include

- The iteration of functions $f : \mathbb{C} \to \mathbb{C}$
- Julia sets of quadratic functions, higher order polynomials, and rational functions
- The Mandelbrot set and other bifurcation loci
- Motivation for iteration, such as Newton's method

Real iteration is also discussed to ease the entrance into the study of dynamical systems. However, the topics in real iteration are chosen only to facilitate the study of complex iteration. Important topics in real iteration (such as the Schwarzian derivative, the period three theorem, and the Sharkovski ordering) are not discussed at all. Symbolic dynamics makes only a hidden appearance through the study of the doubling map, as that helps us understand the complex square function.

Another important aspect of this text is its emphasis on computation. We introduce Python code that runs live in the online version to generate images of many of the sets that we'll meet. Such code is generally quite simple using basic constructs such as function definition, conditionals, and iterative loops.

This text was written with PreTeXt which makes groovy things like live Python code and nicely formatted online and print versions easy.

Contents

Chapter 1

Introduction

1.1 Images of complex dynamics

I remember the moment I found I could become a mathematician. It was the summer between my sophomore and junior years of college and I happened upon the August, 1985 edition of Scientific American. The cover featured one of the most captivating images I've ever seen with swirling colors radiating out of a convoluted central structure. I quickly turned to the article in the "Computer Recreations" column of that issue and found many more such images that, according to the article, lived in the complex plane.

To be clear, mathematics consists of much more than images. Even in that first, introductory article, I found several mathematical expressions, including the fairly benign looking $z \to z^2 + c$. For me, mathematical intuition deals largely with the relationship between images and formulae. Yet, images alone can extremely captivating and don't need a lot of formal effort to appreciate. Thus, let's start by looking a few of the images that we'll learn about.

1.1.1 The images

1.1.1.1 The Mandelbrot set

The Mandelbrot set is one of the most iconic images in all of complex dynamics. As we will learn, it arises in the study of the family of functions

$$f_c(z) = z^2 + c.$$

Note that the variables are all complex numbers and the Mandelbrot set itself lives in the complex plane.

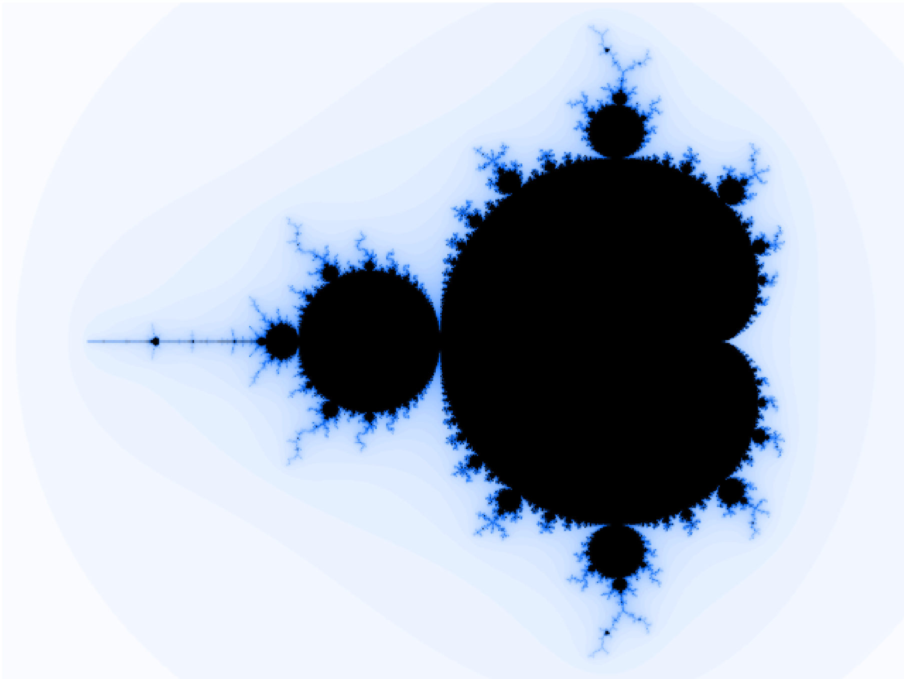

Figure 1.1.1: The Mandelbrot set

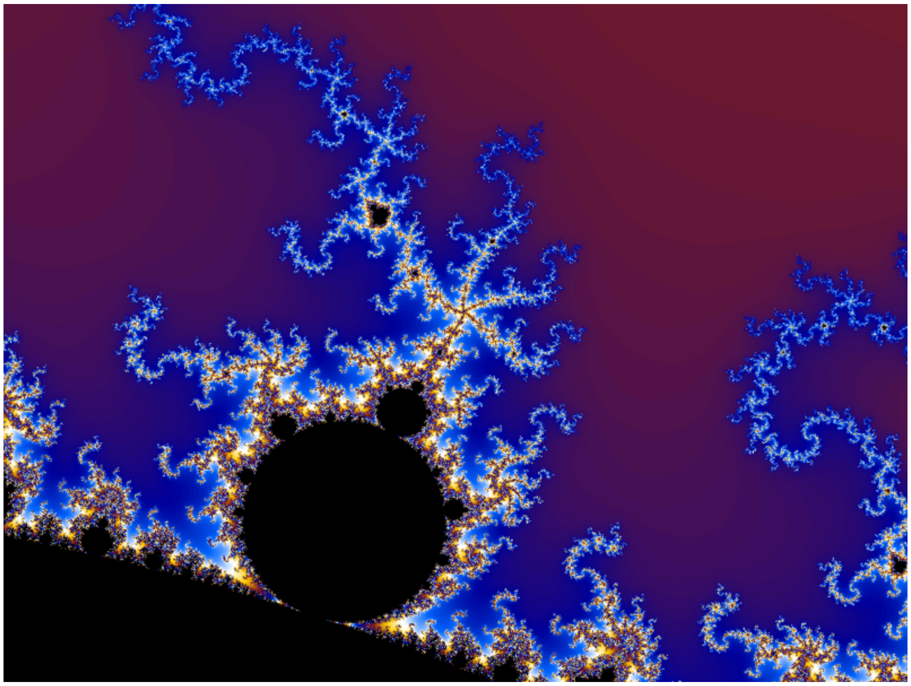

Figure 1.1.2: The region displayed on Scientific American's cover

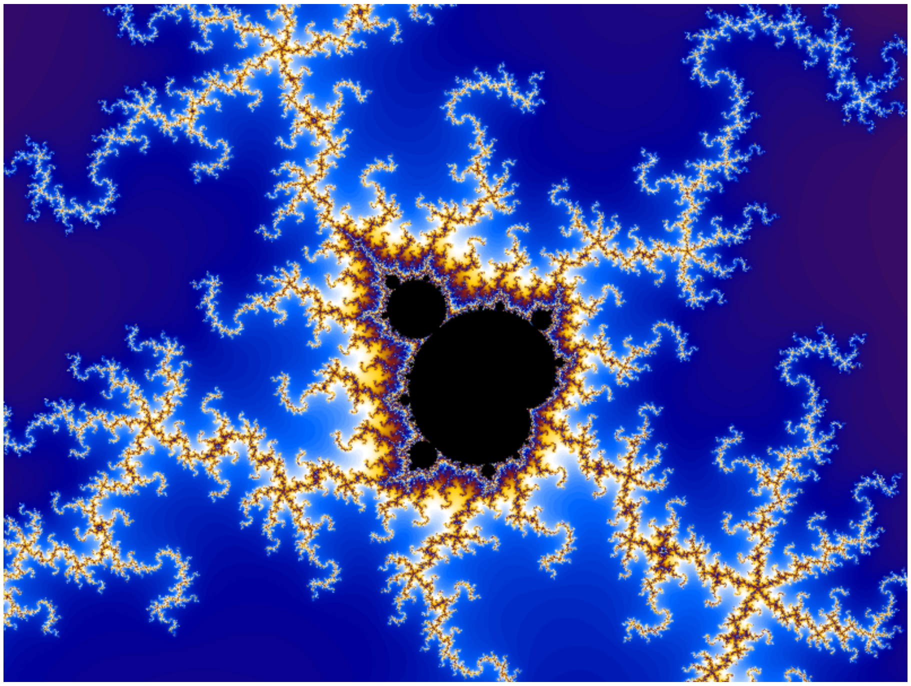

Figure 1.1.3: A zoom into the cover image

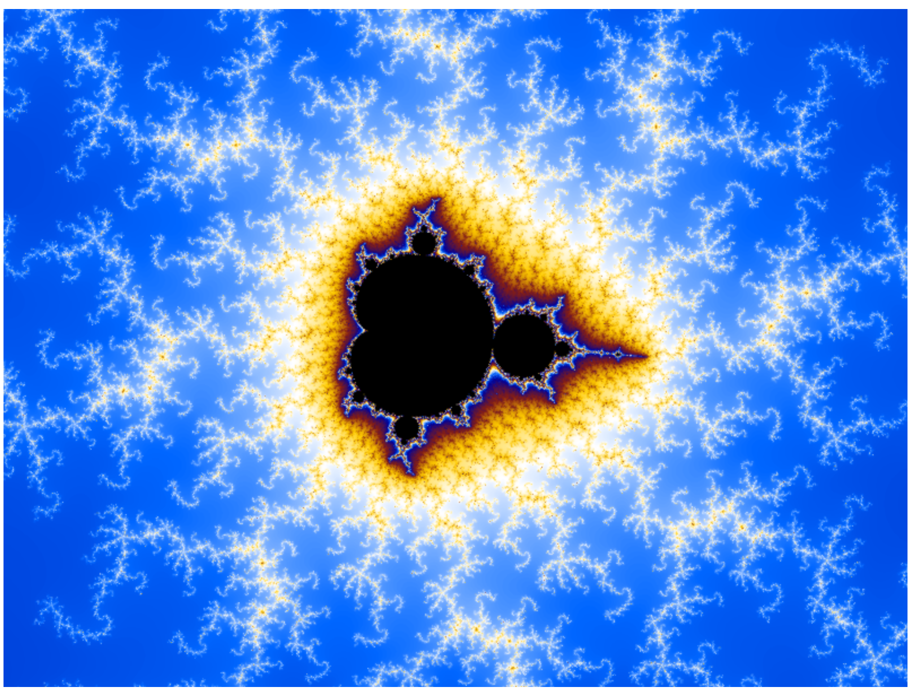

Figure 1.1.4: A much deeper zoom into the cover image

The preceding images were all generated with the following web app, where you can easily make many more such images:
https://marksmath.org/visualization/mandelzoom/.

1.1.1.2 Quadratic Julia sets

The Mandelbrot set arises in the study of the family of functions $f_c(z) = z^2 + c$. If we iterate f_c for a fixed c, we can generate a Julia set. Here are a few examples:

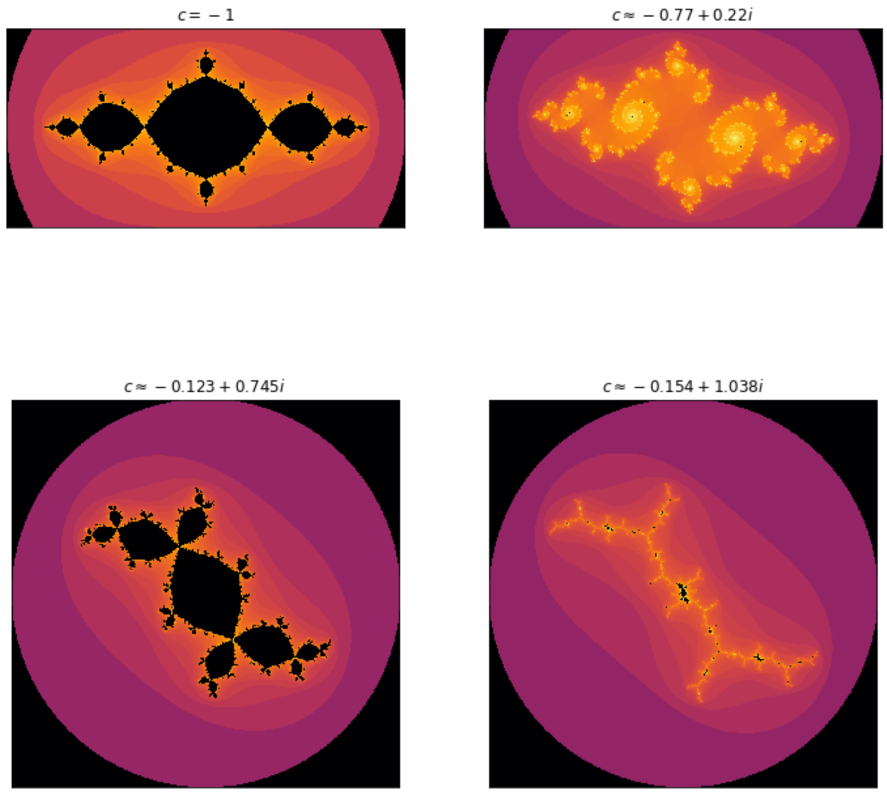

Figure 1.1.5: A few quadratic Julia sets

The preceding images were all generated with custom Python code. Here is an easy to use, point and click web app where you can explore the relationship between the Mandelbrot set and quadratic Julia sets:
https://marksmath.org/visualization/julia_sets/.

1.1.1.3 Other Julia sets

If we iterate other types of functions, more behavior is possible. Here are a couple more examples:

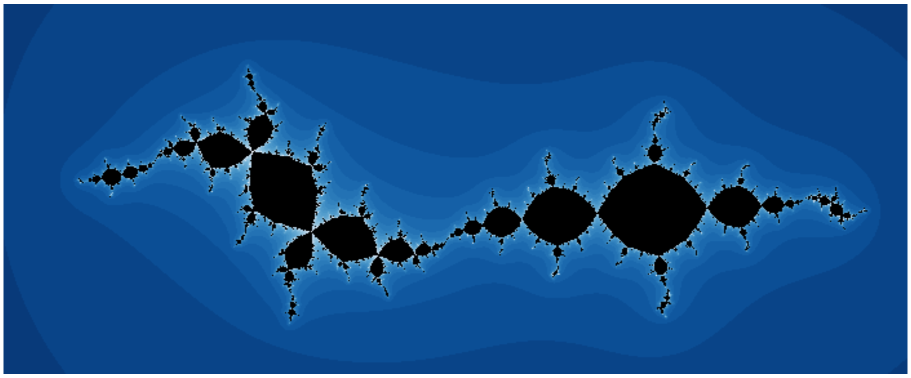

Figure 1.1.6: A cubic Julia set

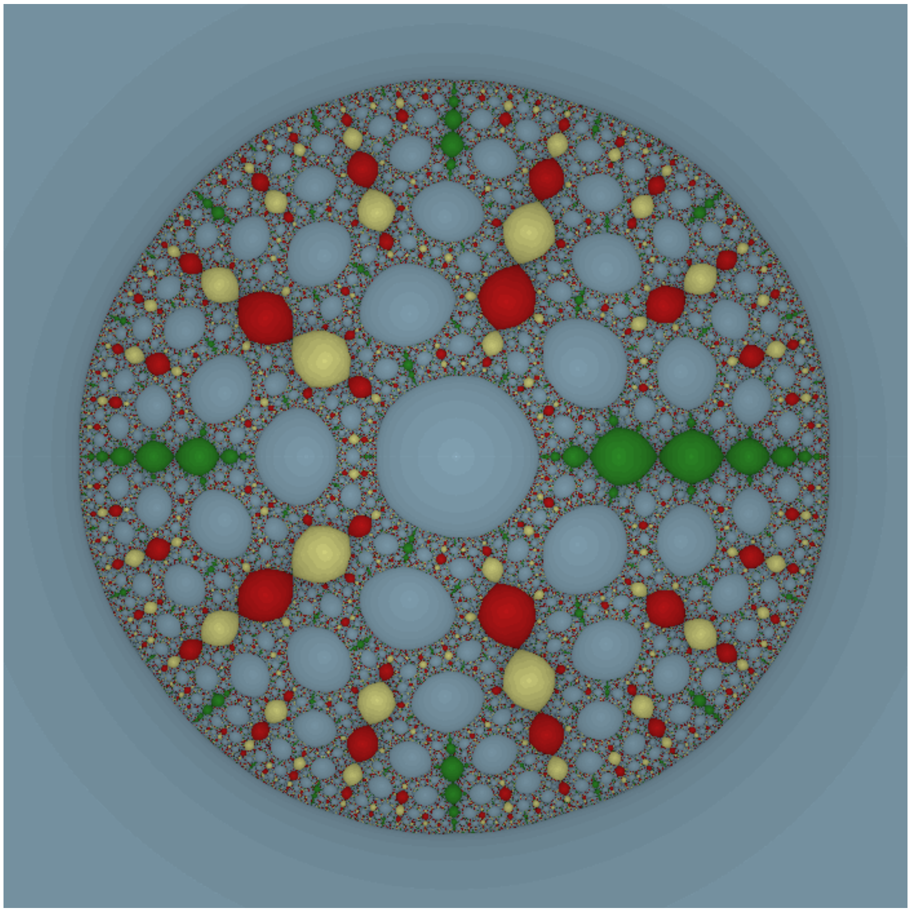

Figure 1.1.7: A rational Julia set

Figure Figure 1.1.6 shows the Julia set of the cubic polynomial

$$f(z) = (-0.090706 + 0.27145\ i) + (2.41154 - 0.133695\ i)\, z^2 - z^3.$$

It was generated with the web app here:
https://marksmath.org/visualization/polynomial_julia_sets/.

Figure Figure 1.1.7 shows the Julia set of the rational function

$$f(x) = \frac{z^5 + 0.01}{z^3}.$$

It was generated with the web app here:
https://marksmath.org/visualization/rational_julia_sets/.

1.2 Surprise in Newton's method

Newton's method is a technique to estimate roots of a differentiable function. Invented by Newton in 1669, it remains a stalwart tool in numerical analysis. When used as intended it's remarkably efficient and stable. After a little experimentation, Newton's method yields some surprises that are very nice illustrations of chaos.

1.2.1 The basics of Newton's method

Let's begin with a look at the example that Newton himself used to illustrate his method. Let $f(x) = x^3 - 2x - 5$. The graph of f shown in Figure 1.2.1 seems to indicate that f has a root just to the right of $x = 2$.

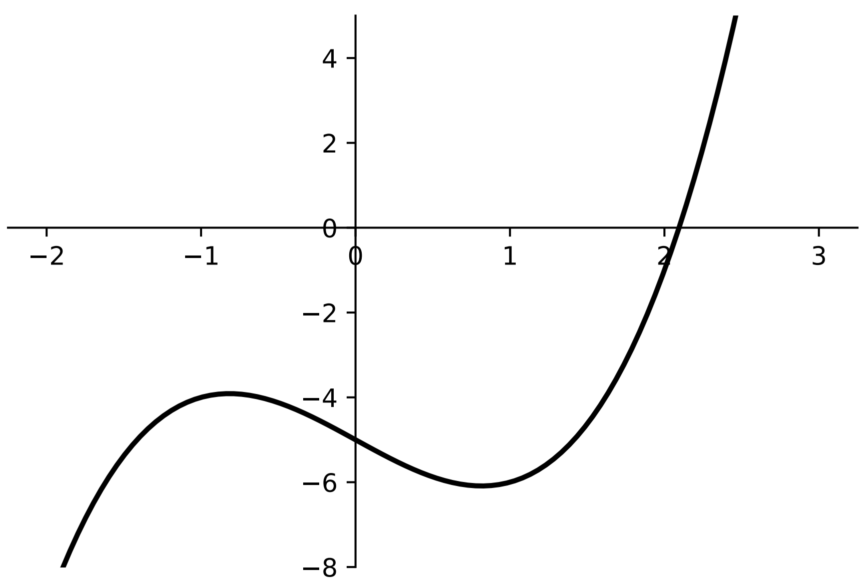

Figure 1.2.1: Newton's example function for his method

Newton figured that, if the root of f is a little bigger than 2, then he could write it as $2 + \Delta x$. He then plugged this into f to get

$$\begin{aligned} f(2+\Delta x) &= (2+\Delta x)^3 - 2(2+\Delta x) - 5 \\ &= 8 + 3 \times 2^2 \Delta x + 3 \times 2 \Delta x^2 + \Delta x^3 - 4 - 2\Delta x - 5 \\ &= -1 + 10\Delta x + 6\Delta x^2 + \Delta x^3 \\ &\approx -1 + 10\Delta x. \end{aligned}$$

In that last step, since Δx is small, he figures that higher powers of Δx are negligibly small. Thus, he figures that $-1 + 10\Delta x \approx 0$ so that $\Delta x \approx 1/10$.

The point is that, if 2 is a good guess at a root of f, then $2 + 1/10 = 2.1$ should be an even better guess. A glance at the graph seems to verify this. Of course, we could then repeat the process using 2.1 as the initial guess. We should get an *even better* estimate. The process can be repeated as many times as desired. This is the basis of *iteration*.

1.2.2 Newton's method in the real domain

Let's take a more general look at Newton's method. The problem is to estimate a root of a real, differentiable function f, i.e. a value of x such that $f(x) = 0$. Suppose also that we have an initial guess x_0 for the actual value of the root, which we denote r. Often, the value of x_0 will be based on a graph. Since f is differentiable, we can estimate r with the root of the tangent line approximation to f at x_0; let's call this point x_1. When x_0 is close to r, we often find that x_1 is even closer. This process is illustrated in Figure 1.2.2 where it indeed appears that x_1 is much closer to r than x_0.

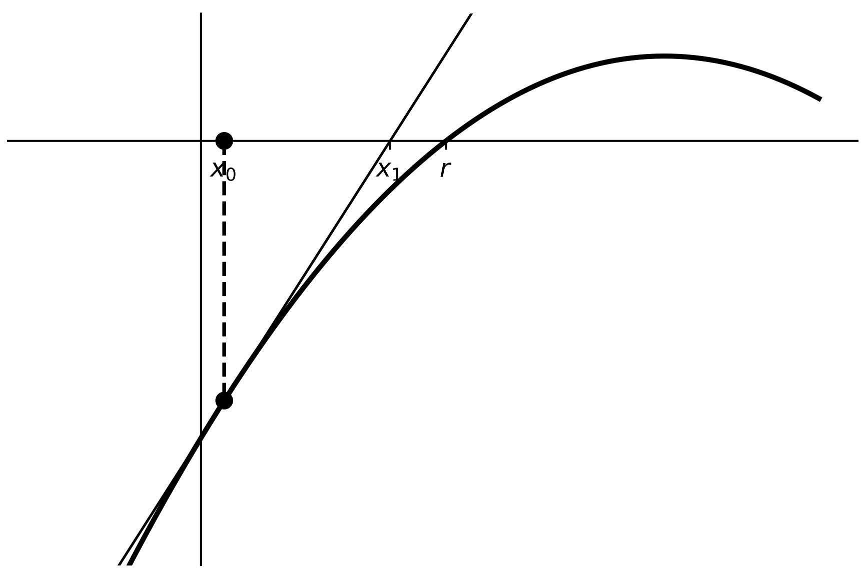

Figure 1.2.2: One step in Newton's method

We now find a formula for x_1 in terms of the given information. Recall that x_1 is the root of the tangent line approximation to f at x_0; let's call this tangent line approximation ℓ. Thus,

$$\ell(x) = f(x_0) + f'(x_0)(x - x_0)$$

and $\ell(x_1) = 0$. Thus, we must simply solve

$$f(x_0) + f'(x_0)(x - x_0) = 0$$

for x to get $x_1 = x_0 - f(x_0)/f'(x_0)$.

Now, of course, x_1 is again a point that is close r. Thus, we can repeat the process with x_1 as the guess. The new, better estimate will then be

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

The process can then repeat. Thus, we can define a sequence (x_n) *recursively* by

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

This process is called *iteration* and the sequence it generates often converges to the root r.

1.2.2.1 Examples

We now present several examples to illustrate the variety of things that can happen when we apply Newton's method. Throughout, we have a function f and we defined the corresponding Newton's method iteration function

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

We then iterate the function N from some starting point. That is, we start with an initial seed x_0 and we compute the sequence (x_n) by setting

$$x_n = N(x_{n-1}),$$

for each n.

Example 1.2.3 We start with $f(x) = x^2 - 2$. Of course, f has two roots, namely $\pm\sqrt{2}$. Thus, we might think, of the application of Newton's method to f as a tool to find good approximations to $\sqrt{2}$.

First, we compute N:

$$\begin{aligned} N(x) = x - f(x)/f'(x) &= x - \frac{x^2-2}{2x} \\ &= x - \left(\frac{x^2}{2x} - \frac{2}{2x}\right) = x - \left(\frac{x}{2} - \frac{1}{x}\right) = \frac{x}{2} + \frac{1}{x}. \end{aligned}$$

Now, suppose that $x_0 = 1$. Then,

$$\begin{aligned} x_1 &= N(1) = \frac{1}{2} + \frac{1}{1} = \frac{3}{2} \\ x_2 &= N(3/2) = \frac{3/2}{2} + \frac{1}{3/2} = \frac{17}{12} \\ x_3 &= N(17/12) = \frac{17/12}{2} + \frac{1}{17/12} = \frac{577}{408} \end{aligned}$$

Note that

$$(577/408)^2 = 332929/166464 = 2 + 1/166464$$

so that third iterate is quite close to $\sqrt{2}$.

Note that we've obtained a *rational* approximation to $\sqrt{2}$. At the same time, it's clear that it would be nice to perform these computations on a computer. In that context, we might generate a *decimal* approximation to $\sqrt{2}$. Here's how this process might go in Python:

```python
def N(x): return x/2 + 1/x
x = 1.0
for i in range(6):
  x = N(x)
  print(x)
```

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

Note how quickly the process has converged to 12 digits of precision.

Of course, f has two roots. How can we choose x_0 so that the process converges to $-\sqrt{2}$? You'll explore this question computationally in Exercise 1.3.2. It's worth noting, though, that a little geometric understanding can go a long way. Figure 1.2.4, for example, shows us that if we start with a number x_0 between zero and $\sqrt{2}$, then x_1 will be larger than $\sqrt{2}$. The same picture shows us that any number larger than $\sqrt{2}$ leads to a sequence that converges to $\sqrt{2}$.

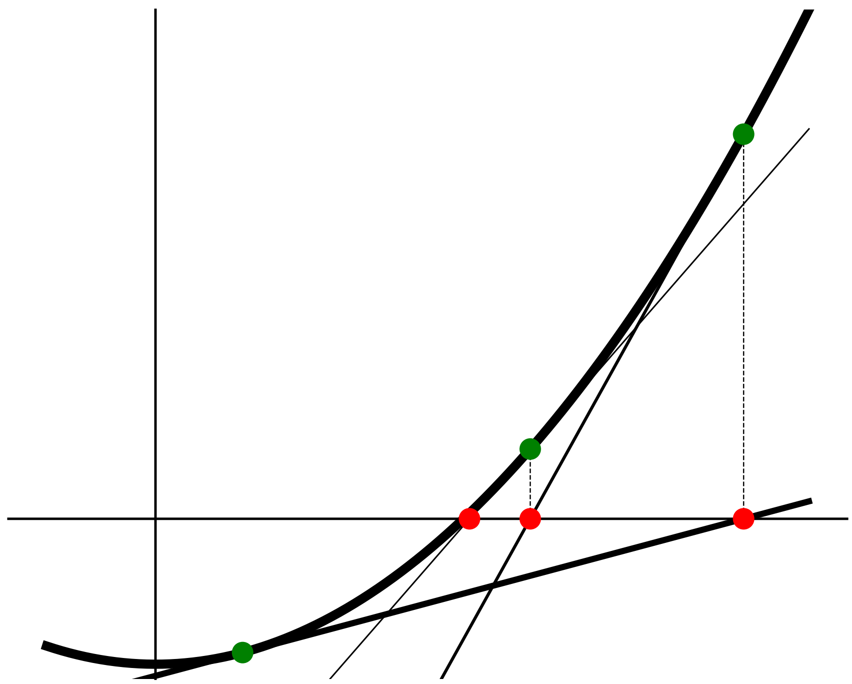

Figure 1.2.4: Three steps in Newton's method for $f(x) = x^2 - 2$

□

Example 1.2.5 We now take a look at $f(x) = x^2 + 3$. A simple look at the graph of f shows that it doesn't even hit the x-axis; thus, f has no roots. It's not at all clear what to expect from Newton's method.

A simple computation shows that the Newton's method iteration function is

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

Note that $N(1) = -1$ and $N(-1) = 1$. In the general context of iteration that we'll consider later, we'll say that the points 1 and -1 lie on an orbit of period 2 under iteration of the function N.

Sequences of other points seem more complicated so we turn to the computer. Suppose that we change the initial seed $x_0 = 1$ just a little tiny bit and iterate with Python.

```
def f(x): return x**2 + 3
def fp(x): return 2*x
def N(x): return x-f(x)/fp(x)
xi = 0.99
for i in range(100):
    xi = N(xi)
    print(xi)
```

```
Crazy list of random-looking numbers
```

Well, there appears to be no particular pattern in the numbers. In fact, if we generate 1000 iterates an plot those that lie within 10 units of the origin on a number line, we get Figure 1.2.6. This is our first illustration of chaotic behavior. Not just because we see points spread all throughout the interval but also because we appeared to have a recurring orbit when $x_0 = 1$. Why should the behavior be so different when we change that initial seed to $x_0 = 0.99$?

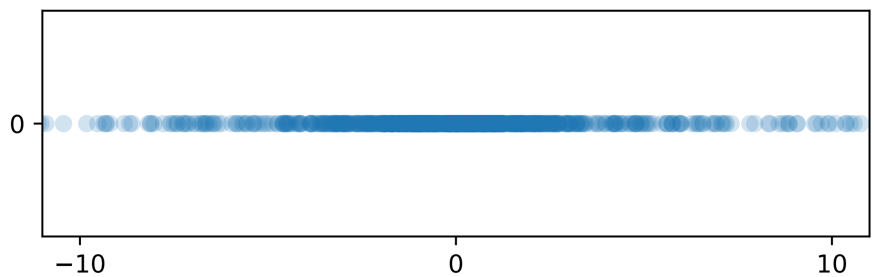

Figure 1.2.6: Chaotic behavior from Newton's method

□

Example 1.2.7 Newton's original example had one real root, Example 1.2.3 had two real roots and Example 1.2.5 had no real roots. Let's take a look at an example with *lots* real roots, namely $f(x) = \cos(x)$.

Generally, the closer your initial seed is to a root, the more likely the sequence starting from that seed is to converge to that root. What happens, though, if we start some place that's not so close to a root? What if we start close to the maximum - near zero? Let's investigate in code.

```
from numpy import sin,cos
from numpy.random import random, seed
def n(x): return x+cos(x)/sin(x)
seed(1)
for j in range(10):
    xi = random()/10
    for i in range(8):
        xi = n(xi)
    print([xi,cos(xi)])
```

```
[23.56194490192345, 8.5787174003973557e-16]
[14.137166941154069, 5.5109105961630896e-16]
[87432.094345730744, -3.2299922344716947e-12]
[32.986722862692829, -4.9047770029552963e-16]
[67.54424205218055, -4.4093474575670601e-15]
[108.38494654884786, 2.4486746176581181e-15]
[58.119464091411174, 4.8923973902235293e-16]
[29.845130209103036, 6.1294238021026498e-16]
```

```
[36.128315516282619, -3.1847006584197066e-15]
[14.137166941154069, 5.5109105961630896e-16]
```

OK, let's pick this code apart. The inner for loop looks like so:

```
xi = random()/10
for i in range(8):
  xi = n(xi)
```

Thus, xi is set to be a random number between 0 and 0.1; the for loop then iterates the Newton's method function in for the cosine from that initial seed 8 times. The outer for loop simply performs this experiment 10 times. After each run of the experiment, we print the resulting xi - along with the value of the cosine at that point, to check that we're indeed close to a root of the cosine. It's striking that we get 9 different results over 10 runs even though the starting points are so close to one another.

Figure 1.2.8 gives some clue as to what's going on. Recall that we can envision a Newton step for a function f from a point x_i by drawing the line that is tangent to the graph of f at the point $(x_i, f(x_i))$. The value of x_i is then the point of intersection of this line with the x-axis. Because the slope at the maximum is zero, the value of this point of intersection is very sensitive to small changes. In fact, there are infinitely many roots of the cosine any one of which could be hit by some initial seed in this tiny interval.

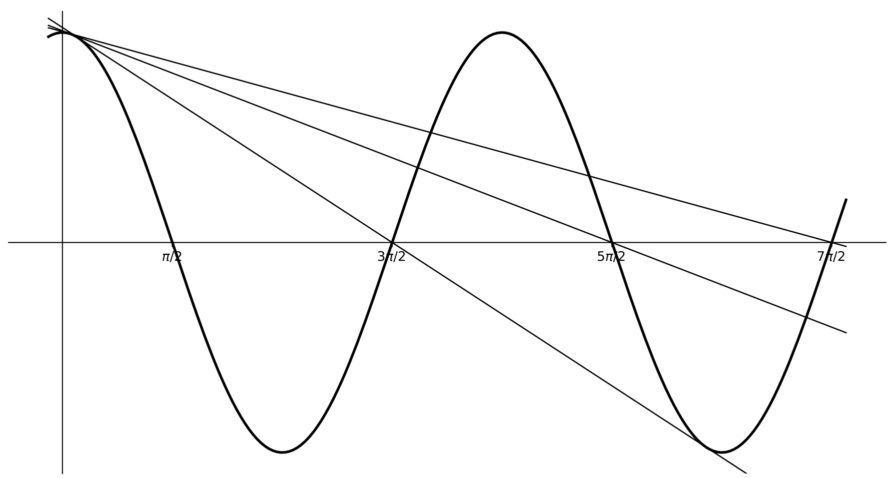

Figure 1.2.8: Initial Newton steps for the cosine

□

1.2.3 Newton's method in the complex domain

Let's move now to the complex domain, where even crazier things can happen. To understand this stuff, of course, you'll need a basic understanding of complex numbers but it's really not a daunting amount of information. You'll need to know that a complex number z has the form

$$z = x + iy,$$

where x and y are real numbers (the real and imaginary parts of z) and i is the imaginary unit (thus $i^2 = -1$). You'll also need to know (or accept) that you can do arithmetic with complex numbers and plot them in a plane, called the complex plane.

1.2.3.1 Cayley's question

In the 1870s, the prolific British mathematician Arthur Cayley (whose name is all over abstract algebra) posed the following interesting question: Suppose that p is a complex polynomial and z_0 is an initial seed that we might use for Newton's method. Generally, p will have several roots scattered throughout the complex plane. Is it possible to tell to which of those roots (if any) Newton's method will converge to when we start at z_0?

Generally, the closer an initial seed is to a root, the more likely that seed will lead to a sequence that converges to the root. It might be reasonable to guess that the process always converges to the root that is closest to the initial seed. Cayley proved that this is true for quadratic functions but commented that the general question is much more difficult for cubics. With the power of the computer, there is a very colorful way to explore the question

Algorithm 1.2.9 Coloring basins of attraction for Newton's method.

1. *Given: A function $f : \mathbb{C} \to \mathbb{C}$.*
2. *Choose a rectangle R in the complex plane with sides parallel to the real and imaginary axes.*
   - *The points in R represent potential initial seeds when applying Newton's method to f.*
3. *Discretize the rectangle into points of the form*

$$z_{0,jk} = (a + j\Delta x) + (b + k\Delta y)k.$$

4. *For each point $z_{0,jk}$ perform Newton's iteration to generate a sequence $(z_{n,jk})_n$ until one of two things happen:*
   - *$|f(z_{n,jk})| < \varepsilon$, where ε is some pre-specified small number. We assume that the process has converged to a root; color the initial seed $z_{0,jk}$ according to which root the process converged to and shade the initial seed according to how many iterates this process took.*
   - *The iteration count exceeds some pre-specified bailout; we color the initial seed $z_{0,jk}$ black.*

Let's take a look at a simple, quadratic example.

Example 1.2.10 Let $p(z) = z^2 + 3$. We already played with this function in the real case back in Example 1.2.5 and the computer indicated that there might be chaos on the real line. When we allow complex inputs, though, there are two roots - one at $\sqrt{3}i$ (above the real axis) and one at $-\sqrt{3}i$ (below the real axis). If the process always converges to the root that is closest to the initial seed, then seeds in the upper half plane should converge to $\sqrt{3}i$ while seeds in the lower half plane should converge to $-\sqrt{3}i$. We don't have the tools to prove this yet, but we can investigate with Algorithm 1.2.9. The result is shown in Figure 1.2.11.

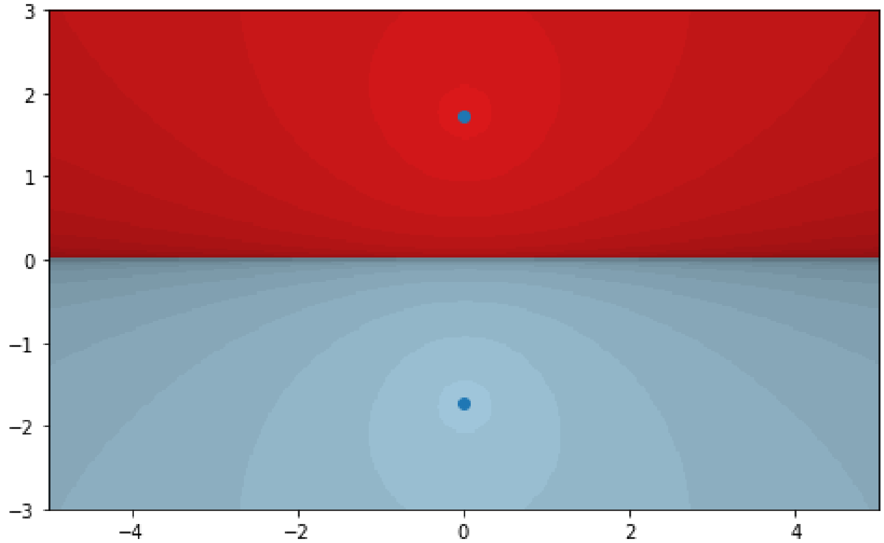

Figure 1.2.11: Attractive basins for $f(z) = z^2 + 3$

□

Now, let's take a look at a classic, cubic example.

Example 1.2.12 Let $p(z) = z^3 - 1$. As a complex, cubic polynomial, we expect p to have three roots. It's easy to see that $z = 1$ is a root. This means that $z - 1$ is a factor, which makes it easier to find that

$$p(z) = (z - 1)(z^2 + z + 1).$$

We can then apply the quadratic formula to find that the other two roots are

$$\frac{-1 \pm \sqrt{-3}}{2} = -\frac{1}{2} \pm \frac{\sqrt{3}}{2} i.$$

Note that all three roots can be clearly seen in Figure 1.2.13, which shows the result of Algorithm 1.2.9.

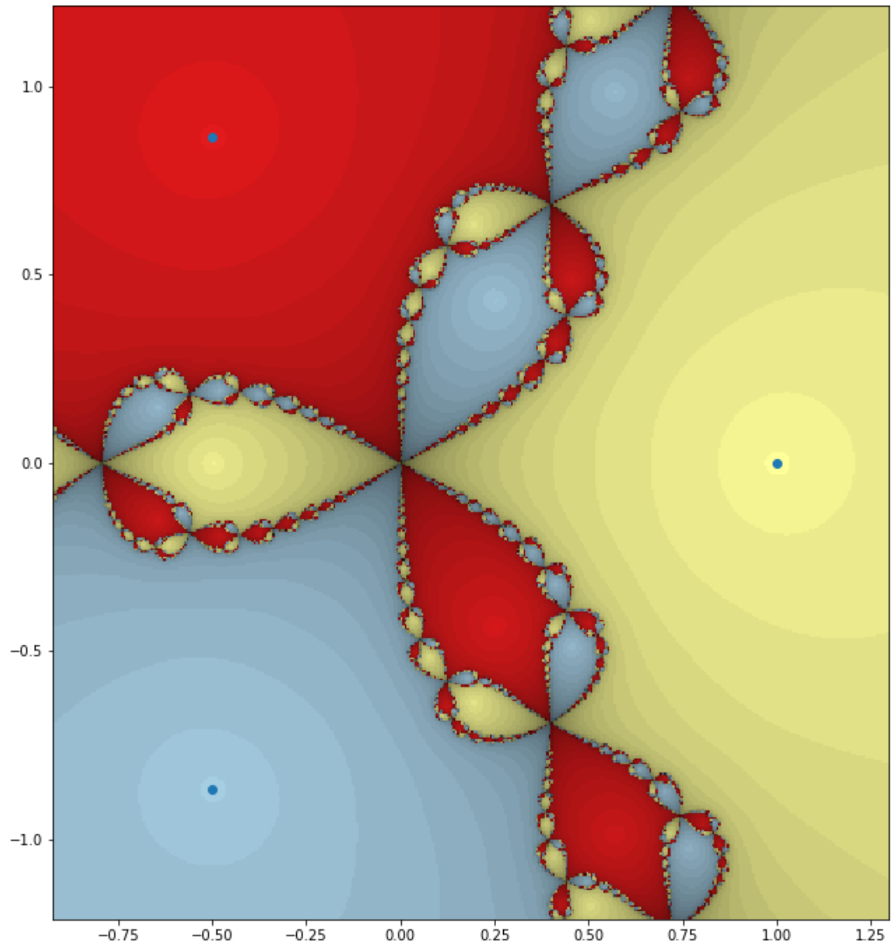

Figure 1.2.13: Attractive basins for $f(z) = z^2 + 3$

□

Figure 1.2.13 is just an incredible picture. We can clearly see the three roots of p and, indeed, if we start close enough to a root we do converge to that root. The boundary between the basins seems to be incredibly complicated, though.

As a final example, let's take a look at the complex cosine.

Example 1.2.14 Let $f(z) = \cos(z)$. Of course, we played with the real cosine back in Example 1.2.7. There, we saw that in any tiny interval containing zero, an initial seed might converge to any of infinitely many roots. In the complex case, the basins of attraction are shown in Figure 1.2.15, and a zoom is shown in Figure 1.2.16.

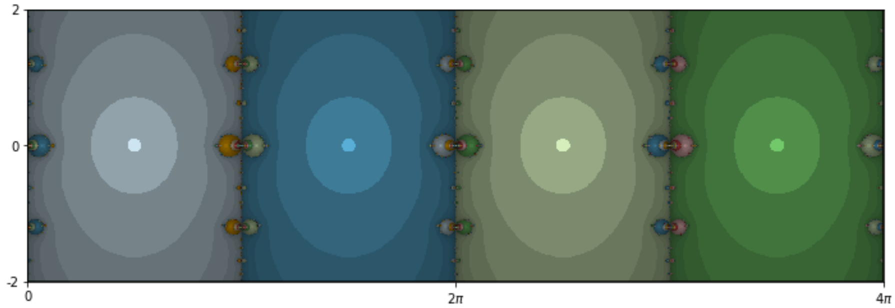

Figure 1.2.15: Attractive basins for the cosine

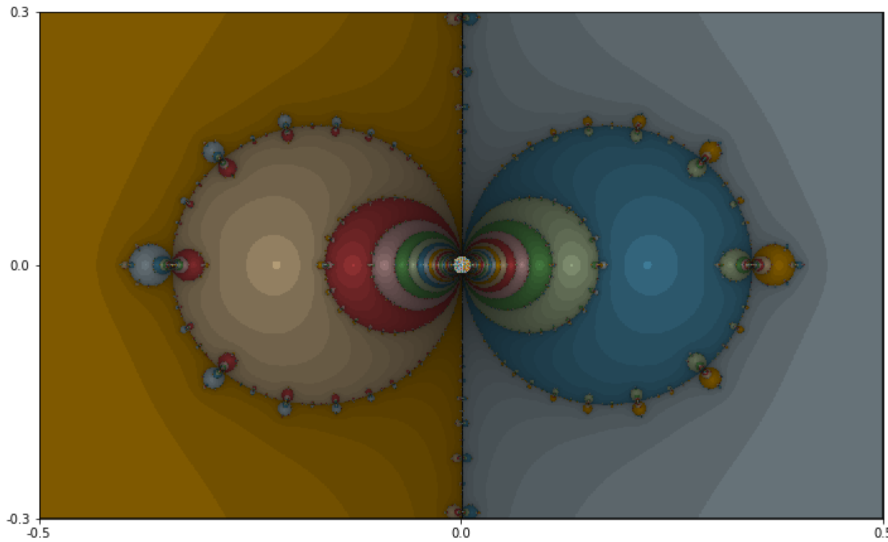

Figure 1.2.16: A zoom into the basins of the cosine

Two more incredible pictures! Note the periodicity in Figure 1.2.15. Each root seems to have its own column of attraction. The boundaries between the basins are again very complicated. In the zoom of Figure 1.2.16, t looks like there's an infinite sequence of circles collapsing on either side of the origin. This agrees with our observations in the real case of Example 1.2.7. □

Generating pictures for the Newton's method is great fun. There is an online tool that allows you to generate the basins of attraction for an (almost) arbitrary polynomial here:`https://marksmath.org/visualization/complex_newton/`.

1.3 Exercises

Beyond the first problem, this set of exercises will be mostly experimental. So, fire up your favorite computational environment. This text will include examples using both Python and Mathematica.

1. Let $f(x) = x^3 - x - 1$. Perform two Newton steps for f from $x_0 = 2$ by hand.

2. Continuing with the example of $f(x) = x^2 - 2$ explored in Example 1.2.3, compute ten Newton iterations for several values of x_0. Be sure to choose both positive and negative values and values that are both large and small in magnitude.

3. In the previous exercise, what happens when $x_0 = 0$? Draw a graph to illustrate the situation.

4. Let f be a quadractic function that has two, distinct, real roots but that is otherwise arbitrary. Using a geometrical understanding of the real Newton's method, show why an initial seed x_0 always leads to a sequence that converges to the closer of the two roots of f.

5. Let's modify Newton's original example just a little bit to consider

$$f(x) = x^3 - 2x - 2.$$

(a) Compute the corresponding Newton's method iteration function, N.

(b) Iterate N from the initial point $x_0 = 0$. What behavior do you see?

(c) Iterate N from several initial points x_0 close to zero. Now, what behavior do you see?

6. Figure 1.3.1 shows the graph of the function

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

The green dots represent points on the graph with x-coordinates that we might consider as initial seeds for Newton's method.

(a) Suppose we start at the green dot whose x coordinate is just slightly larger than 1. To which root do you think the process will converge?

(b) Suppose we start at the green dot whose x coordinate is between 2 and 3. To which root do you think the process will converge?

(c) Find a specific value of the initial seed x_0 between 2 and 3 with the property that the process converges to the smallest root of the function.

(d) Find a specific value of the initial seed x_0 between 2 and 3 with the property that the process converges to the value 1.

7. Launch the interactive tool for generating the basins of attraction of Newton's method for polynomials here: `https://marksmath.org/visualization/complex_newton/`.

Now, use the tool to generate images for the following polynomials and answer any additional questions that are asked.

(a) $f(z) = z^4 - 1$

i. What are the four roots of the function? Where do they fit into the picture?

ii. Click on the picture. How do you interpret the line that is drawn?

(b) $f(z) = (z^2 - 1)(z - 10)$

i. What initial step should you take to enter your input?

ii. What are the roots of the function? How could you account for this when generating the picture?

(c) $f(z) = z^3 - 2z - 2$

i. You should see some black regions. What's up with that?

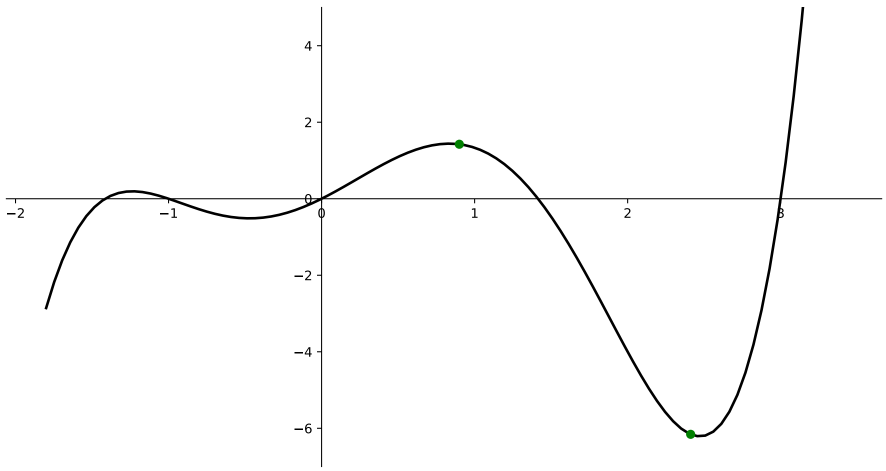

Figure 1.3.1: The graph of the function for problem 5

Chapter 2

The basics of real iteration

Introduction. Here are some introductory notes on the study of discrete, real dynamics. That is, we iterate a function $f : \mathbb{R} \to \mathbb{R}$. The objective is to introduce just the basics of discrete dynamics at the undergraduate level with a focus on topics that will be useful in the study of complex dynamics.

2.1 Basic notions

We begin with some of the most fundamental definitions and examples. While these definitions are stated for real functions, many of them extend quite easily to other contexts.

Definition 2.1.1 $x_0 \in \mathbb{R}(x_n)x_{n+1} = f(x_n)$*the orbit*$x_0 f$ ◊

Some orbits don't move; they are fixed.

Definition 2.1.2 $x_0 \in \mathbb{R}$*fixed point*$f f(x_0) = x_0$ ◊

Sometimes an orbit might return to the original starting point.

Definition 2.1.3 (x_n)

$$x_0 \to x_1 \to x_2 \cdots \to x_{n-1} \to x_0$$

$x_n = x_0$*periodic orbitperiodic points*$x_k \neq x_0 k = 1, 2, \ldots, n-1 n$*period* ◊

Note that a fixed point is a periodic point with period one.

Sometimes, the orbit of a non-periodic point might land on a periodic orbit.

Definition 2.1.4 $x_0(x_n)x_n n x_0$*pre-periodic* ◊

Example 2.1.5 Let $f(x) = x^2 - 1$. Then zero is a periodic point, since

$$0 \to -1 \to 0 \to -1 \to 0 \cdots.$$

One is a pre-periodic point, as the reader may easily verify.

To find a fixed point, we can simply set $f(x) = x$ and solve the resulting equation. In this case, we get

$$x^2 - 1 = x \text{ or } x^2 - x - 1 = 0.$$

We can then apply the quadratic formula to find that

$$x = \frac{1 \pm \sqrt{5}}{2}$$

are both fixed. □

Often, it helps to express these ideas in terms of composition of functions. We denote the n fold composition of a function with itself by f^n. That is, $f^2 = f \circ f$ and $f^n = f \circ f^{n-1}$. (Be careful note to confuse this with raising a function to a power.) A more complete understanding of periodicity arises from the study of the functions f^n. For example, a point x_0 has period n iff $f^n(x_0) = x_0$ but $f^k(x_0) \neq x_0$ for $k = 1, 2, \ldots, n-1$.

2.2 Experimentation

One of the great things about the subject of dynamics is that it is readily amenable to experimentation. If you want to know the behavior of a point under iteration of a function, simply try it out! Of course, the use of a computer comes in handy; we'll often take a look at some Python code when appropriate.

2.2.1 Examples

Example 2.2.1 Cosine iteration. Suppose we'd like to experiment with the iteration of the cosine function. We can simply do the following:

```
from numpy import cos
x = 1.0
for i in range(20):
    x = cos(x)
    print(x)
```

```
0.5403023058681398
0.8575532158463934
0.6542897904977791
0.7934803587425656
0.7013687736227565
0.7639596829006542
0.7221024250267077
0.7504177617637605
0.7314040424225098
0.7442373549005569
0.7356047404363474
0.7414250866101092
0.7375068905132428
0.7401473355678757
0.7383692041223232
0.7395672022122561
0.7387603198742113
0.7393038923969059
0.7389377567153445
0.7391843997714936
```

Note that the output appears to be converging to some fixed point, though much more slowly than we saw in the Newton's method computation. □

Checkpoint 2.2.2 Cosine iteration with various initial seeds. Iterate the cosine function from several different initial seeds. What outcomes do you see? Can you find an initial seed with a different fate from the example?

2.2.2 Resources for experimentation

There are many reasonable choices for computational tools that allow you to experiment in complex dynamics. This text will use several tools, including:

- Easy to use web apps that are implemented in Javascript
- Python code that allows you to fiddle with the code directly
- Mathematica code for solving some nasty algebraic problems

Here are a couple of easy ways to access Python:

- The Sage Cell Server is an online resource that allow you to enter blocks of code for several open source, mathematical tools - including Python. The live code in the online version of this text are all embedded Sage Cells. Be aware that the Sage Cell Server runs only Python V2 code.
- Anaconda: A full scientific Python distribution that you can download an run inside a Jupyter notebook on your machine. Anaconda supports both Python V2 and V3.

2.3 Graphical analysis

There is an efficient geometric tool to visualize functional iteration. The basic idea is simple: Suppose we graph the function f together with the line $y = x$. If those two graphs intersect; that point of intersection is a fixed point. Now, suppose we're on the line at the point (x_i, x_i). If we move vertically to the graph of the function, we preserve the x coordinate but change the y coordinate to $f(x_i)$. Thus, we arrive at the point $(x_i, f(x_i)) = (x_i, x_{i+1})$. If we then move horizontally back to the line $y = x$ we now preserve the y coordinate but change the x coordinate so that the x and y coordinates are the same. Thus, we arrive at the point (x_{i+1}, x_{i+1}).

In summary: The process of moving vertically from a point on the line $y = x$ to the graph of f and back to the line horizontally is a geometric representation of one application of the function f. This step is illustrated in figure Figure 2.3.1(a). Repeated application of this process represents repeated application of f, i.e. iteration. This is illustrated in figure Figure 2.3.1(b). Note that the orbit appears to be attractive.

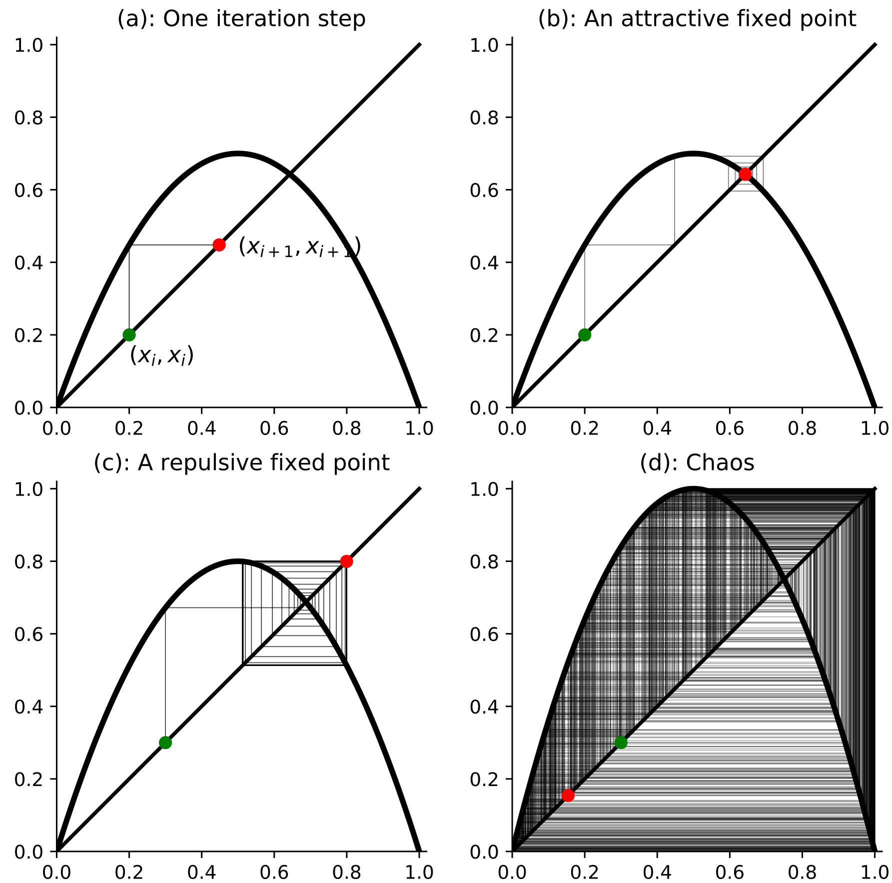

Figure 2.3.1: Some cobweb plots

It turns out that the process is quite sensitive to the slope of the function at the point of intersection. A slightly steeper function is shown in figure Figure 2.3.1(c); we notice that the fixed point now appears to be repelling. Finally, figure Figure 2.3.1(d) illustrates the fact that all hell can break loose.

There is an easy to use tool to generate cobweb plots here:
`https://marksmath.org/visualization/cobwebs/`

2.4 The classification of fixed points

The cobweb plots in the previous section illustrate that the slope of the function at the point where it crosses the fixed point might play a role in the behavior of the iterates near that fixed point. We explore that further here. First, we explore the simplest situation - functions with constant slope.

Example 2.4.1 Linear iteration. Suppose that f is a linear function: $f(x) = ax$. It's easy to see that the origin $x = 0$ is a fixed point of f. Show that any non-zero initial point x_0 moves away from the origin under the iteration of f whenever $|a| > 1$ but moves towards the origin under iteration of f if $|a| < 1$.**Solution**. This is easy, once we recognize that there is a closed form for the n^{th} iterate of f, namely $f^n(x) = a^n x$. Note that cobweb plots for these functions are shown in figure Figure 2.4.2.

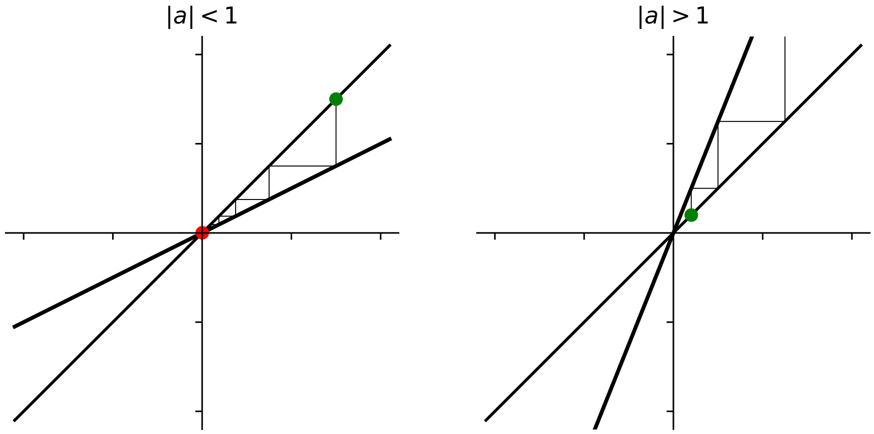

Figure 2.4.2: Some linear cobweb plots

□

Checkpoint 2.4.3 Affine iteration. Suppose, that f is an *affine function*, which just means that it has the form $f(x) = ax + b$, where $a \neq 0$. Suppose also that $x_0 \in \mathbb{R}$ and let's consider the iterates $x_{n+1} = f(x_n)$

1. Show that f has a unique fixed point iff $a \neq 1$. What if $a = 1$?
2. Suppose that $|a| < 1$. Show that the sequence of iterates converges to the fixed point of f.
3. Suppose that $|a| > 1$. Show that the sequence of iterates diverges.
4. What happens if $a = -1$?

Example 2.4.1 and Checkpoint 2.4.3 together classify the dynamical behavior of first order polynomials completely and show that their behavior is fairly simple. For that reason, we focus on polynomials of degree two and higher. Already in the quadratic case, we can find much more complicated and interesting behavior. Motivated by the behavior we see in linear and affine functions, we make the following defintion.

Definition 2.4.4 $f : \mathbb{R} \to \mathbb{R} x_0 \in \mathbb{R} f x_0$

1. *attractive*, if $0 < |f'(x_0)| < 1$,
2. *super-attractive*, if $f'(x_0) = 0$,
3. *repulsive* or *repelling*, if $|f'(x_0)| > 1$, or
4. *neutral*, if $|f'(x_0)| = 1$,

$f'(x_0)$ *multiplier* x_0 *super-attractive* ◊

The following theorem justifies this notation.

Theorem 2.4.5 $f : \mathbb{R} \to \mathbb{R} x_0 \in \mathbb{R} f$

1. *If x_0 is an attractive or super-attractive fixed point for f, then there is an $\varepsilon > 0$ such that the orbit of x under iteration of f tends to x_0 for every x such that $|x - x_0| < \varepsilon$.*
2. *If x_0 is a repelling fixed point for f, then there is an $\varepsilon > 0$ such that the orbit of x under iteration of f tends (initially) away from x_0 for every x*

such that $|x - x_0| < \varepsilon$.

Proof. We prove part one; the second part is similar. Since $|f'(x_0)| < 1$ and f' is continuous, we may choose an $\varepsilon > 0$ and a postive number $r < 1$ such that $|f'(x)| < r$ for all x such that $|x - x_0| < \varepsilon$. Then, given x such that $|x - x_0| < \varepsilon$, we can apply the Mean Value Theorem to obtain a number c such that

$$|f(x) - x_0| = |f(x) - f(x_0)| = |f'(c)||x - x_0| \leq r\varepsilon.$$

By induction, we can show that

$$|f^n(x) - x_0| \leq r^n \varepsilon.$$

The result follows, since $r^n\varepsilon \to 0$ as $n \to \infty$. ■

From the proof, we see that $x_n \to x_0$ exponentially and that the magnitude of $|f'(x_0)|$ dictates the base of that exponential. When $f'(x_0) = 0$, the rate is faster than exponential.

Example 2.4.6 The function $f(x) = 4.8\,x^2(1 - x)$ is graphed in figure Figure 2.4.7, along with the line $y = x$. The points of intersection are fixed points and, from left to right, they are super-attractive, repulsive, and attractive. The reader should consider the appearance of a cobweb plot for initial values starting near each of those fixed points.

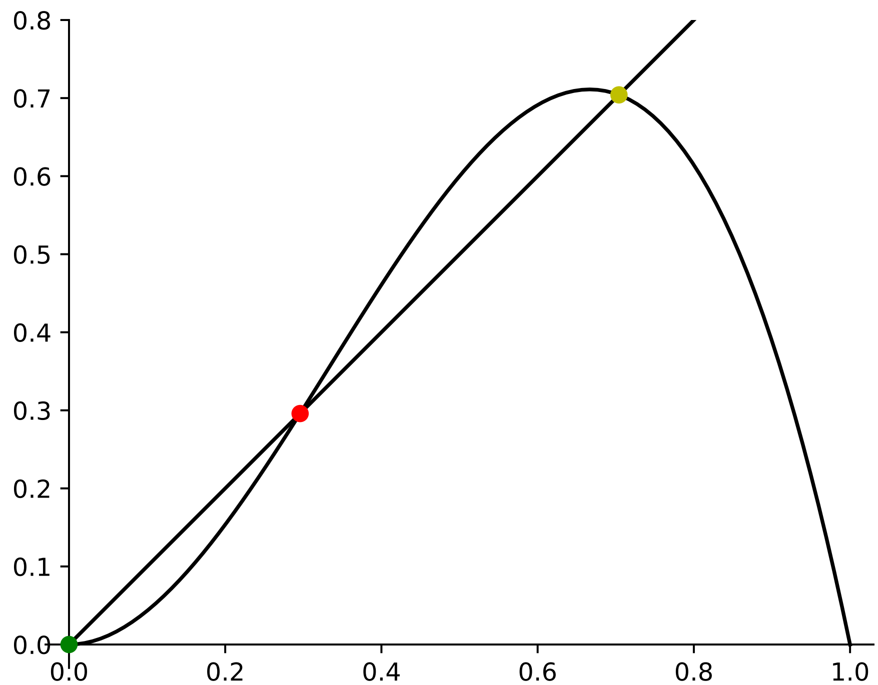

Figure 2.4.7: Three types of fixed points

□

The behavior of iterates near a neutral fixed point can be more varied.

Checkpoint 2.4.8 Dynamical behavior near neutral fixed points. For each of the following scenarios, find an example of a function $f : \mathbb{R} \to \mathbb{R}$ and a fixed point x_0 of f satisfying that scenario.

1. There is an $\varepsilon > 0$ such that the orbit of x tends to x_0 for all x such that $|x - x_0| < \varepsilon$.

2. There is an $\varepsilon > 0$ such that the orbit of x tends initially away from x_0 for all x such that $|x - x_0| < \varepsilon$.

3. There is an $\varepsilon > 0$ such that the orbit of x tends to x_0 for all x such that $0 < x - x_0 < \varepsilon$ but the orbit of x tends initially away from x_0 for all x such that $0 < x_0 - x < \varepsilon$.

4. There is an $\varepsilon > 0$ such that the orbit of x tends to x_0 for all x such that $0 < x_0 - x < \varepsilon$ but the orbit of x tends initially away from x_0 for all x such that $0 < x - x_0 < \varepsilon$.

2.5 Classification of periodic orbits

As mentioned at the end of section one a periodic point for f of period n is a fixed point of f^n. Treating the points of a periodic orbit this way allows us to extend the classification as fixed points to periodic orbits.

Definition 2.5.1 $f : \mathbb{R} \to \mathbb{R} x_0 \in \mathbb{R} f n F = f^n x_0$

1. *attractive*, if $|F'(x_0)| < 1$,

2. *super-attractive*, if $F'(x_0) = 0$,

3. *repulsive* or *repelling*, if $|F'(x_0)| > 1$, or

4. *neutral*, if $|F'(x_0)| = 1$,

$F'(x_0)$ *multiplier super-attractive* ◊

There is a nice characterization of the multiplier of an orbit that allows us to compute it without explicitly computing a formula for f^n.

Lemma 2.5.2 *Suppose that*

$$x_0 \to x_1 \to x_2 \to \cdots \to x_{n-1} \to x_0$$

is an orbit of period n for $f : \mathbb{R} \to \mathbb{R}$. Then the multiplier of the orbit is

$$f'(x_0) f'(x_1) \cdots f'(x_{n-1}).$$

Proof. First note that for an $n = 2$, we can apply the chain rule to obtain

$$\frac{d}{dx} f^2(x) = \frac{d}{dx} f(f(x)) = f'(f(x)) f'(x).$$

Thus, if $x_0 \to x_1 \to x_0$ is an orbit of period two and we evaluate that equation at x_0, we obtain

$$\left. \frac{d}{dx} f^2(x) \right|_{x = x_0} = f'(x_1) f'(x_0).$$

The result for orbits longer than two can be proven by induction, since

$$\frac{d}{dx} f^n(x) = \frac{d}{dx} f(f^{n-1}(x)) = f'(f^{n-1}(x)) \frac{d}{dx} f^{n-1}(x).$$

■

Note that the only way the product in Lemma 2.5.2 is zero, is if one of the terms is zero. This yields the following corollary.

Corollary 2.5.3 *A periodic orbit is super-attracting if and only if it contains a critical point.*

Example 2.5.4 Let $f(x) = x^2 - 1$. Note that $f(0) = -1$ and $f(-1) = 0$ so that $0 \to 1 \to 0$ forms an orbit of period 2. To see if this orbit is attractive, we examine

$$F(x) = f(f(x)) = (x^2 - 1)^2 - 1 = x^4 - x^2.$$

Note that $F'(0) = 0$ and $F'(-1) = 0$; thus, the orbit is super-attractive.

The plots of f and f^2, together with $y = x$, are shown in figure Figure 2.5.5. Note that f has two fixed points shown in red. They can found by solving the equation $x^2 - 1 = x$ and they are both repulsive under iteration of f. The two super-attractive orbits of f^2 are shown in green.

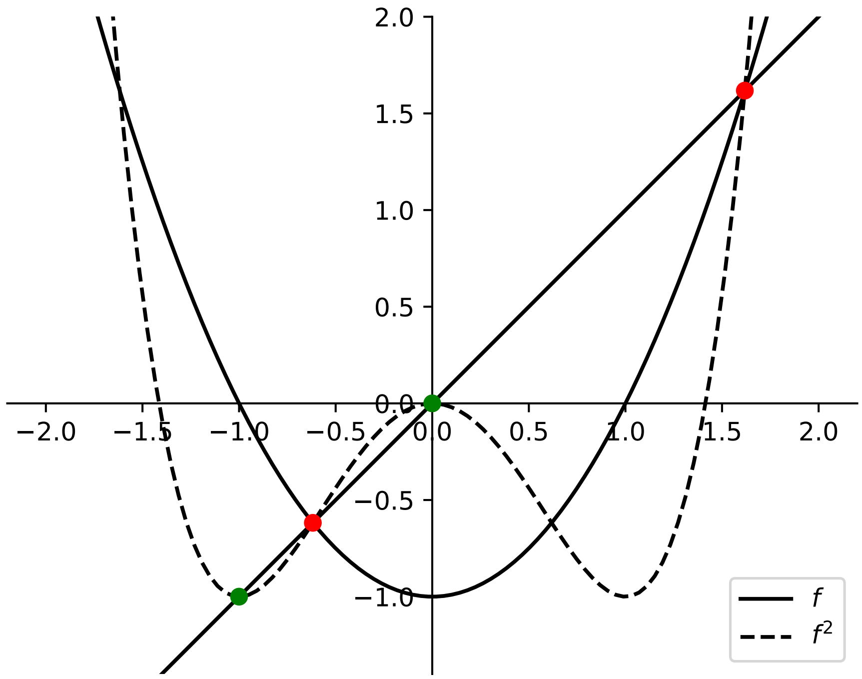

Figure 2.5.5: An attractive orbit of period two

□

2.6 Parametrized families of functions

Rather than explore the behavior of a single function at a time, we can introduce a parameter and explore the range of behavior that arises in a whole family of functions. Two basic examples that we'll spend some time with are

1. *The quadratic family*: $f_c(x) = x^2 + c$
2. *The logistic family*: $f_\lambda(x) = \lambda x(1 - x)$

The cobweb plots shown back in figure Figure 2.3.1 are all chosen from the logistic family with $\lambda = 2.8$, $\lambda = 3.2$, and $\lambda = 4$. Even in those three pictures with graphs that look so very similar, we see three different types of behavior: an attractive fixed point, an attractive orbit of period two, and chaos (which can be given a very technical meaning.

Figure Figure 2.6.1 shows some cobweb plots for the quadratic family of functions. Note that the behavior we see is very similar to the behavior we see for the logistic family - a fact that will become more understandable once we study conjugacy in Section 2.9.

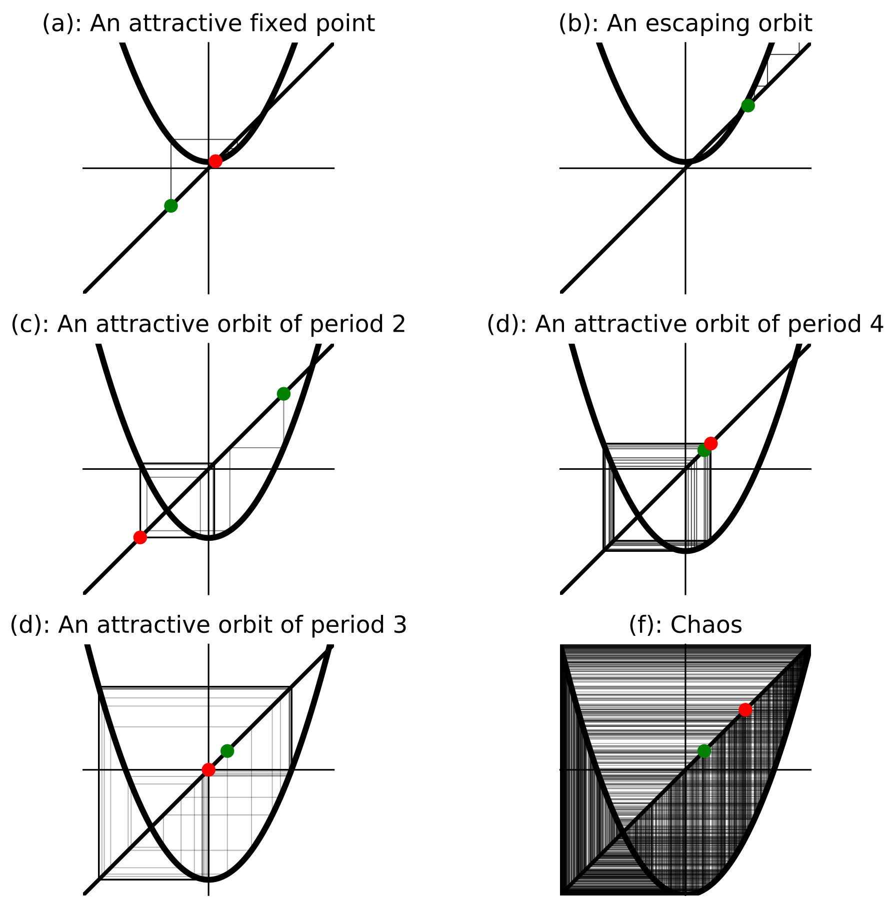

Figure 2.6.1: Some cobweb plots for the quadratic family

These pictures beg the question - how can we find values of c that generate these specific behaviors? What other types of behavior are possible? Given some desired behavior, how can we find functions that yield that behavior?

Our classification of fixed points Definition 2.5.1 is a nice algebraic tool that will help, particularly when combined with appropriate geometric tools.

2.6.1 The bifurcation diagram

A fabulous illustration of the types of behavior that can arise in a family of functions indexed by a single parameter and each with a single critical point can be generated as follows: For each value of the parameter, compute a large number points of the orbit of the critical point (maybe 1000 iterates). Since we're interested in long term behavior, rather than any transient behavior, discard the first few iterates (maybe 100). Then, plot the remaining points in a vertical column at the horizontal position indicated by the parameter.

The orbit of a critical point is called a *critical orbit* and its importance is due to the following theorem.

Theorem 2.6.2 *If $f : \mathbb{C} \to \mathbb{C}$ has an attractive or super-attractive orbit, then that orbit must attract at least one critical point.*

Note that this is really a theorem of complex dynamics. There is an analogous statement for real dynamics but it's a bit more complicated and its proof takes us a bit farther astray than we want. This is a great example of complex analysis being, in some ways, more elegant than real analysis.

Regardless, the theorem has important implications for real iteration. For example, a polynomial of degree n can have at most $n-1$ attractive orbits. Furthermore, if all the critical points happen to be real, we can find all the attractive behavior by simply iterating from the critical points. If we do this systematically for the quadratic family, plotting the columns to generate the bifurcation diagram, we get figure Figure 2.6.3

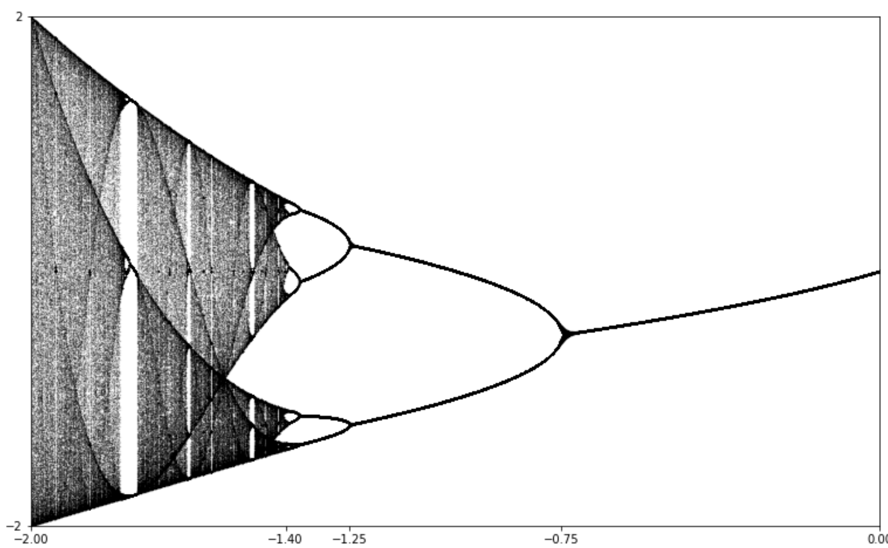

Figure 2.6.3: The bifurcation diagram for the quadratic family

We can interpret this diagram as follows:

- For $-0.75 < c < 0$, there is an attractive fixed point.
- For $-1.25 < c < -0.75$, there is an attractive orbit of period 2.
- As c passes from just above -0.75 to just below -0.75, the dynamics of f_c undergo a *bifurcation*.
- For c just a little less than -1.25, there is an attractive orbit of period four. This orbit bifurcates soon into an attractive orbit of period 8. It appears that this behavior continues as c decreases.
- For c somewhere around $c \approx -1.4$, the period doubling appears to stop and we get more complicated behavior.

This interpretation is illustrated in Figure 2.6.4

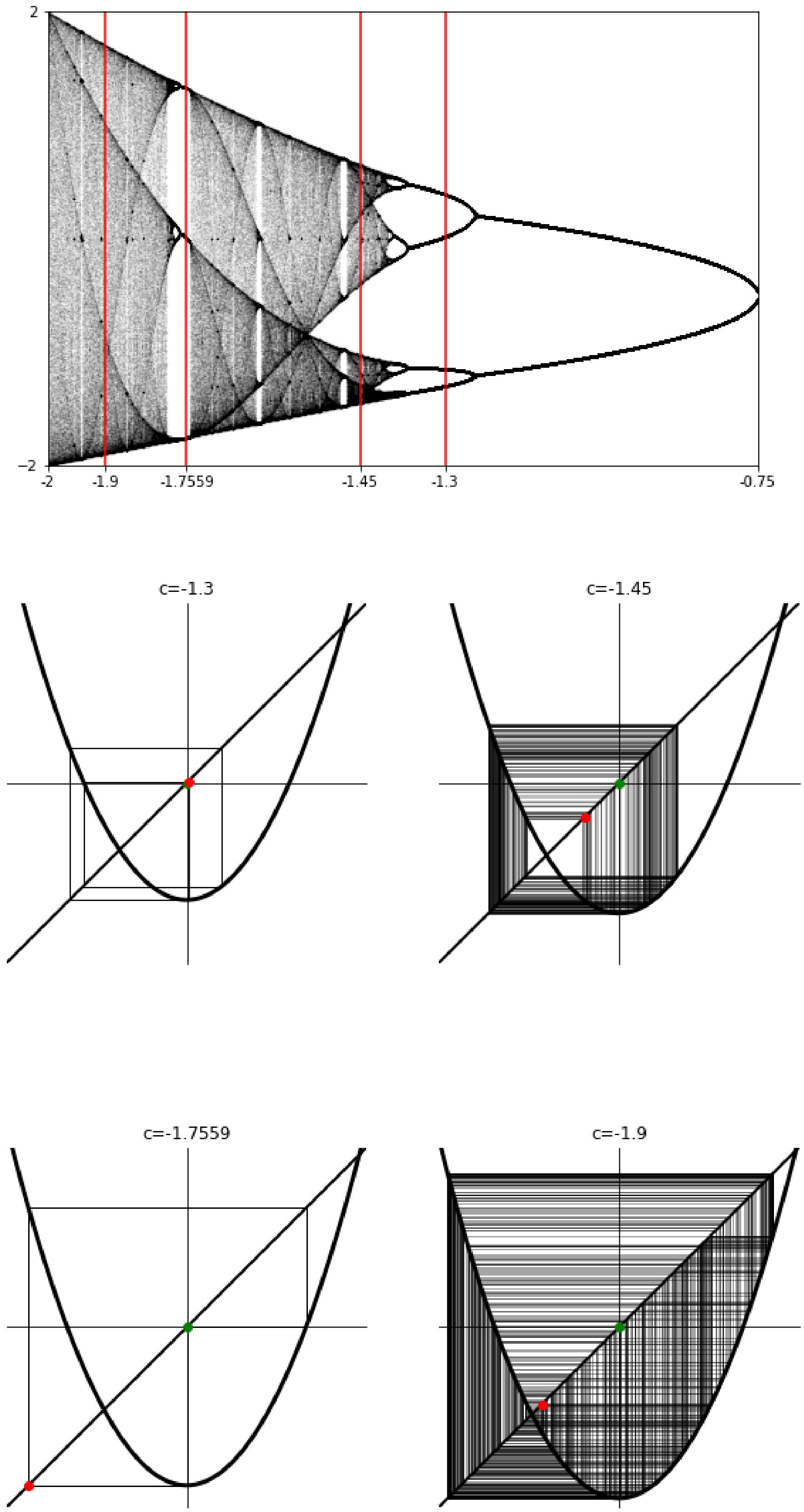

Figure 2.6.4: The correspondence between the bifurcation diagram and several cobweb plots

Generally, a bifurcation occurs at a parameter value $c = c_0$ if the global dynamical behavior of the function f_c undergoes some qualitative change as c passes through c_0. There are number of different types of bifurcations that can occur, depending on the nature of the qualitative behavior under consideration. The bifurcations that are evident in Figure 2.6.3 in the range $-1.4 < c < 0$ are called *period doubling bifurcations*.

2.6.2 The period doubling cascade

Let's work towards a deeper, theoretical understanding of the period doubling that we see in the bifurcation diagram of figure Figure 2.6.3. Again, we are dealing with the family of functions $f_c(x) = x^2 + c$. For c just a bit larger than -0.75 it appears that we have an attractive fixed point while, for c just a bit smaller than -0.75, it appears that we have an attracting orbit of period two. Why, exactly, does this happen?

First, let's explore the fixed points of f_c; we can find them by solving $f_c(x) = x$:

$$x^2 + c = x \iff x^2 - x + c = 0.$$

Applying the quadratic formula, we find

$$x = \frac{1 \pm \sqrt{1-4c}}{2}.$$

For $c < 1/4$, we have two real fixed points but a glance at the graphs from figure Figure 2.6.1 shows that it's the smaller of these two fixed points we're interested in. Of course, $f'(x) = 2x$, so the value of the derivative at the smaller fixed point is $1 - \sqrt{1-4c}$. Plugging $c = -3/4$ into this formula, we find that this is -1. For c slightly larger than $-3/4$, this is bigger than -1 and for c slightly smaller than $-3/4$, this is smaller than -1. This explains why we have an attractive fixed point for c slightly larger than $-3/4$ that is no longer attractive once c passes below $-3/4$.

Now, we ask - why does the attractive orbit of period two appear as the attractive fixed point disappears? To see this, we consider the function

$$F_c(x) = f_c \circ f_c(x) = (x^2 + c)^2 + c = x^4 + 2cx^2 + (c^2 + c).$$

We are interested in the fixed points, thus we must solve

$$x^4 + 2cx^2 + (c^2 + c) = x \text{ or } x^4 + 2cx^2 - x + (c^2 + c) = 0. \tag{2.6.1}$$

Here is an observation that helps us factor this polynomial: Any point that is fixed by f_c must also be fixed by F_c. Thus, we expect $x^2 + c - x$ to be a factor of the polynomial in (2.6.1). Using this, we find that

$$x^4 + 2cx^2 - x + (c^2 + c) = (x^2 - x + c)(x^2 + x + c + 1).$$

We can then apply the quadratic formula to get the two new fixed points of F_c, namely

$$x = \frac{-1 \pm \sqrt{1 - 4(c+1)}}{2} = \frac{-1 \pm \sqrt{-(3+4c)}}{2}.$$

These two points form an orbit of period two for f_c. Since $f_c'(x) = 2x$ we can multiply those points by two and multiply the results to get the multiplier for the orbit. The result is:

$$\left(-1 + \sqrt{-(3+4c)}\right)\left(-1 - \sqrt{-(3+4c)}\right) = 4 + 4c.$$

When $c = -3/4$, the multiplier is 1. For c a little less than $-3/4$, the multiplier is a little less than one. Hence the orbit has become attractive.

A nice way to visualize this is to plot f_c^2 together with f_c and $y = x$ on the same set of axes for a few different choices of c. This is shown in figure Figure 2.6.5 where we can see exactly how The fixed point went from attractive to repulsive while an attractive orbit of period two showed up as c passed below -0.75.

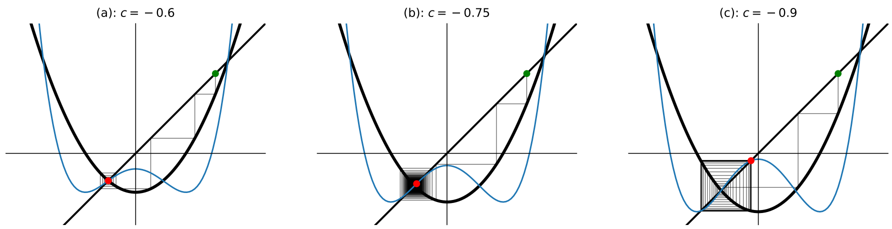

Figure 2.6.5: Bifurcation

2.7 A closer look at the bifurcation diagram

The bifurcation diagram shown in Figure 2.6.3 contains plenty of structure, in addition to the period doubling cascade, that begs for examination. Furthermore, the techniques to do so will prove useful in complex dynamics, as well.

2.7.1 The period three window

After the overall shape initiated by the period doubling cascade, the most conspicuous structure in the bifurcation diagram is probably the collection of vertical white strips. These correspond to regions where the c values lead to functions with attractive orbits. The largest such strip is called *the period three window* and a zoom into the period three window is shown in Figure 2.7.1.

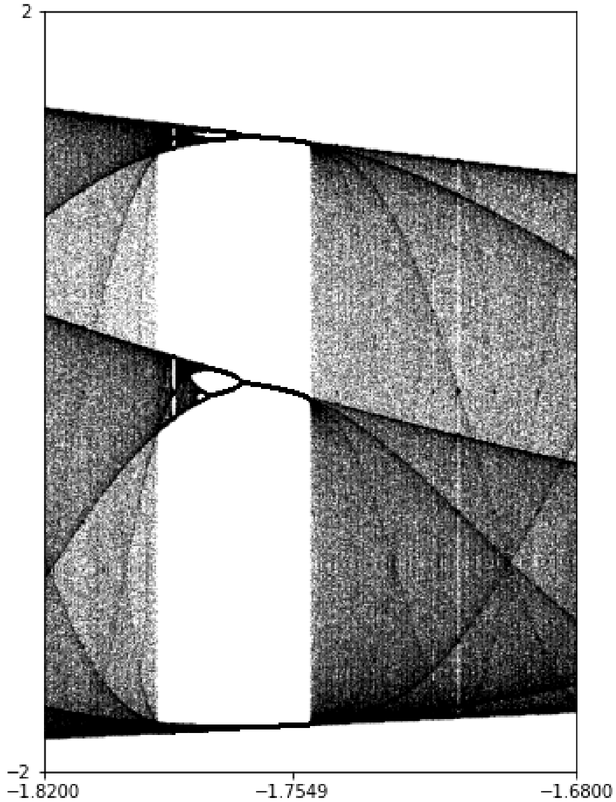

Figure 2.7.1: A zoom into the period three window

To understand the appearance of the period three window, we should plot the function $f_c(x) = x^2 + c$, together with its third iterate, as well as with the line $y = x$. It looks like we should do so for several choices of c near $c = -1.75$. This is illustrated in Figure 2.7.2.

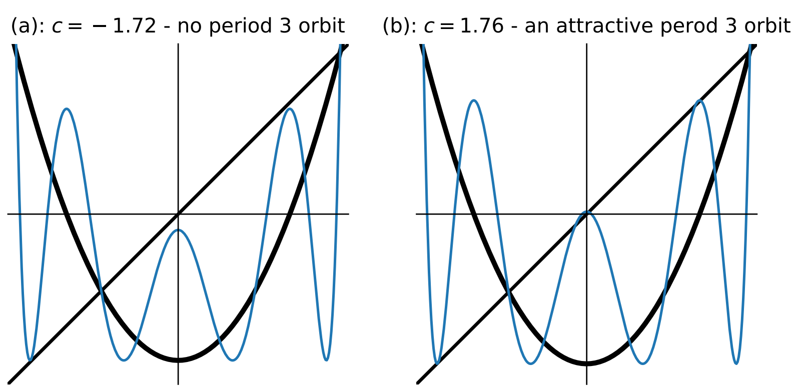

Figure 2.7.2: Occurrence of the period 3 orbit

If we would like a more precise idea as to the location of the period 3 window, we can use our characterization of periodic oribts Definition 2.5.1. As c decreases through $c \approx -1.75$ and beyond, there should be a value of c where f_c has a super-attractive orbit of period 3. By Corollary 2.5.3, this super-attractive orbit must contain the only critical point available, namely $x = 0$.

Thus, the value of c where this super-attractive orbit appears must satisfy the equation

$$f_c^3(0) = \left(c^2 + c\right)^2 + c = 0. \tag{2.7.1}$$

This value of c should lie squarely in the period 3 window and is, in fact labeled in Figure 2.7.1. We can find a good numerical estimate with the following Sage code:

```
x,c = var('x,c')
F(c,x) = x
for i in range(3):
        F(c,x) = F(c, x**2 + c)
print(F(c,x))
F(c,0).roots(ring=RR)
```

```
((x^2 + c)^2 + c)^2 + c
[(-1.75487766624669, 1), (0.00000, 1)]
```

Checkpoint 2.7.3 Find a value of c where f_c has a super-attractive orbit of period 5. Illustrate the resulting cobweb plot.

2.7.2 The critical curves

Another noticeable feature of Figure 2.6.3 is the appearance of several curves that seem to flow throughout the diagram. These are called the *critical curves* and we can characterize them precisely.

Definition 2.7.4 Critical curves. The n^{th} *critical polynomial* for the quadratic family is the polynomial in c defined by $f_c^n(0)$. The graphs of the critical polynomials are called the *critical curves*. ◊

Equation (2.7.1), for example, defines the third critical polynomial. The first six critical curves are shown on top of the bifurcation diagram in figure Figure 2.7.5. Note how spots where a critical curve crosses the x-axis correspond to a periodic window.

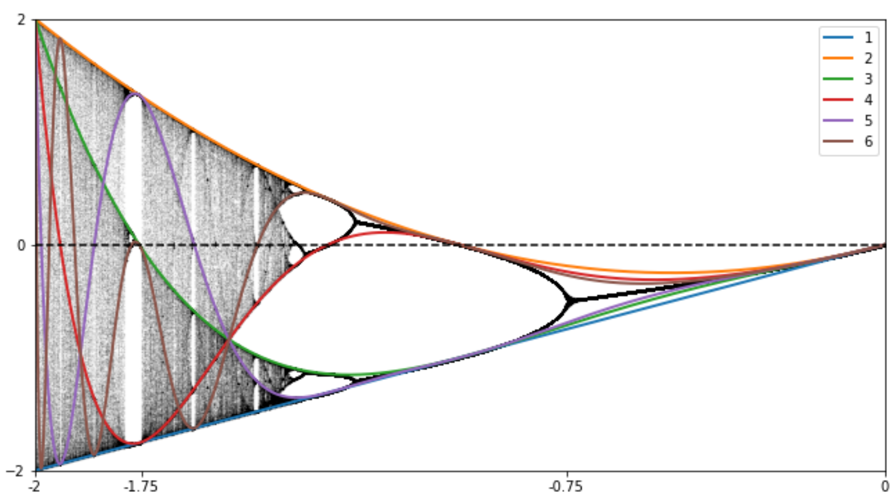

Figure 2.7.5: The first six critical curves

This explains what the curves that we see in the bifurcation diagram shown in Figure 2.6.3 *are* but not why we see them so clearly in that unadulterated version. To understand why they emerge, we examine the image under the critical polynomials of a strip centered on the x-axis. As Figure 2.7.6 shows, such a strip maps to much thinner of curved strips on one side of the critical curves. As a result, the distribution of points in the bifurcation diagram is not uniform but tend to cluster near the curves.

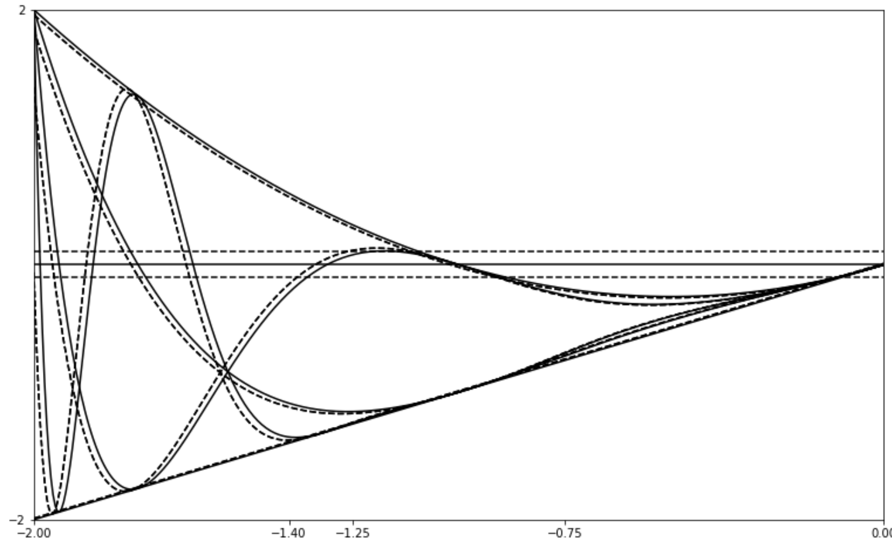

Figure 2.7.6: Mapping of a strip under the critical polynomials

2.8 The doubling map and chaos

A glance at the cobweb plots of $f(x) = x^2 - 2$ in Figure 2.6.1 and $g(x) = 4x(1-x)$ in Figure 2.3.1 shows that they both exhibit very complicated behavior. In fact, they are chaotic in a perfectly quantitative sense. In this section, we'll introduce the doubling map, which is (in a sense) the prototypical chaotic map. After studying conjugacy in Section 2.9, we'll be able to extend this result to show that f and g are chaotic as well.

2.8.1 The doubling map

Let H denote the half-open, half-closed unit interval:

$$H = [0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}.$$

The doubling map d is the function $d : H \to H$ defined by

$$d(x) = 2x \bmod 1.$$

A graph of the doubling map together with a typical cobweb plot starting at an irrational number is shown in figure Figure 2.8.1

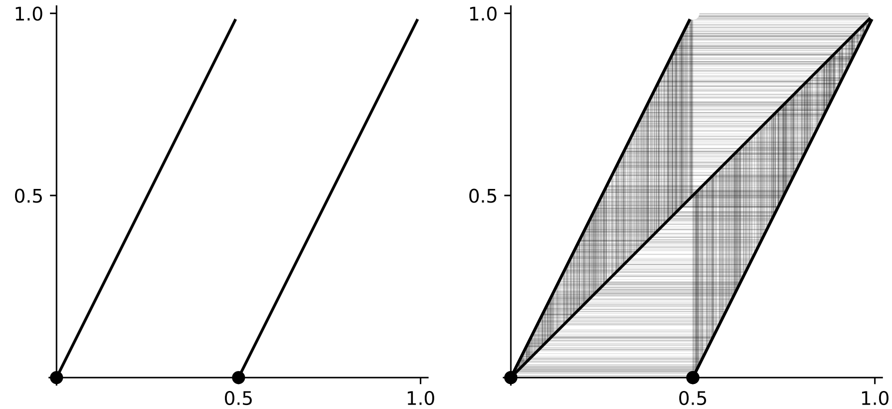

Figure 2.8.1: The doubling map

As it turns out, the doubling map is particularly easy to analyze if we consider its effect on the binary representation of a number. Suppose that $x \in H$ has binary representation

$$x = 0_{\dot{2}} b_1 b_2 b_3 b_4 b_5 \cdots ,$$

where each b_i is a zero or a one. (In computer parlance, the b_is are called the *bits* of the number.) Of course, some numbers have multiple binary representations. For example,

$$0_{\dot{2}}1 = 0_{\dot{2}}0\overline{1} = \frac{1}{2},$$

To ensure uniqueness, we agree to consistently represent numbers with a terminating binary expansion, if possible. Thus, representations that end with a repeating 1 are excluded.

Now, the effect of d on the binary representation of x is simple:

$$d(x) = 0_{\dot{2}} b_2 b_3 b_4 b_5 \cdots .$$

That is, the effect of d is to simply shift the bits of x to the left, discarding the bit that shifted into the ones place. This observation makes it very easy to find orbits with specific properties. Suppose, for example, we want an orbit of period 3. Simply pick (almost) any number of the form

$$x = 0_{\dot{2}}\overline{b_1 b_2 b_3}$$

The only caveat is that we can't have all b_is the same for that would lead to either zero (which is fixed) or one (which is not in H). As a concrete example,

$$x = 0_{\dot{2}}\overline{001} = \sum_{k=1}^{\infty} \frac{1}{8^k} = \frac{1}{7}$$

has period 3. In fact, it's easy to verify that

$$\frac{1}{7} \to \frac{2}{7} \to \frac{4}{7} \to \frac{8}{7} = \frac{1}{7} \bmod 1$$

under the doubling map.

Another nice feature of this representation is that there is a simple correspondence between the binary expansion of a number and its position in the unit interval. Every number with a binary expansion starting with a zero lies in the left half of the unit interval, while every number starting with a one lies in the right half. The first two bits of a number specify in which quarter of the interval the number lies; the first three bits specify in which eighth of the unit interval the number lies, as shown in figure Figure 2.8.2

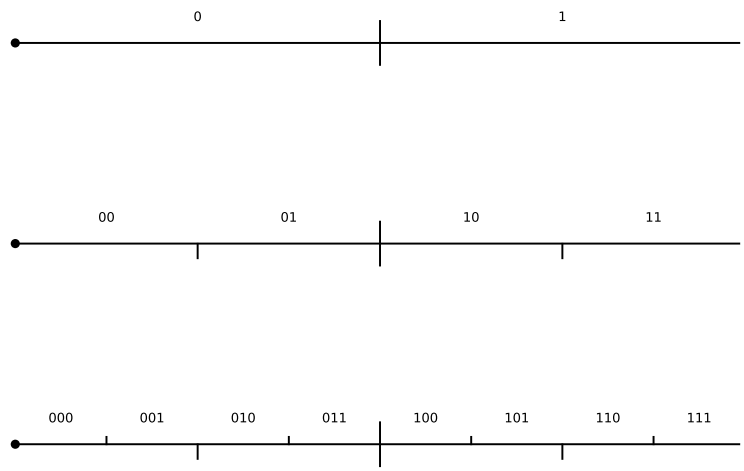

Figure 2.8.2: Dyadic intervals

More generally, given $n \in \mathbb{N}$, we can break the unit interval up into n pieces with length $1/2^n$ and endpoints $i/2^n$ for $i = 0, 1, \ldots, 2^n$. These are called *dyadic intervals* and their endpoints (numbers of the form $1/2^n$) are called dyadic rationals. The first n bits of a number specify in which n^{th} level dyadic interval that number lies. In fact, the left hand endpoint of a dyadic interval has a terminating binary expansion which tells you exactly the first n bits of all the points in that interval.

Now, suppose that

$$x_1 = 0_{\dot{2}} b_1 b_2 \cdots b_n b_{n+1} b'_{n+2} \cdots \quad \text{and} \quad x_2 = 0_{\dot{2}} b_1 b_2 \cdots b_n b'_{n+1} b'_{n+2} \cdots .$$

Thus, the binary expansions of x_1 and x_2 agree up to at least the n^{th} spot but potentially disagree after that. Then, our geometric understanding of dyadic intervals allows us to easily see that,

$$|x_1 - x_2| \leq \frac{1}{2^n}.$$

Of course, there's also a simple algebraic proof of this fact, based on the fact that the bits cancel for $k \leq n$

$$\begin{aligned} |x_1 - x_2| &= \left| \sum_{k=n+1}^{\infty} \frac{x_k - x'_k}{2^k} \right| \\ &\leq \sum_{k=n+1}^{\infty} \frac{|b_k - b'_k|}{2^k} \leq \sum_{k=n+1}^{\infty} \frac{1}{2^k} = \frac{1}{2^n}. \end{aligned}$$

2.8.2 Chaos

We can now prove three claims about the doubling map that, together, assert that the doubling map displays some of the essential features of chaos. First, we'll need to state and prove a lemma.

Lemma 2.8.3 *Suppose that*

$$x = 0_{\dot{2}} b_1 b_2 b_3 \cdots \quad and \quad y = 0_{\dot{2}} b'_1 b'_2 b'_3 \cdots$$

are elements of H *that satisfy* $b_1 \neq b'_1$ *but* $b_2 = b'_2$*. Then* $|x - y| \geq 1/4$.

Proof. Computing the difference using the binary representations, taking into account that the terms disagree in the first spot and agree in the second, and finally applying the reverse triangle inequality, we get

$$\begin{aligned} |x-y| &= \left| \sum_{i=1}^{\infty} \frac{b_i - b'_i}{2^n} \right| = \left| \pm\frac{1}{2} + \sum_{i=3}^{\infty} \frac{b_i - b'_i}{2^n} \right| \\ &\geq \left| \left| \pm\frac{1}{2} \right| - \left| \sum_{i=3}^{\infty} \frac{b_i - b'_i}{2^n} \right| \right| \geq \left| \frac{1}{2} - \frac{1}{4} \right| = \frac{1}{4}. \end{aligned}$$

■

A geometric interpretation of this lemma is as follows. The fact that the two points disagree in the first spot means that they cannot lie in the same half of H. The fact that they do agree in the second spot means that they lie in the same quarter relative to their half, as shown in figure Figure 2.8.4. Clearly, any two such points cannot be within 1/4 of one another.

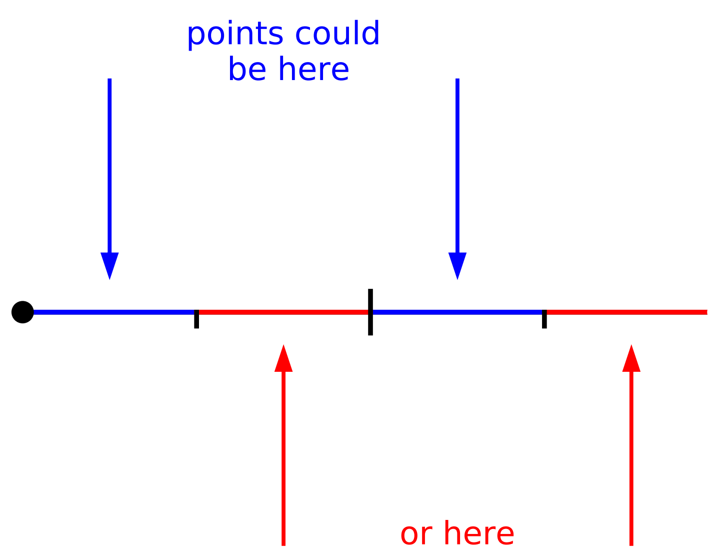

Figure 2.8.4: Possible positions of points in lemma Lemma 2.8.3

Claim 2.8.5 Sensitive dependence on initial conditions. *For every $x \in H$ and for every $\varepsilon > 0$, there is some $y \in H$ and an $n \in \mathbb{N}$ such that $|x - y| < \varepsilon$ yet $|d^n(x) - d^n(y) \geq 1/4$.*

Proof. Choose $n \in \mathbb{N}$ large enough so that $1/2^n < \varepsilon$. Now suppose that $x \in H$ has binary expansion

$$x = 0_{\dot{2}} b_1 b_2 \cdots b_n b_{n+1} b_{n+2} \cdots .$$

Define $y \in H$ so that

$$y = 0_{\dot{2}} b_1 b_2 \cdots b_n (1 - b_{n+1}) b_{n+2} \cdots .$$

That is, the bits of y agree with those of x in the first n spots, disagree with x in the $(n+1)^{\text{st}}$ spot, and finally agree with x again in the $(n+2)^{\text{nd}}$ spot.

Then, the numbers $d^n(x)$ and $d^n(y)$ satisfy the hypotheses of lemma Lemma 2.8.3, thus $|d^n(x) - d^n(y)| \geq 1/4$. ■

Claim 2.8.6 Denseness of periodic orbits. *For every open interval $I \subset H$, there is some periodic orbit with an element in I.*

Proof. Let $x \in I$ and choose $n \in \mathbb{N}$ large enough so that

$$(x, x + 1/2^n) \subset I.$$

Now suppose that

$$x = 0_{\dot{2}} b_1 b_2 \cdots b_n \cdots .$$

Then,

$$\hat{x} = 0_{\dot{2}} \overline{b_1 b_2 \cdots b_n}$$

is a periodic point in I. ■

Claim 2.8.7 A dense orbit. *There is a point $x \in H$ with the property that, for every open interval $I \subset H$, there is some iterate of x in I.*

Proof. We'll define x by specifying its binary expansion. We begin by writing down all possible *finite* binary strings:

$$0,\ 1,\ 00,\ 01,\ 10,\ 11,\ 000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111, \ldots$$

We then concatenate these to obtain the binary representation of x

$$x = 0_{\dot{2}} 0\,1\,00\,01\,10\,11\,000\,001\,010\,011\,100\,101\,110\,111 \ldots$$

Now, let $I \subset H$ be an open interval. We claim that there is some iterate of x in I. To see that, let L denote the length of I and choose $n \in \mathbb{N}$ large enough so that

$$\frac{1}{2^n} < \frac{1}{2} L.$$

Let i be the smallest integer such that $i/2^n \in I$. Note that we then also have $(i+1)/2^n \in I$. Thus, the dyadic interval $[i/2^n, (i+1)/2^n)$ is wholly contained in I and the first n bits of every point in that interval agree with $i/2^n$. So, let

$$\frac{i}{2^n} = 0_{\dot{2}} b_1 b_2 \cdots b_n$$

and note that, by construction, the string $b_1 b_2 \cdots b_n$ appears somewhere in the binary expansion of x. Thus, we can apply the doubling function to the point x some number, say m, times to obtain

$$d^m(x) = 0_{\dot{2}} b_1 b_2 \cdots b_n \cdots .$$

The number $d^m(x)$ is then an iterate of x that lies in I. ■

While there is no truly universally accepted definition of chaos, claims Claim 2.8.5, Claim 2.8.6, and Claim 2.8.7 are generally agreed to express some of the essential features of chaos. Taken together, they form what is now called *Denaney's definition of chaos*.

2.9 Conjugacy

Figure 2.3.1 and Figure 2.6.1 show that the iterative behavior of the logistic family and the quadratic family are very similar. In a sense, they are identical in a sense that we'll make precise in this section. Furthermore, we'll show that the extreme cases $f_{-2}(x) = x^2 - 2$ and $g_4(x) = 4x(1-x)$ are identical to the doubling map in this same sense and, therefore, chaotic.

2.9.1 The definition and basic examples

Definition 2.9.1 Conjugacy. Let S and T be sets and suppose that $f : S \to S$ and $g : T \to T$. We say that f is *semi-conjugate* to g if there is a surjective function $\varphi : T \to S$ such that

$$f \circ \varphi = \varphi \circ g. \tag{2.9.1}$$

The function φ is called a *semi-conjugacy*. In the case that φ is bijective, then we say that φ is a conjugacy and that f and g are conjugate. ◊

A geo-symbolic way to remember the semi-conjugation formula is in the form of a commutative diagram:

$$\begin{array}{ccc} T & \xrightarrow{g} & T \\ \downarrow \varphi & & \downarrow \varphi \\ S & \xrightarrow{f} & S \end{array}$$

The formula states if you follow the arrows from the upper left to the lower right in either direction, you get the same result.

An immediate consequence of the definition of conjugacy is

$$f^2 \circ \varphi = f \circ f \circ \varphi = f \circ \varphi \circ g = \varphi \circ g \circ g = \varphi \circ g^2$$

and, by induction

$$f^n \circ \varphi = \varphi \circ g^n.$$

As a result, if (t_i) is an orbit of g, then $(\varphi(t_i))$ is an orbit of f.

Generally, the nicer φ is, the closer the relationship between the dynamics of f and the dynamics of g. If φ is bijective, then the relationship goes both ways. In this case, equation (2.9.1) is often written as

$$f = \varphi \circ g \circ \varphi^{-1}.$$

If φ is continuous with continuous inverse, then topological properties of the orbits will be preserved. If S and T are sets of real or complex numbers and $\varphi(x) = ax + b$, then an orbit of one function will be geometrically similar to an orbit of the other. The dynamical systems are truly identical, up to a scaling.

Example 2.9.2 Show that $f(x) = x^2 - 1$ is conjugate to $g(x) = \frac{1}{2}x^2 + 2x - 2$ via the conjugacy $\varphi(x) = \frac{1}{2}x + 1$.

Solution. We simply compute

$$f(\varphi(x)) = \left(\frac{1}{2}x+1\right)^2 - 1 = \frac{1}{4}x^2 + x$$
$$\varphi(g(x)) = \frac{1}{2}\left(\frac{1}{2}x^2+2x-2\right)+1 = \frac{1}{4}x^2 + x.$$

Figure 2.9.3 illustrates the similarity between the two functions.

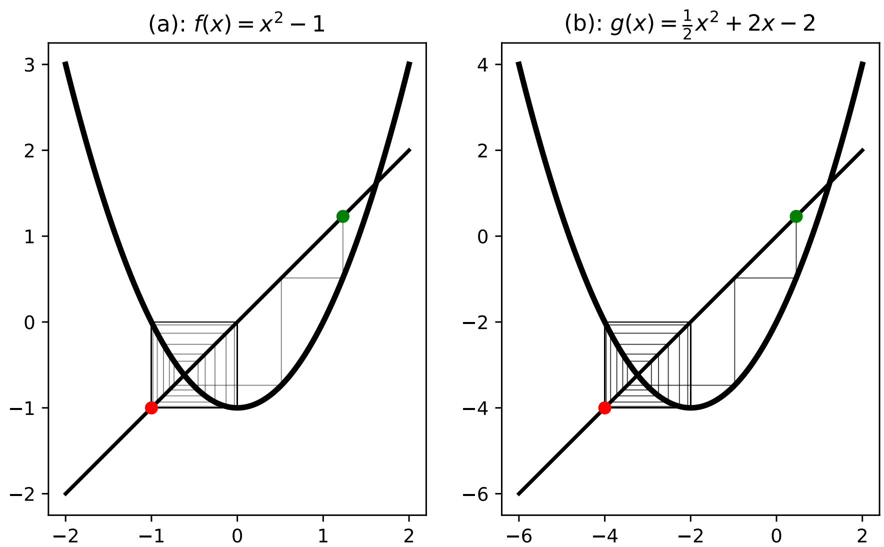

Figure 2.9.3: Cobweb plots for conjugate functions

□

If you suspect that f is conjugate to g via a conjugacy of the form $\varphi(x) = ax+b$, then you can find that conjugacy by setting $f(\varphi(x)) = \varphi(g(x))$. If you compare coefficients, you should get a system of equations that you can solve for a and b yielding the conjugacy.

Checkpoint 2.9.4 Find a conjugacy of the form $\varphi(x) = ax+b$ from $f(x) = x^2-2$ to $g(x) = 4x(1-x)$.

Exercise Checkpoint 2.9.4 can be generalized. In fact, the quadratic family for $-2 \le c \le 1/4$ is identical to the logistic family for $1 \le \lambda \le 4$.

Checkpoint 2.9.5 Show that $f(x) = x^2 + (2\lambda - \lambda^2)/4$ is conjugate to $g(x) = \lambda x(1-x)$ via the conjugacy $\varphi(x) = -\lambda x + \lambda/2$.

2.9.2 A chaotic quadratic

Let $f(x) = x^2 - 2$. We now show that f is semi-conjugate to the doubling map d under the semi-conjugacy $\varphi(x) = 2\cos(2\pi x)$. As a result, φ maps all the orbits of d has to orbits of f with similar properties. Thus, f is chaotic.

Claim 2.9.6 *The map* $f(x) = x^2 - 2$ *is semi-conjugate to the doubling map* $d(x) = 2x \bmod 1$ *under the semi-conjugacy* $\varphi(x) = 2\cos(2\pi x)$.

Proof. We must simply show that $f \circ \varphi = \varphi \circ d$, so let's compute. First,

$$f(\varphi(x)) = 2(2\cos(2\pi x))^2 - 2 = 4\cos^2(2\pi x) - 2.$$

Well, that was easy. The next part is a little trickier - we just need to apply a couple of trig identities and use the fact that we can drop the mods inside the squared trig functions due to the symmetries of those functions.

$$\begin{aligned}
\varphi(d(x)) &= 2\cos(2\pi(2x \bmod 1)) \\
&= 2(\cos^2(\pi(2x \bmod 1)) - \sin^2(\pi(2x \bmod 1))) \\
&= 2(\cos^2(2\pi x) - \sin^2(2\pi x)) \\
&= 2(\cos^2(2\pi x) - (1 - \cos^2(2\pi x))) \\
&= 2(2\cos^2(2\pi x) - 1) \\
&= 4\cos^2(2\pi x) - 2
\end{aligned}$$

■

Again, the key fact about semi-conjugacy is that φ maps orbits of d to orbits of f. Thus, since d has a dense orbit f too has a dense orbit. Here's a concrete example illustrating this idea.

Example 2.9.7 Find a point of period 11 for the chaotic quadratic $f(x) = x^2 - 2$.

Solution. First, it's easy to find an orbit of period 11 for the doubling map. One example is

$$0_{\dot{2}}\overline{00000000001} = \sum_{k=1}^{\infty} \frac{1}{2^{11k}} = \frac{1}{2047}.$$

The point behind conjugacy is that $\varphi(1/2047) = 2\cos(2\pi/2047)$ will be a point of period 11 for f. The reader is advised to check this numerically! □

2.10 Tent maps and Cantor sets

For $\lambda \in [0, 4]$, the logistic map g_λ has the nice property that $g_\lambda : [0, 1] \to [0, 1]$. That is, it maps the unit interval into itself. As a result, the dynamics are bounded and we don't really need to worry about escaping orbits. When $\lambda = 2$, the logistic map is surjective; it maps $[0, 1]$ *onto* $[0, 1]$.

In this section, we'll consider what happens when $\lambda > 4$. We'll see there is a bizarre set, called a *Cantor set*, that forms an invariant set of points that never escapes under iteration of g_λ.

First, we'll study a family of functions called the *tent maps* that are somewhat simpler. For this family of functions, the prototypical Cantor set, the ternary set, can arise as the invariant set.

2.10.1 Tent maps

The tent map $T_a : [0, 1] \to \mathbb{R}$ is defined by

$$T_a(x) = \begin{cases} ax & \text{if } 0 \le x \le 1/2 \\ a - ax & \text{if } 1/2 \le x \le 1. \end{cases}$$

Note that $T_a(1/2) = a/2$ using either definition; thus, the map is well defined and continuous at $x = 1/2$. Figure 2.10.1 shows the graph of T_a for several choices of a.

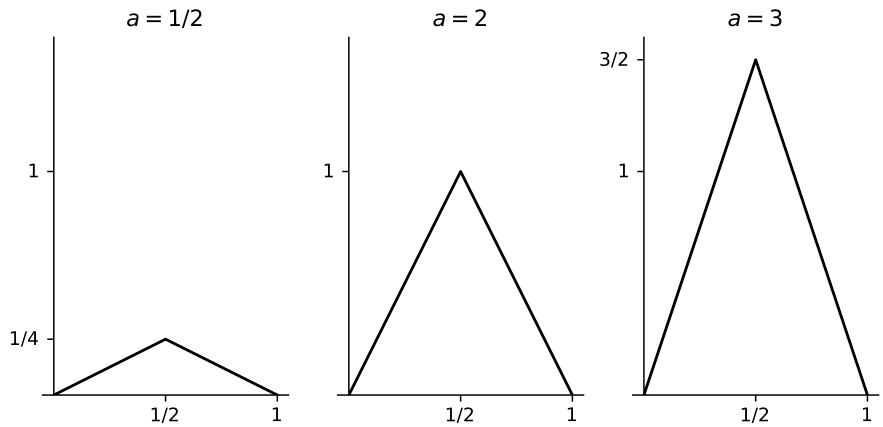

Figure 2.10.1: Three tent maps

Note that $T_a : [0,1] \to [0,1]$ for $0 \leq a \leq 2$. We say that the the unit interval is *invariant* under the action of T_a for these parameters.

2.10.1.1 $a = 2$

When $a = 2$, the function T_a has the nice property that it maps $[0,1]$ *onto* $[0,1]$. Thus, f defines a dynamical system restricted to that interval.

2.11 A few notes on computation

Many of the results in these notes have been illustrated on the computer and some of the exercises require a computational approach. Whenever using the computer, it is always wise to examine the results critically. Here's a simple numerical example where things clearly go awry. In it, we are iterating the function $f(x) = x^2 - 9.1x + 1$ from the fixed point $x_0 = 0.1$. We should generate just a constant sequence.

```
x = 0.1
for i in range(20):
    x = x**2 - 9.1*x+ 1
    print(x)
```

```
# Output:
0.09999999999999998
0.1000000000000002
0.0999999999999982
0.10000000000001608
0.09999999999985698
0.10000000000127285
0.0999999999886717
0.10000000010082188
0.0999999991026852
0.10000000798610176
0.09999992892369436
0.10000063257912528
0.09999437004618517
0.1000501066206484
0.09955405358690272
```

```
0.10396912194476915
0.06469056862056699
0.41550069522129274
-2.608415498784386
31.540412453236513
```

Uh-oh!

It must be understood that this is a simple consequence of the nature of floating point arithmetic. Part of the issue is that the decimal number 0.1 or 1/10 is not exactly representable in binary. In fact,

$$\frac{1}{10} = 0_{\dot{2}}0\overline{0011} = \frac{1}{2}\sum_{k=1}^{\infty}\frac{3}{16^k}.$$

Thus, the computer *must* introduce round-off error in the computation. Furthermore, 0.1 is a repelling fixed point of the function. Thus, that round-off error is magnified with each iteration. Our study of dynamics has illuminated a critical issue in numerical computation!

In the terminology of numerical analysis, computation near an attractive fixed point is stable while computation near a repelling fixed point is unstable. Generally speaking, stable computation is trustworthy while unstable computation is not. The implication for the pictures that we see here is that illustration of attractive behavior should be just fine. In figure Figure 2.3.1, for example, images (b) and (c) illustrate attraction to a fixed point that not only involves stable computation but also agrees with our theoretical development. We are happy with those figures. The cobweb plot shown in Figure 2.3.1 (d), however, should frankly be viewed with some suspicion.

The same can be said for the bifurcation diagram in figure Figure 2.6.3. In much of that image, we see a gray smear indicating chaos. How can we trust that? Well, first, theory tells us that there really *is* chaos. That is, there are orbits that are dense in some interval for many c values. Furthermore, much of the image shows attractive regions and we can be confident in that portion.

In fact, in many of the images that we will generate later - Julia sets, the Mandelbrot set, and similar images - the stable region dominates. Thus, we can be confident in overall image because the unstable region is the complement of the stable region. We might not be confident in computations involving some particular point, but we can be confident in the overall picture. (This will, perhaps, be more clear as we move into complex dynamics.)

Nonetheless, sometimes we want to experiment with genuinely unstable dynamics. One way to improve our confidence in these kinds of computations is to use high precision numbers. Consider, for example, the cobweb plot of the doubling map shown in figure Figure 2.8.1. A naive approach to generate the first few terms of an orbit associated with the doubling map might be as follows:

```
import numpy as np
x = 1/np.pi
for i in range(55):
    x = 2*x%1
    print(x)
```

```
# Truncated output
0.636619772368
0.273239544735
0.54647908947
...
```

```
0.375
0.75
0.5
0.0
0.0
```

We've reached the fixed point zero and now we're stuck! Even if we iterate 1000 times, we'll generate a cobweb plot that looks like figure Figure 2.11.1

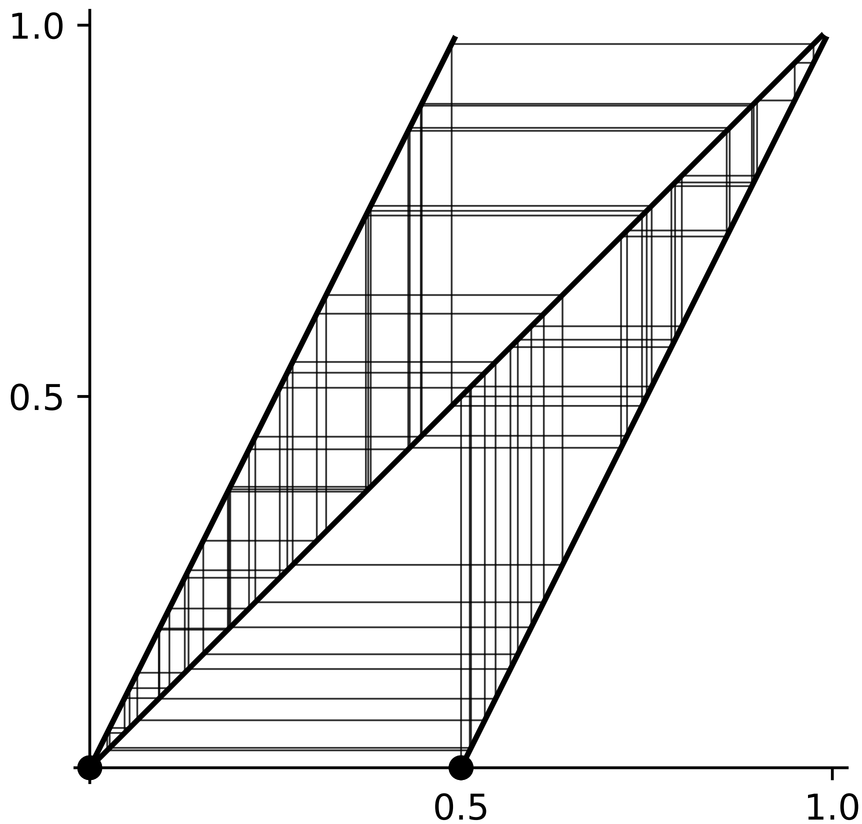

Figure 2.11.1: An inaccurate cobweb plot

The cobweb plot shown in figure Figure 2.8.1 was generated using the mpmath multi-precision library for Python with code that looked something like so:

```
from mpmath import mp
mp.prec = 1000
x = 1/mp.pi
for i in range(1000):
    x = 2*x%1
    print(float(x))
```

```
# Truncated output
0.6366197723675813
0.27323954473516265
0.5464790894703253
...
0.375
```

```
0.75
0.5
0.0
0.0
```

While the truncated output looks the same, note that this was after 1000 iterates. This behavior makes perfect sense if you understand that the doubling map loses one bit of precision with every iterate.

2.12 Exercises

1. $f : \mathbb{R} \to \mathbb{R} x_0 f f(x_0) = 0 f'(x_0) \neq 0 x_0 f x_0 N f$

2. $f : \mathbb{R} \to \mathbb{R}$

Hint. Draw a graph. Of course, you can't violate theorem Theorem 2.6.2.

3. Let $f(x) = x^2 - 4x + 5$. Show that f has a super-attractive orbit of period 2.

4. Let $f(x) = 3x^2 - 6x + 3.415$. Find all attractive orbits of f.

5. Use software to find all orbits of period 2, 3, and 4 for $f(x) = x^2 - 2$.

6. Use software to find one orbit of period 11 for $f(x) = x^2 - 2$.

7. Let $f_\lambda(x) = \lambda x(1 - x)$. Find the value of λ such that f_λ has a super-attractive orbit of period 3.

8. Find an orbit of period 11 for the doubling map.

9. Show that there is no orbit of the doubling map that is dense in some proper subinterval of H but not dense in H itself.

Chapter 3

The complex quadratic family

In this chapter, we turn to *complex* dynamics and focus our attention on the complex quadratic family:

$$f_c(z) = z^2 + c.$$

In spite of its simplicity, much of the full scope of complex dynamics and chaos arises in this family. Furthermore, this is exactly the context in which the Mandelbrot set arises.

Note that $\{f_c\}$ forms a *parametrized family of quadratics* that we discussed in some detail in the context of real iteration in section Section 2.6. Now, though, the variable z and the parameter c are complex valued.

We will be studying complex dynamics freely throughout the rest of the text so, if you're not on top of complex variables, now would be a good time to study up on it. One does *not* need to fully digest the complete body of knowledge of complex variables in order to follow the topics in this text. One just needs a basic understanding of the algebra and geometry of the complex plane, together with an understanding of how polynomial functions work. I recommend studying chapters 1 and 2 of A First Course in Complex Analysis by Mathias Beck et. al. When moving beyond the dynamics of polynomials, some information from chapter 3 would also be useful - particular, section 3.1 on Möbius transformations.

3.1 An illustrative example

Let's begin by setting $c = 0$ so we can study what must be the simplest quadratic, namely $f(z) = z^2$. In fact, it's easy to obtain a closed form expression for the n^{th} iterate of f, namely: $f^n(z) = z^{2^n}$. In spite of this formula, iteration of f displays tremendously interesting dynamics indicative of much of what is to come. As we'll see, in fact, the dynamics of f naturally decomposes the complex plane into two regions: a stable region and an unstable region. A similar type of decomposition is possible for *all* polynomials and, even, for all rational functions. We'll come to call the stable region the *Fatou set* and the unstable region the *Julia set*.

We'll try to paint a static picture of the dynamics here but there is a dynamic and interactive illustration of the actual orbits of f here: `https://www.marksmath.org/visualization/complex_square_iteration/`.

Let's suppose that $|z_0| < 1$, i.e. z_0 is in the interior of the unit circle in the complex plane. Then, since $f^n(z_0) = z_0^{2^n}$, it's easy to see that the iterates of z_0 converge to zero. In addition, the argument of the iterate doubles with each application of f. As a result, there is a spiral effect as well.

If $|z_0| > 1$, i.e. z_0 is in the exterior of the unit circle in the complex plane, then the iterates spiral out away from the circle and towards infinity. The dynamics of f is shown for several choices of z_0 in figure Figure 3.1.1.

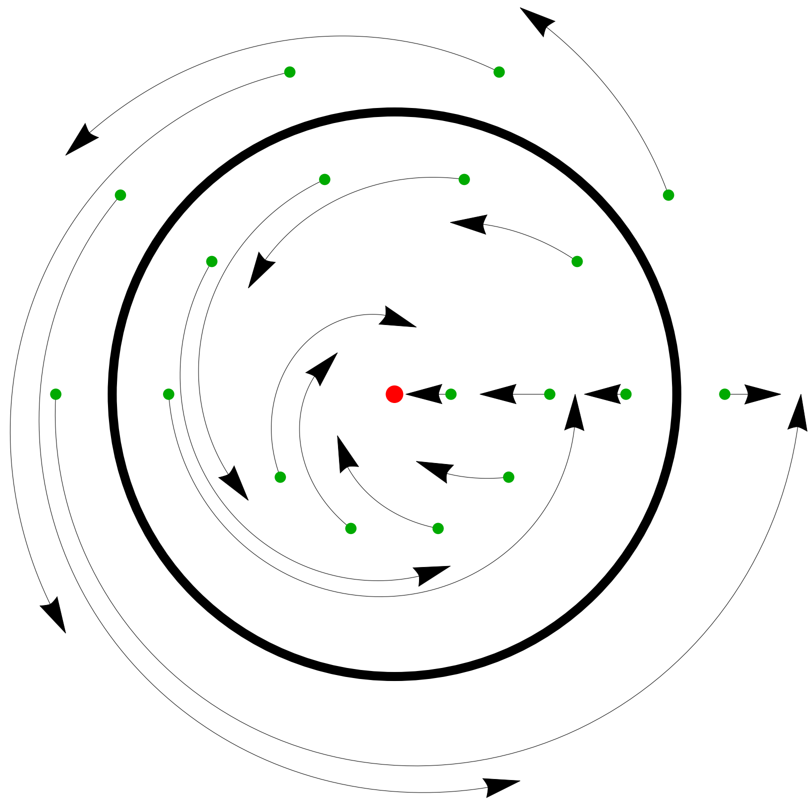

Figure 3.1.1: The dynamics of z^2

Let F denote the complement of the unit circle. Thus, F is an open set consisting of two, disjoint parts - the interior of the unit circle and the exterior. The dynamics of f on F are stable in a concrete sense. Suppose that $z_0 \in F$. Then, there is a disk D centered at z_0 whose radius is small enough that $D \subset F$. This means that D lies entirely within the interior of the unit circle or within the exterior. Either way, the long term behavior of every point in D is the same as the long term behavior of z_0. This is exactly what we mean by stability in F: changing the initial input a little bit does *not* change the long term behavior.

The dynamics right on the unit circle C, by contrast are quite complicated. We'll show later that the dynamics on C displays all of the hallmarks of chaos. For the time being, we claim that f displays *sensitive dependence to initial conditions*. That is, given any point $z_0 \in C$ and any $\varepsilon > 0$, there is some $w_0 \in \mathbb{C}$ satisfying $|z_0 - w_0| < \varepsilon$ whose orbit is drastically different from the orbit of z_0. In particular, it is possible that the orbit of w_0 tends to ∞ while the orbit of z_0 stays on the unit circle. The details are left as an exercise.

3.2 The filled Julia set

When we iterate the function $f(z) = z^2$ from any starting point $z_0 \in \mathbb{C}$ we find that there are two, very general possibilities:

- The orbit stays bounded or
- the orbit diverges to ∞.

In fact, this is true for *any* function f_c in the quadratic family, a fact that follows easily from theorem Theorem 3.2.1.

Theorem 3.2.1 The quadratic escape criterion. *Let*

$$f_c(z) = z^2 + c.$$

Then, the orbit of z_0 diverges to ∞ whenever $|z_0|$ exceeds

$$R = \max(2, |c|).$$

The number R is called the escape radius *for the quadratic.*

Proof. Suppose that $z_0 \in \mathbb{C}$ satisfies $|z_0| > 2$ and $|z_0| \geq |c|$. Then, by the reverse triangle inequality,

$$\begin{aligned} |z_1| = |f(z_0)| = |z_0^2 + c| &\geq |z_0|^2 - |c| \\ &\geq |z_0|^2 - |z_0| = |z_0|\,(|z_0| - 1) = \lambda |z_0|, \end{aligned}$$

where $\lambda > 1$. Furthermore, z_1 now also satisfies $z_1 > R$ so that, by induction, $|z_n| \geq \lambda^n |z_0|$. As a result, $z_n \to \infty$ as $n \to \infty$. ■

Note that this applies to any iterate z_i even if $z_0 \leq 2$. As a result, once some iterate exceeds 2 in absolute value, the orbit is guaranteed to escape. As a result, a fundamental dichotomy arises whenever we iterate any function f_c in the quadratic family; this dichotomy gives rise the definition ofthe so-called *filled Julia set* of f_c.

Definition 3.2.2 The filled Julia set. Given a point $z_0 \in \mathbb{C}$, either:

- The orbit of z_0 stays bounded by $R = \max(2, |c|)$ under iteration of f_c, in which case we say that z_0 lies in the filled Julia set of f_c or
- the orbit of $z_0 >$ diverges to ∞ under iteration of f_c, in which z_0 does not lie in the filled Julia set of f_c.

◊

3.3 An algorithm for the filled Julia set

Theorem Theorem 3.5.1 yields an algorithm for generating Julia sets of quadratics.

Algorithm 3.3.1 The escape time algorithm for filled Julia sets.

1. *Choose and fix a number $c \in \mathbb{C}$. We'll generate the filled Julia set of f_c,*
2. *Choose some rectangular region in the complex plane bound on the lower left by, say, z_{min} and on the upper right by z_{max}.*
3. *Partition this region into large number of rows and columns. We'll call the intersection of a row and column a "pixel", which corresponds to a*

complex value z_0 which will be used an initial seed for the iteration of f_c.

4. *For each pixel, iterate f_c from z_0 until one of two things happens:*
   (a) *We exceed the escape radius R in absolute value, in which case we shade the pixel according to how many iterates it took to escape.*
   (b) *We exceed some pre-specified maximum number of iterations, in which case we color the pixel black.*

An implementation of this algorithm is shown below.

```
from matplotlib import pyplot as plt
from numpy import zeros

def orbit_cnt(c,z0):
    R2 = max(abs(c),2)**2
    z = z0
    cnt = 0
    while cnt < 100 and z.real**2 + z.imag**2 <= R2:
        cnt = cnt + 1
        z = z**2+c
    return cnt

def orbit_cnts(c):
    res = 300
    delta = 4.0/res
    counts = zeros((res,res))
    for i in range(res):
        for j in range(res):
            z0 = complex(-2 + j*delta, 2 - i*delta)
            count = orbit_cnt(c,z0)
            if count < 100:
                counts[i,j] = count
    return counts


counts = orbit_cnts(-1.2 + 0.2j)
plt.figure(figsize = (5,5))
plt.imshow(counts, extent=[-2,2,-2,2], cmap='gist_stern')
plt.show()
```

3.4 Another look at conjugacy

In this section, we'll look at some consequences of conjugacy within the quadratic family.

3.4.1 Representing *all* quadratics

One reason that the quadratic family is so important is that the dynamics of *every* quadratic is captured by the behavior. This is made precise in Checkpoint 3.4.1

Checkpoint 3.4.1 Let $g(z) = \alpha z^2 + \beta z + \gamma$ and let $\phi(z) = az + b$. Show that $f_c \circ \varphi = \varphi \circ g$ when $a = \alpha$, $b = \beta/2$, and

$$c = \frac{1}{4}\left(4\alpha\gamma - \beta^2 + 2\beta\right).$$

That is, φ conjugates f_c to g.

Checkpoint 3.4.2 Let $g(z) = (1+i)z^2 - z + i/4$. Find values of a, b, and c such that $\varphi(z) = az + b$ conjugates f_c to g. Use the computer to generate images of the Julia sets of both f_c and g.

3.4.2 The chaotic squaring map

Checkpoint 3.4.3 Let $H = [0, 1)$ denote the half-open/half-closed unit interval and let $d : H \to H$ denote the doubling map

$$d(x) = 2x \mod 1.$$

Show that $\varpi(x) = e^{2\pi i x}$ defines a conjugacy between d and the complex function $f_0(z) = z^2$ restricted to the unit circle.

3.4.3 A line segment

We now let $c = -2$ so that $f_c(z) = z^2 - 2$. From our work in real iteration, we already know this function maps $[-2, 2] \to [-2, 2]$ and that it is chaotic on that interval. On the complement, $\mathbb{C} \setminus [-2, 2]$, it turns out that f_{-2} is stable. In fact, the iterates of f_{-2} diverge to ∞ for every initial point $z_0 \in \mathbb{C} \setminus [-2, 2]$.

To see this, let $F = \mathbb{C} \setminus [-2, 2]$ and let E denote the exterior of the unit disk, i.e. $E = \mathbb{C} \setminus \{z : |z| \leq 1\}$. We'll show that the action of f_{-2} on F is conjugate to the action of the squaring function $g(z) = z^2$ on E.

The conjugacy function will be $\varphi(z) = z + 1/z$. The geometric action of φ is shown in figure Figure 3.4.4 We first show that φ maps the boundary of E (the unit circle) to the boundary of F (the interval $[-2, 2]$). Of course, the points on the unit circle are exactly those points of the form e^{it} for some $t \in [0, 2\pi)$. Thus, we compute

$$\begin{aligned}
\varphi(e^{it}) &= e^{it} + e^{-it} \\
&= (\cos(t) + i\sin(t)) + (\cos(t) - i\sin(t)) \\
&= 2\cos(t).
\end{aligned}$$

The expression $2\cos(t)$ traces out the interval $[-2, 2]$ (twice, in fact) as t ranges through $[0, 2\pi)$ as claimed.

We next show that φ is one-to-one on E. To this end, suppose that

$$z + \frac{1}{z} = w + \frac{1}{w}.$$

Then,

$$z - w = \frac{1}{w} - \frac{1}{z} = \frac{z - w}{wz}.$$

Assuming that $z \neq w$, we can divide off the numerators and then take reciprocals to obtain $wz = 1$. Thus, if z is a point in the exterior of the unit disk, there is exactly one other point w such that $\varphi(w) = \varphi(z)$, namely the reciprocal of z which lies in the interior of the unit disk.

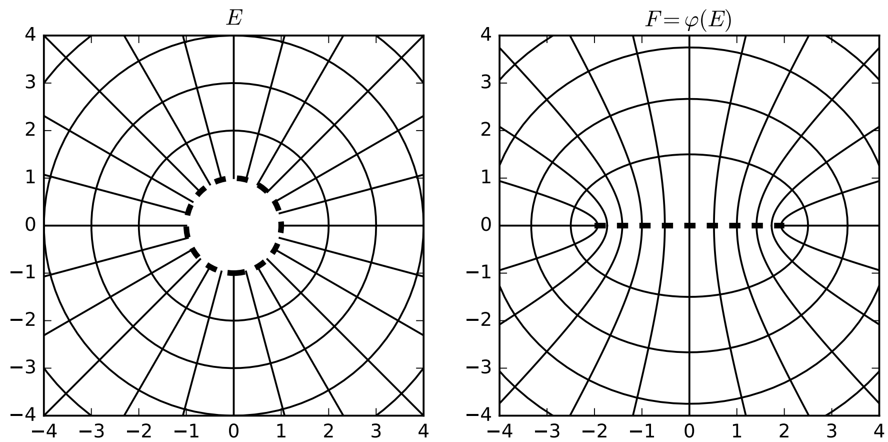

Figure 3.4.4: The conjugacy from F to E

3.5 The critical orbit

Each function $f_c(z) = z^2 + c$ has exactly one critical point at the origin. It turns out that the orbit of this critical point dominates the global dynamics of the iteration of f_c.

Theorem 3.5.1 f_c

1. *the critical orbit stays bounded by the escape radius, in which case the Julia set and filled Julia sets are connected, or*

2. *the critical orbit diverges to ∞, in which case the Julia set is a Cantor set.*

For the time being, we will content ourselves with a proof that the Julia set is totally disconnected when $|c| > 2$ and a sketch of the rest. The general idea is based on a process called *inverse iteration*. Given and function $f : \mathbb{C} \to \mathbb{C}$ and a set $S \in \mathbb{C}$ let

$$f^{-1}(S) = \{z \in \mathbb{C} : f(z) \in S\}.$$

The set $f^{-1}(S)$ is called the *inverse image* or *pre-image* of S. We define an initial approximation J_0 and let $J_n = f_c^{-n}(J_0) \equiv f_c^{-1}(J_{n-1})$. As we will see, if the initial approximation is chosen correctly, the sequence of iterates will collapse down to filled Julia set.

Proof. Suppose that $|c| > 2$ and consider the disk D of radius $|c|$ centered at the origin. By the proof of theorem Theorem 3.2.1, every point on the boundary of this disk maps to a point whose absolute value is larger than $|c|$. As a result, the image of the boundary is a loop that completely encircles D. (In fact, it's a circle of radius $|c|^2$ centered at c, as you'll show in exercise Exercise 3.8.1.) Thus, $f(D) \supset D$. As a result, $f^{-1}(D) \subset D$. We now perform inverse iteration from D.

To help us understand $f^{-1}(D)$, note that $f_c(0) = c$, which lies on the boundary of D. In fact, $f_c(z) = z^2 + c = c$ is equivalent to $z^2 = 0$ so that zero is the *only* point that maps to c. Every other point on the boundary of D has exactly two pre-images. Thus, the pre-image of the boundary of D is a figure 8 with a cut point at the origin. The pre-image of D itself consists of two lobes bounded by the two curves forming the figure 8.

We next consider $f_c^{-2}(D)$. This set consists of two separate pieces, one in each of the lobes of $f_c^{-1}(D)$. More generally, $f_c^{-n}(D)$ consists of 2^{n-1} obtained by cutting the pieces of $f_c^{-(n-1)}$ in half.

Finally, the filled Julia set itself is exactly

$$\bigcap_{n=1}^{\infty} f_c^{-n}(D).$$

As the diameters of these sets decrease down to zero, the filled Julia set coincides exactly with the Julia set. The construction above yields an obvious correspondence between the points in the Julia set and the points in the Cantor set.

The construction just described in illustrated in figure Figure 3.5.2(a). ■

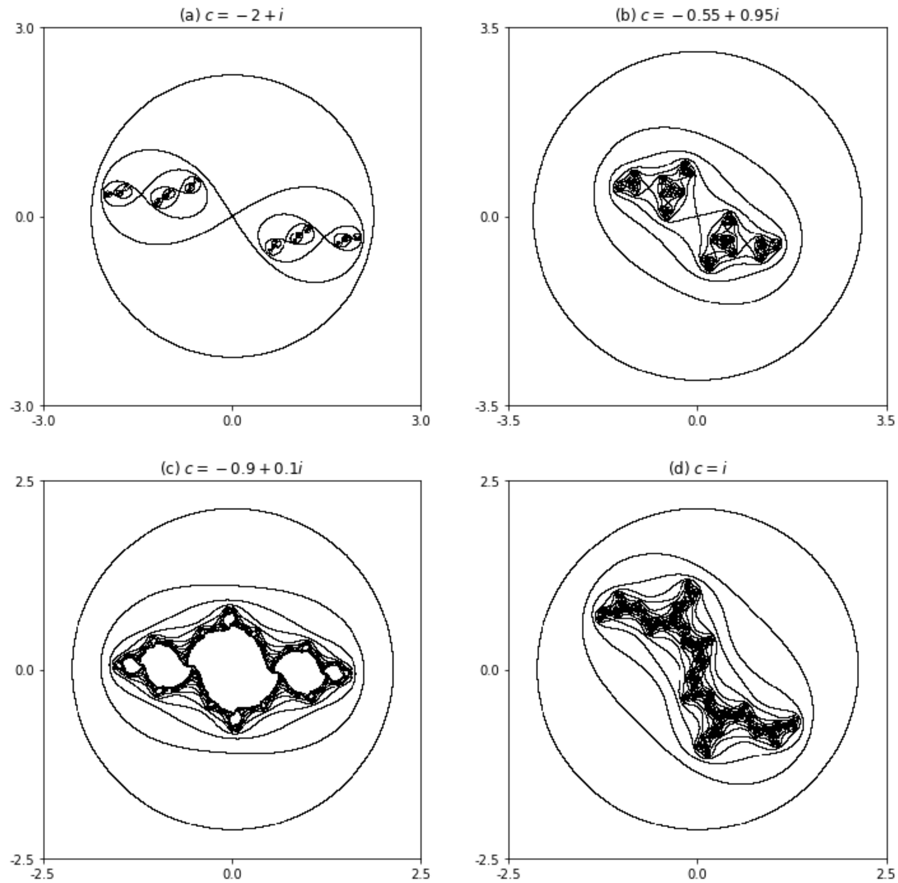

Figure 3.5.2: Inverse images collapsing to filled Julia sets

The situation just described is more complicated when $|c| \leq 2$. If $|c| \leq 2$ and the critical orbit escapes, then there is some first value of n with the property that $|f_c^n(0)| > 2$. We can then apply the inverse iteration idea to the disk of radius $|f_c^n(0)|$ centered at the origin. The figure 8 configuration then appears after n iterates and the ideas of the proof are then applicable. This situation is illustrated in figure Figure 3.5.2(b).

If the critical orbit does not escape, then we can attempt inverse iteration with a disk D whose radius is larger than two. In this case, the figure 8 never forms so that $f^{-n}(D)$ is a connected set for all n. As a result, the filled Julia set is a connected set since it's the intersection of a nested family of connected sets. This set is illustrated in Figure 3.5.2(c) and (d). In figure (c), f_c has an attractive orbit. In figure (d), f_c does not have an attractive orbit so that the sets $f^{-n}(D)$ collapse down to have area zero. Since any attractive behavior must attract the critical orbit, one easy way to ensure the lack of attractive behavior is to chose c so that the orbit of zero lands exactly on a repelling orbit. In fact, one can prove that this always happens when zero is pre-periodic. In Figure 3.5.2(d), $c = i$ and zero eventually maps to an orbit of period 2.

3.6 The Mandelbrot set

Theorem Theorem 3.5.1 expresses a fundamental dichotomy amongst quadratic Julia sets. There are two types: connected and totally disconnected. Furthermore, the type that arises from a particular value of c depends completely on the orbit of the critical point. Thus, it might be fruitful to partition the parameter plane into two portions: the portion where the critical orbit stays bounded and the portion where the critical orbit diverges to infinity. This partition leads to one of the most iconic images from the study of chaos - the Mandelbrot set.

Definition 3.6.1 The Mandelbrot set. The *Mandelbrot set* is the set of all complex parameters c such that the Julia set of the function $f_c(z) = z^2 + c$ is connected. Equivalently, by theorem Theorem 3.5.1, it is the set of all c values such that the orbit of zero under iteration of f_c is bounded by 2 ◊

Theorem Theorem 3.5.1 yields an algorithm to generate images of the Mandelbrot set that's very similar to our algorithm for generating Julia sets of quadratics.

Algorithm 3.6.2 The escape time algorithm for the Mandelbrot set.

1. *Choose some rectangular region in the complex plane bound on the lower left by, say,* c_{min} *and on the upper right by* c_{max}.
2. *Partition this region into large number of rows and columns. We'll call the intersection of a row and column a "pixel", which corresponds to a complex value of the parameter* c.
3. *For each pixel, iterate the corresponding function* f_c *from the origin until one of two things happens:*
   (a) *We exceed the escape radius 2 in absolute value, in which case we shade the pixel according to how many iterates it took to escape.*
   (b) *We exceed some pre-specified maximum number of iterations, in which case we color the pixel black.*

```
from matplotlib import pyplot as plt
from numpy import zeros

def critical_orbit_cnt(c):
    z = 0.0
    cnt = 0
    while cnt < 100 and z.real**2 + z.imag**2 <= 4:
        cnt = cnt + 1
        z = z**2+c
    return cnt

def critical_orbit_cnts():
    res = 400
```

```
    delta = 2.6/res
    counts = zeros((res,res))
    for i in range(res):
        for j in range(res):
            c = complex(-2 + j*delta, 1.3 - i*delta)
            count = critical_orbit_cnt(c)
            if count < 100:
                counts[i,j] = count
    return counts


counts = critical_orbit_cnts()
plt.figure(figsize = (5,5))
plt.imshow(counts, extent=[-2,2,-2,2], cmap='gist_stern')
plt.show()
```

The results of this algorithm are shown in figure Figure 3.6.3. While the algorithm is very similar to our algorithm for generating Julia sets, it's very important to understand the distinctions. The Mandelbrot set lives in the *parameter plane*, while Julia sets live in the *dynamical plane*. For the Julia set of a quadratic function f_c, we fix the value of c and explore the behavior of the orbit of z_0 for a grid of choices of z_0. For the Mandelbrot set, we explore the global behavior of f_c by examining the critical orbit; we do this for a grid of choices of c.

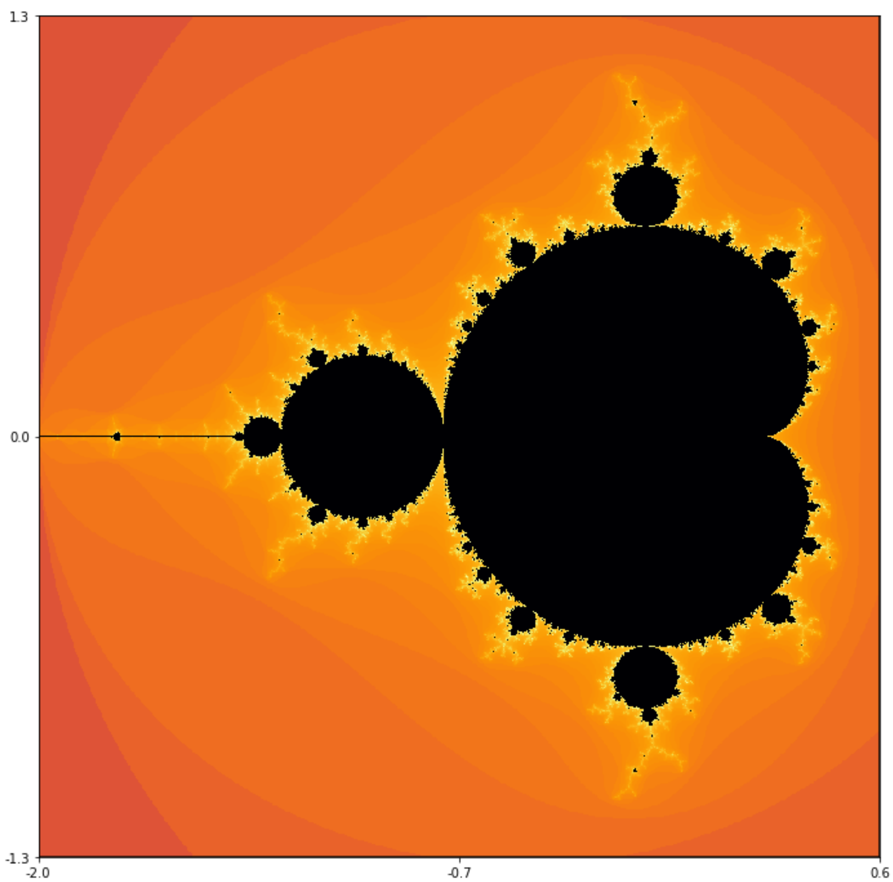

Figure 3.6.3: The Mandelbrot set

It's worth listing a few elementary properties of the Mandelbrot set.

- The Mandelbrot set is *bounded*. In fact, our escape criterion for quadratics implies that the Mandelbrot set is contained in the disk of radius 2 centered at the origin.
- The Mandelbrot set is *closed*. This follows fairly easily from the continuity of f_c^n.
- The Mandelbrot set is *connected*. While this an elementary statement to understand, it's not so easy to prove.

3.7 The components of the Mandelbrot set

By definition, the Mandelbrot set decomposes the complex parameter plane into two components corresponding to a very coarse decomposition of the types of behavior of f_c. A glimpse at figure Figure 3.6.3 (or any of the many images of it available on the web) shows that it consists of rather conspicuous subparts. It appears that there is a large "main cardioid" and that, attached to that main cardioid are a slew of disks or near disks. If we zoom in, we see many smaller copies of the whole set. As it turns out, all these components correspond to particular dynamical behavior. Values of c chosen from within one component yields functions f_c with similar qualitative behavior. As c moves from one component to another, that qualitative behavior changes; we say that a *bifurcation* occurs. In this section, we'll describe the most conspicuous behavior that we see.

3.7.1 Quadratics with super-attractive orbits

We've already seen that $f_0(z) = z^2$ has a super-attractive fixed point at the origin and that the points $z_0 = 0$ and $z_1 = -1$ form a super-attractive orbit of period 2 for $f_{-1}(z) = z^2 - 1$. We also know, from our work with real iteration, that there is a c value around -1.75 so that f_c has a super-attractive orbit of period 3. It turns out that there are two more *complex* super-attractive period 3 parameters. We can find them with the following Sage code:

```
z,c = var('z,c')
F(c,z) = z
for i in range(3):
    F(c,z) = F(c, z^2 + c)
F(c,0).roots(ring=CC)
```

```
[(-1.75487766624669, 1),
 (0.000000000000000, 1),
 (-0.122561166876654 - 0.744861766619744*I, 1),
 (-0.122561166876654 + 0.744861766619744*I, 1)]
```

Listing 3.7.1: Sage code to find the complex super-attractive period three parameters.

Checkpoint 3.7.2 Locate the points from output of the previous code in the Mandelbrot set and sketch the corresponding Julia sets.

Checkpoint 3.7.3 Find the complex $c$values such that f_c has a super-attractive orbit of period 4 or 5. Locate these points in the Mandelbrot set and sketch the corresponding Julia sets.

3.7.2 The main cardioid

The interior of the main cardioid consists of all those values of c with the property that f_c has an attractive fixed point. This can be characterized algebraically as

$$\begin{aligned} f_c(z) &= z \\ |f_c'(z)| &< 1. \end{aligned}$$

Now, on the *boundary* we should have

$$\begin{aligned} f_c(z) &= z \\ |f_c'(z)| &= 1 \end{aligned}$$

That is, the fixed point is no longer strictly attractive on the boundary but just neutral. Since any complex number of absolute value 1 can be expressed in the form $e^{2\pi it}$, this pair of equations can be written without the absolute value:

$$\begin{aligned} f_c(z) &= z \\ f_c'(z) &= e^{2\pi it}, \end{aligned}$$

for some t. Taking into account the fact that $f_c(z) = z^2 + c$, we get

$$\begin{aligned} z^2 + c &= z \\ 2z &= e^{2\pi it}. \end{aligned}$$

This system of equations can be solved for z and c in terms of t. It's not even particularly hard. The second equation yields $z = e^{2\pi it}/2$. This can be plugged into the first equation to get

$$c = e^{it}/2 - e^{2it}/4.$$

This happens to be the parametric representation of a cardioid and is exactly the formula used to generate figure Figure 3.7.4

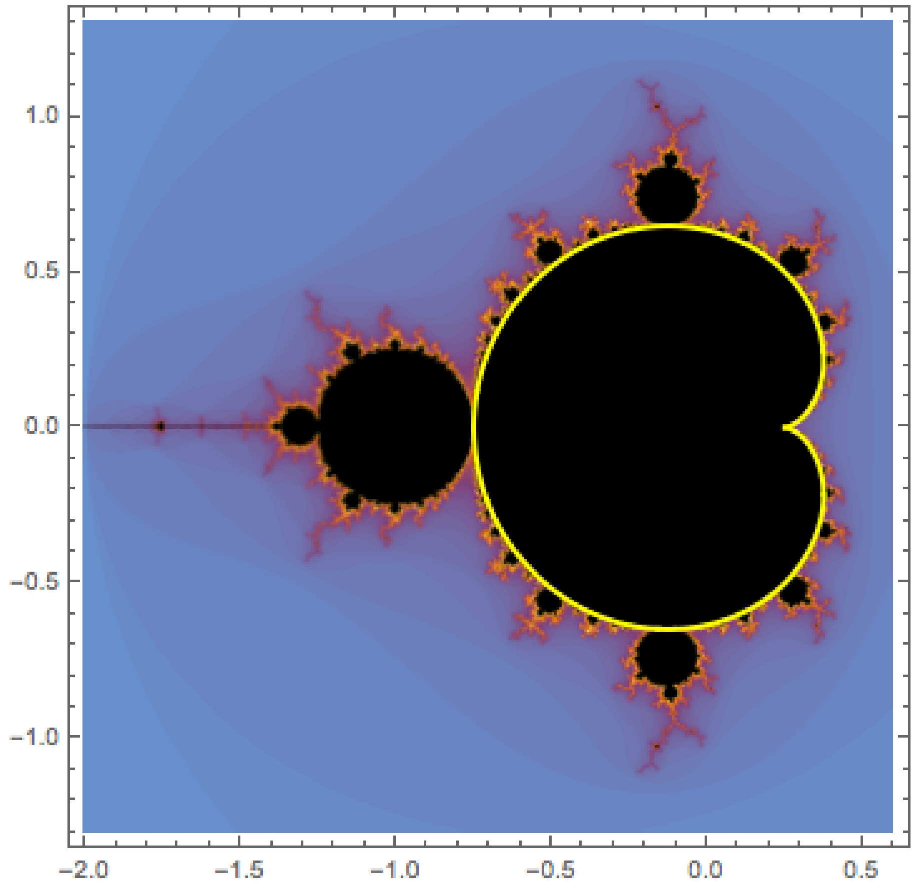

Figure 3.7.4: The Mandelbrot set with the main cardioid highlighted

3.7.3 The period two bulb

After the main cardioid, the next largest component is the disk attached to the main cardioid at the point $c = -3/4$. This is, in fact, an actual disk attached at that exact point. We can prove this in much the same way that we derived the formula for the main cardioid, though the algebra is a little more involved.

This disk is known as the *period two bulb*; every function f_c where c is chosen from the period two bulb has an attractive orbit of period two. We can characterize this algebraically by first defining $F_c(z) = f_c(f_c(z))$. A point is then part of an attractive orbit of period 2 for f_c if

$$F_c(z) = z$$
$$|F_c'(z)| < 1.$$

Mimicking the construction of the main cardioid in the period 1 case, we find that z and c must satisfy

$$F_c(z) = z$$
$$F_c'(z) = e^{2\pi i t},$$

for some t. Taking into account the fact that $F_c(z) = (z^2 + c)^2 + c$, we get

$$(z^2 + c)^2 + c = z$$
$$4z(z^2 + c) = e^{2\pi i t}.$$

While clearly this is a bit more involved than the period one case, a computer algebra system makes quick work of it. Listing Listing 3.7.5 shows how to solve this system with Mathematica and listing Listing 3.7.6 shows how to solve this system with Sage.

```
f[c_][z_] = z^2 + c;
F[c_][z_] = Nest[f[c], z, 2];
eqs = {F[c][z] == z, F[c]'[z] == Exp[2 Pi*I*t]};
Expand[c /. Last[Solve[eqs, {c, z}]]]
(* Out:
        -1 + E^((2*I)*Pi*t)/4
*)
```

Listing 3.7.5: Mathematica code to parametrize the period two bulb

```
z,c,t=var('z,c,t')
def F(c,z): return (z^2+c)^2+c
solve([F(c,z)==z,F(c,z).diff(z)==exp(2*pi*I*t)],[c,z])[0]
```

```
# Out:
[c == 1/4*e^(2*I*pi*t) - 1, z == -1/2*sqrt(-e^(2*I*pi*t) + 1)
    - 1/2]
```

Listing 3.7.6: Sage code to parametrize the period two bulb

Note that both program listings Listing 3.7.5 and Listing 3.7.6 yield the result

$$c = -1 + \frac{1}{4}e^{2\pi i t},$$

which parametrizes a circle of radius $1/4$ centered at the point -1.

Also, it's certainly feasible to solve this system by hand. The expression $F_c(z) - z$ factors so that the system really just involves a couple of quadratics. Use of the computer will be quite convenient as we move into higher order examples, though.

3.7.4 The Mandelbrot set as a bifurcation locus

We've now got a solid understanding of the period 1 component of the Mandelbrot set (the main cardioid) and the period 2 component (the period 2 bulb). The period 2 component is called a "bulb" because it is directly attached to the main cardioid. Indeed, using our parametrizations, you can show easily enough that the point $c = -3/4$ lies on both the boundary of the main cardioid and on the boundary of the period 2 component. Note that, for this value of c, f_c has a neutral fixed point at $z = -1/2$. As c moves from the main cardioid, through the point $-3/4$, and into the period 2 bulb, the function f_c undergoes a bifurcation - that is a qualitative change in the global dynamics of the function. More specifically, f_c changes from having an attractive fixed point to having an attractive orbit of period 2. This is very much like the bifurcation diagram that we saw when studying real iteration. In fact:

The Mandelbrot set is a version of the bifurcation diagram in the context of complex dynamics, rather than real dynamics.

The geometric relationship between the Mandelbrot set and the bifurcation diagram is made explicit in figure Figure 3.7.7.

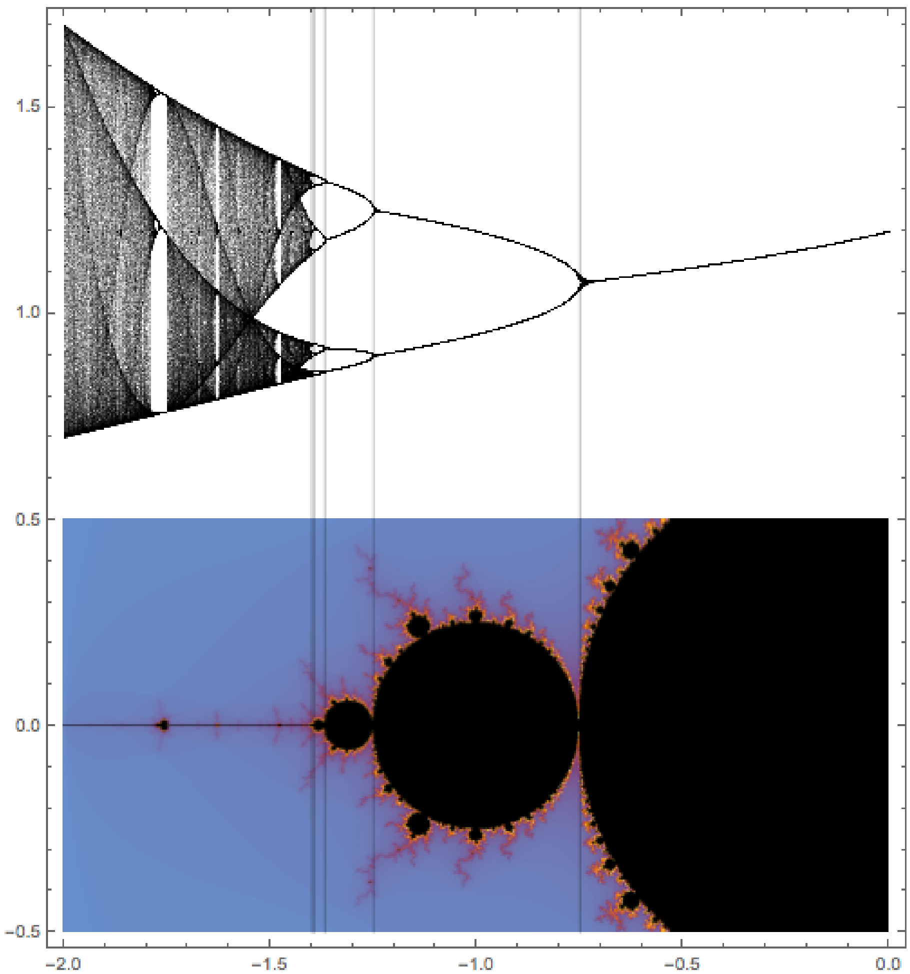

Figure 3.7.7: The Mandelbrot set aligned with the bifurcation diagram

Figure Figure 3.7.7 shows only how the Mandelbrot set aligns with the real bifurcation diagram. There is really much more to the picture, though. All the visible components of the Mandelbrot set correspond to distinct, qualitative behavior. As c moves within one component, the qualitative behavior of f_c stays the same. More precisely, any two values of c chosen from the same component will have attractive orbits of the same period. As c moves from any component of the Mandelbrot set to another in the complex plane, f_c undergoes a bifurcation - period doubling, tripling, or whatever. If c exits the Mandelbrot set, f_c undergoes a more complicated type of bifurcation.

This all suggests another way to draw the Mandelbrot set. We iterate from the critical point zero, as in algorithm Algorithm 3.6.2, but we stop iteration whenever we find some sort of attractive behavior. We then color the point c according to the period of the orbit we find and shade it according to how long it took to detect the periodicity. The result is shown in figure Figure 3.7.8.

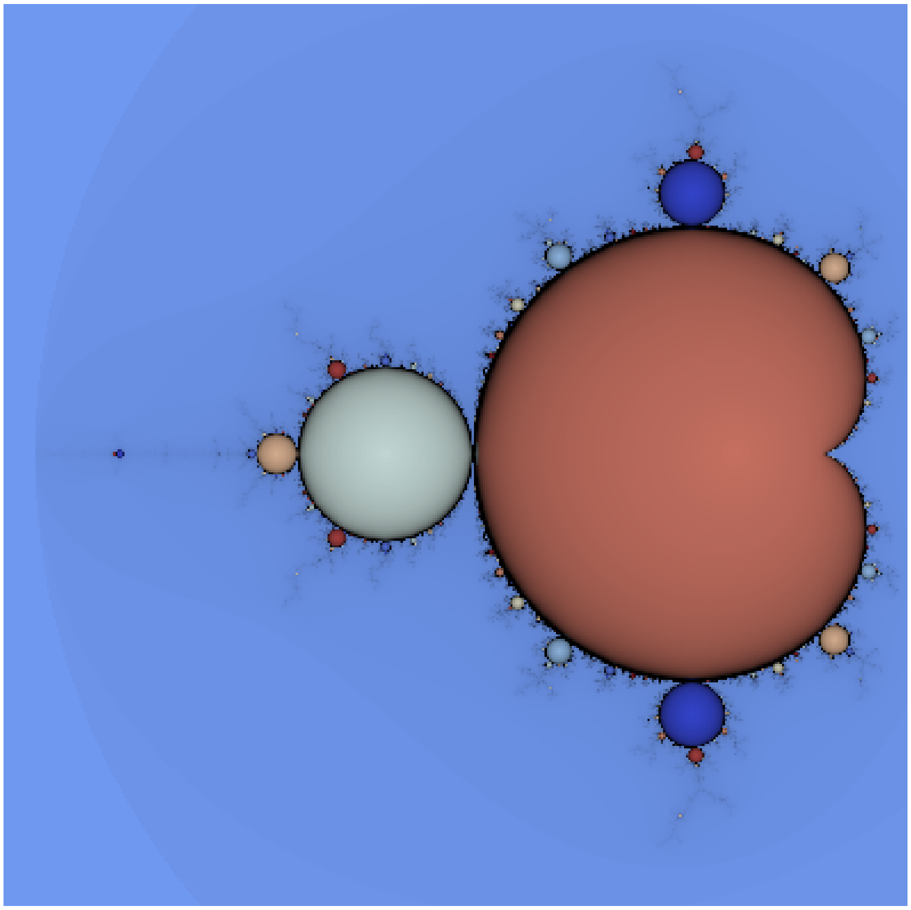

Figure 3.7.8: The components of the Mandelbrot set shaded according to periodicity

3.7.5 Higher period bulbs

Figure Figure 3.7.8 shows that there are lots of disk-like bulbs attached to the main cardioid. As c moves from the main cardioid into one of these bulbs, f_c bifurcates from having an attractive fixed point, to an orbit of period q for some integer $q > 1$. After the period 2 bulb, the two next largest bulbs near the top and bottom of the main cardioid are period 3 bulbs. The other bulbs have larger period still.

Let's try to gain a quantitative understand of the nature of these bulbs. To do so, we re-consider our description of the main cardioid from subsection Subsection 3.7.2. Again, the interior of the main cardioid can be characterized algebraically as

$$\begin{aligned} f_c(z) &= z \\ |f_c'(z)| &< 1. \end{aligned}$$

More generally, though, we can obtain a fixed point with any given multiplier γ that we like by solving

$$\begin{aligned} f_c(z) &= z \\ f_c'(z) &= \gamma. \end{aligned}$$

If we solve this system for c and z, we obtain

$$c = \frac{1}{4}\left(2\gamma - \gamma^2\right)$$
$$z = \gamma/2.$$

Thus, f_c has a fixed point with multiplier γ at $\gamma/2$ when $c = (2\gamma - \gamma^2)/4$.

Let us now define $c(\gamma) = (2\gamma - \gamma^2)/4$ and consider the image of rays of the form $r\exp(2\pi i\theta)$, where θ is fixed and $0 \leq r \leq 1$. These should all lie within the main cardioid and stretch from zero to a point on the boundary of the cardioid. Some of these are plotted in figure Figure 3.7.9.

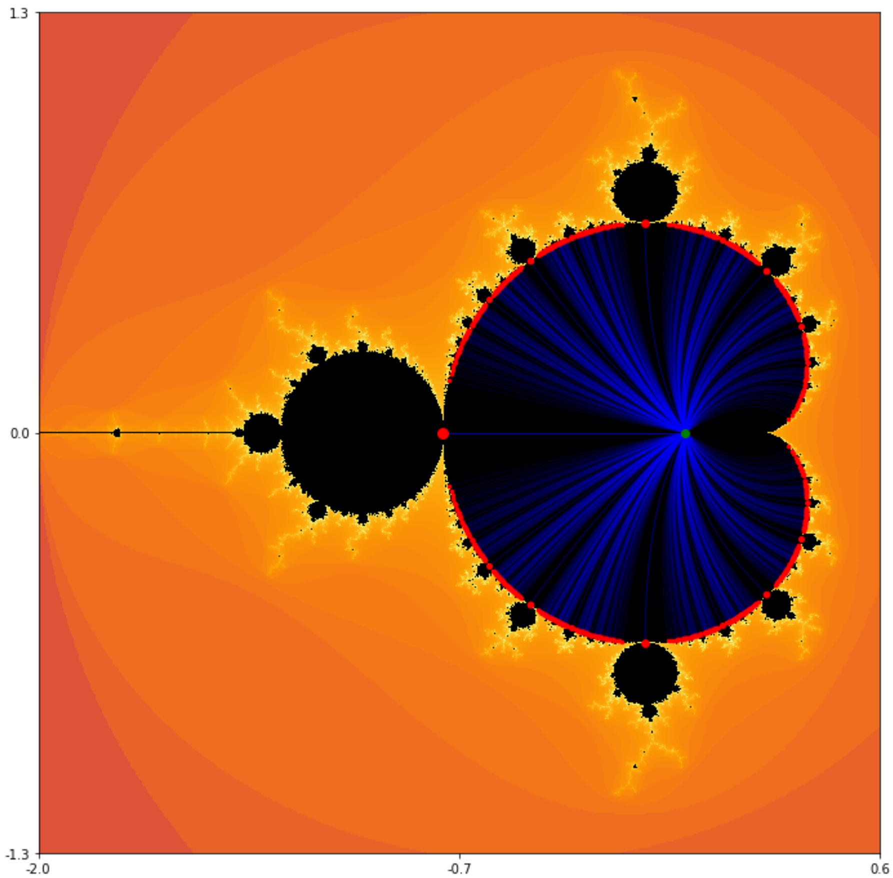

Figure 3.7.9: Bulbs located by the cardioid parametrization

Note that the rays shown in figure Figure 3.7.9 all correspond to rational values of θ and, as it appears, a periodic bulb is attached to the main cardioid at every rational ray. Even more is true - if $\theta = p/q$, then the bulb attached to the main cardioid at the point $c(\exp(2\pi i\theta))$ has period q and rotation number p.

To understand this, recall that $c(\gamma)$ is built specifically so that $f_{c(\gamma)}$ will have multiplier γ. So, what happens to the attractive orbits of $f_c(\gamma)$ for

$$\gamma = re^{2\pi i\theta}$$

for a fixed rational value of θ as r moves from zero up to one and even just a little bit past? For $r < 1$, f_c has an attractive fixed point. For $r = 1$, f_c has a neutral fixed point with multiplier $\gamma = \exp(2\pi i\theta)$. Note that the geometric effect of multiplying by this particular value of γ is to rotate through the angle $2\pi\theta = 2\pi p/q$. If we do this q times, we've rotated through the angle $2\pi\theta = 2\pi p$, which returns to where we started. We might expect an attractive orbit of period q; we don't have it yet because the origin still in the basin of the neutral fixed point. Once $r > 1$, that fixed point becomes slightly repulsive and the attractive orbit of period q appears.

Note that each of these bulbs has further bulbs sprouting off and similar considerations apply.

3.7.6 Baby-brots

There are loads of little copies of the Mandelbrot set strewn throughout. For the time being, I'll refer to this Math.StackExchange post: http://math.stackexchange.com/questions/404066/.

3.8 Exercises

1. Show that the image of the circle of radius $|c|$ centered at c under f_c is the circle of radius $|c|^2$ centered at the origin.

In the next couple of problems, we'll try to get a grip on the family of functions $g_\lambda(z) = \lambda z + z^2$.

2. The escape radius

(a) Use the triangle inequality to show that $|g_\lambda(z_0) \geq |z_0|$, whenever $|z_0| > 2\lambda$. Conclude that the orbit of z_0 escapes whenever an iterate excedes 2λ.

(b) How does this compare to our general polynomial escape criterion?

3. The *escape locus* for g_λ (or any family of quadratics) is a partition of the complex parameter plane into two regions - one where the critical orbit stays bounded and one where the critical orbit diverges. This escape locus is shown in figure Figure 3.8.1. Let's try to understand a couple of things about this image.

(a) Show that the origin is an attractive (or super-attracitve) fixed point of g_λ, whenever $|\lambda| < 1$. This observation yields what prominent feature in figure Figure 3.8.1?

(b) Let $\varphi(z) = z + (1 - \lambda)$. Show that

$$g_\lambda \circ \varphi = \varphi \circ g_{2-\lambda}.$$

How is this observation related to the symmetry that we see in figure Figure 3.8.1?

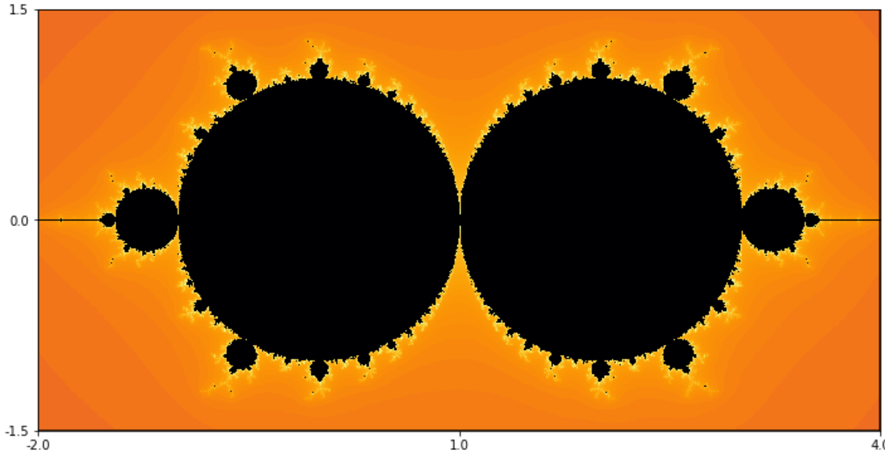

Figure 3.8.1: The escape locus for $g_\lambda(z) = \lambda z + z^2$.

Chapter 4

The iteration of complex polynomials

In this chapter, we'll generalize our work on quadratics to explore the iteration of polynomials more generally. Then, we'll narrow in on the family of cubic polynomials.

4.1 A general escape time algorithm

Although we've only looked at the quadratic family, it seems apparent (for polynomials, at least), that the orbit of any initial seed that is large enough will diverge to ∞. This is not at all hard to prove.

Theorem 4.1.1 Polynomial escape criterion. *Let*

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_0.$$

Then, the orbit of z_0 diverges to ∞ whenever $|z_0|$ exceeds

$$R = \max\left(2|a_n|, 2\frac{|a_{n-1}| + \cdots + |a_0|}{|a_n|}\right).$$

The number R is called the escape radius *for the polynomial.*

Proof. Suppose that $z_0 \in \mathbb{C}$ satisfies $|z_0| > 2|a_n|$ and

$$|z_0| > 2\frac{|a_{n-1}| + \cdots + |a_0|}{|a_n|}.$$

Then,

$$\begin{aligned} |z_1| = |f(z_0)| &= |a_n z_0^n + a_{n-1} z_0^{n-1} + \cdots + a_0| \\ &= |a_n z_0^n| \left|1 + \frac{a_{n-1}}{a_n z_0} + \cdots + \frac{a_0}{a_n z_0^n}\right|. \end{aligned}$$

Now, our condition on z_0 implies that

$$\begin{aligned} S \equiv \left|\frac{a_{n-1}}{a_n z_0} + \cdots + \frac{a_0}{a_n z_0^n}\right| &\leq \left|\frac{a_{n-1}}{a_n z_0}\right| + \cdots + \left|\frac{a_0}{a_n z_0^n}\right| \\ \leq \left|\frac{a_{n-1}}{a_n z_0}\right| + \cdots + \left|\frac{a_0}{a_n z_0}\right| &= \frac{|a_{n-1}| + \cdots + |a_0|}{|a_n z_0|} < \frac{1}{2}. \end{aligned}$$

Thus, the triangle inequality yields

$$|1| = |1 + S + (-S)| \leq |1 + S| + |-S| = |1 + S| + |S|$$

so that $|1 + S| \geq 1 - |S| \geq 1/2$. Applying this to our original inequality for z_1, we obtain

$$\begin{aligned} |z_1| &> |a_n z_0^n| \frac{1}{2} \geq |a_n||z_0||z_0| \frac{1}{2} \\ &> 2|z_0| \frac{1}{2} = z_0. \end{aligned}$$

Now, since $|z_1| > |z_0|$ we can write $|z_1| = \lambda |z_0|$ for some $\lambda > 1$. Furthermore, z_1 now also satisfies $z_1 > R$ so that, by induction, $|z_n| \geq \lambda^n |z_0|$. As a result, $z_n \to \infty$ as $n \to \infty$. ■

It's easy to strengthen this result. Suppose we iterate from an initial seed z_0 that may be less than the escape radius. If any iterate ever does exceed the escape radius in absolute value, then the orbit will escape. This yields the following corollary.

Corollary 4.1.2 *Suppose we iterate a polynomial from an initial seed z_0. Then, there are two mutually exclusive possibilities for the orbit of z_0*

1. *The orbit stays bounded by the escape radius, or*
2. *the orbit diverges to ∞.*

Theorem Theorem 4.1.1 ensures that case 2 happens for all polynomials. It's also easy to see that there are lots of points whose orbits stay bounded. Any fixed point, yields an orbit that stays bounded. Of course, there are always fixed points since the equation $f(x) = x$ yields a polynomial that always has at least one *complex* solution. For that matter, periodic orbits are also bounded and $f^n(x) = x$ yields *lots* of those when n is large.

Definition 4.1.3 Let $p : \mathbb{C} \to \mathbb{C}$ be a polynomial.

1. The *filled Julia set* of p is the set of all complex numbers whose orbits remain bounded under iteration of p.
2. The *Julia set* of p is the boundary of the filled Julia set.
3. The complement of the filled Julia set or, equivalently, the set of all points that diverge to ∞ is called the *basin of attraction of* ∞.

We will denote the filled Julia set of p by K_p and the Julia set of p by J_p. ♢

We should point out that there are other characterizations of the Julia set that work more generally than the definition we give here, simply because other classes of functions might not have an "escape radius". When we redefine the Julia set in those other contexts, it will also be our responsibility to prove that it is consistent with this first definition.

There are a few relatively easy observations that we can make about these sets. Suppose that for some initial seed z_0 and integer n, we have $|p^n(z_0)| > R$. Then, by continuity, the same will be true for points sufficiently close to z_0. Thus, the basin of attraction of p is an *open* set. As a consequence, the filled Julia set K_p is a closed set. The boundary of a set is, by definition, its closure minus its interior. Thus the Julia set itself is also closed and non-empty. In addition, any neighborhood of J_p contains points whose orbits diverge to ∞ and points that stay bounded. Thus, the dynamics of p near J_p display sensitive dependence on initial conditions.

Finally, we note that corollary Corollary 4.1.2 yields a nice algorithm to generate a filled Julia set.

Algorithm 4.1.4 The escape time algorithm (for polynomial Julia sets).

1. *Choose some rectangular region in the complex plane bound on the lower left by, say,* z_{min} *and on the upper right by* z_{max}*.*

2. *Partition this region into large number of rows and columns. We'll call the intersection of a row and column a "pixel", which corresponds to a complex number.*

3. *For each pixel, iterate* p *until one of two things happens:*

   (a) *We exceed the escape radius in absolute value, in which case we shade the pixel according to how many iterates it took to escape.*

   (b) *We exceed some pre-specified maximum number of iterations, in which case we color the pixel black.*

In spite of the simplicity of this section, many of the ideas here are similar to more complicated situations and we will see several variations of the escape time algorithm later. The remainder of our examples rely on application of this algorithm

4.2 A cubic with two attractive orbits

Consider the cubic polynomial

$$p(z) = -z^3 + (2.41154 - 0.133695i)z^2 - (0.090706 - 0.27145i),$$

whose filled Julia set is shown in figure Figure 4.2.1. This polynomial has attractive orbits of periods both two and three. A major point here is that more complicated behavior can arise, when the degree of the polynomial increases or, more generally, as the function becomes more complicated.

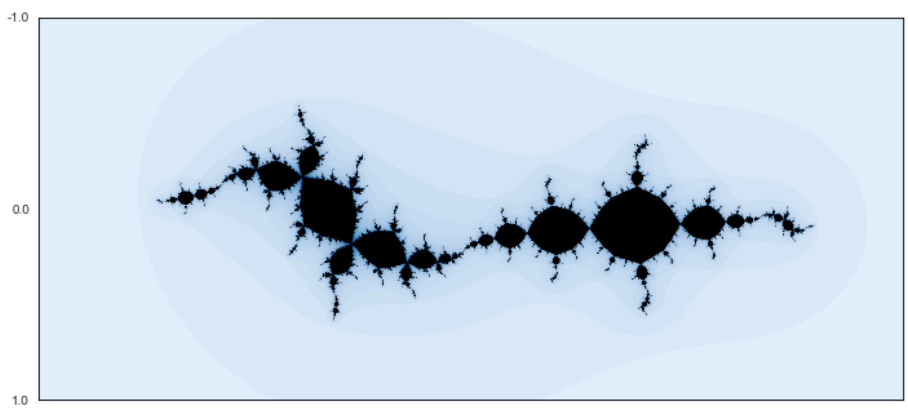

Figure 4.2.1: A cubic with two attractive orbits

An interactive orbit generator for this function is included on this webpage: https://www.marksmath.org/visualization/interactive_basins/.

4.3 Exercises

1. Consider iteration of the function $f(z) = z^3$.

(a) Show that zero is a super-attractive fixed point of f.

(b) Show that the orbit of z_0 tends to zero whenever $|z_0| < 1$ but diverges to ∞

(c) Explain precisely why f displays sensitive dependence on initial conditions.

(d) Compute the orbits of $e^{\pi i/3}$ and $e^{\pi i/4}$.

2. Let $f(z) = z^3 - z - 1$.

(a) Determine the fixed points of f and classify as attractive, repelling, or neutral.

(b) Plot the filled Julia set of f and indicate the locations of the the fixed points.

3. Let $f(z) = 2z^5 - z^4 + 3z^3 - 8z^2 + z - 1$. What is the escape radius guaranteed by theorem Theorem 4.1.1?