Week 7 class 3 - Chaos and Fractals

Phileas Dazeley-Gaist

17/02/2022

Today's goals

- Cellular Automata
- Julia Set Activity
- Hénon Map

Cellular Automata (CAs)

"A cellular automaton is a collection of"colored" cells on a grid of specified shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells. The rules are then applied iteratively for as many time steps as desired." (*Wolfram Mathworld*)

Cellular Automaton, *Wikipedia*

See things like Conway's game of life.

Things to think about:

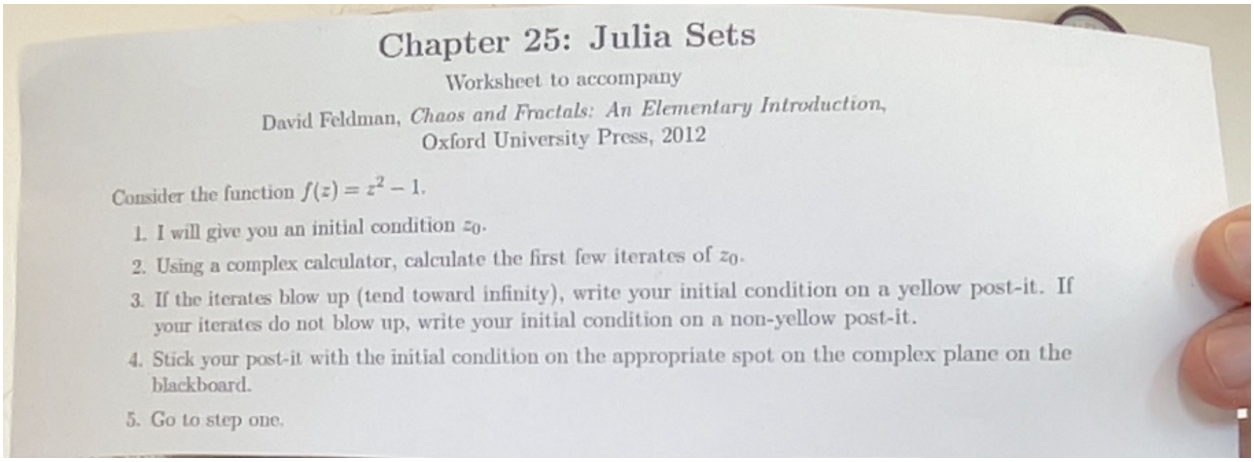
- Cellular automata are typically on grid worlds. But how would continuous-space-inhabiting cellular automata differ from grid-world-inhabiting ones?

Complex numbers

Complex numbers are numbers composed of real numbers + a number of imaginary units (i). The imaginary unit i has the property that $i^2 = -1$

Complex number coordinate plotter:

Let's do an quick experiment with complex numbers:



Chapter 25: Julia Sets

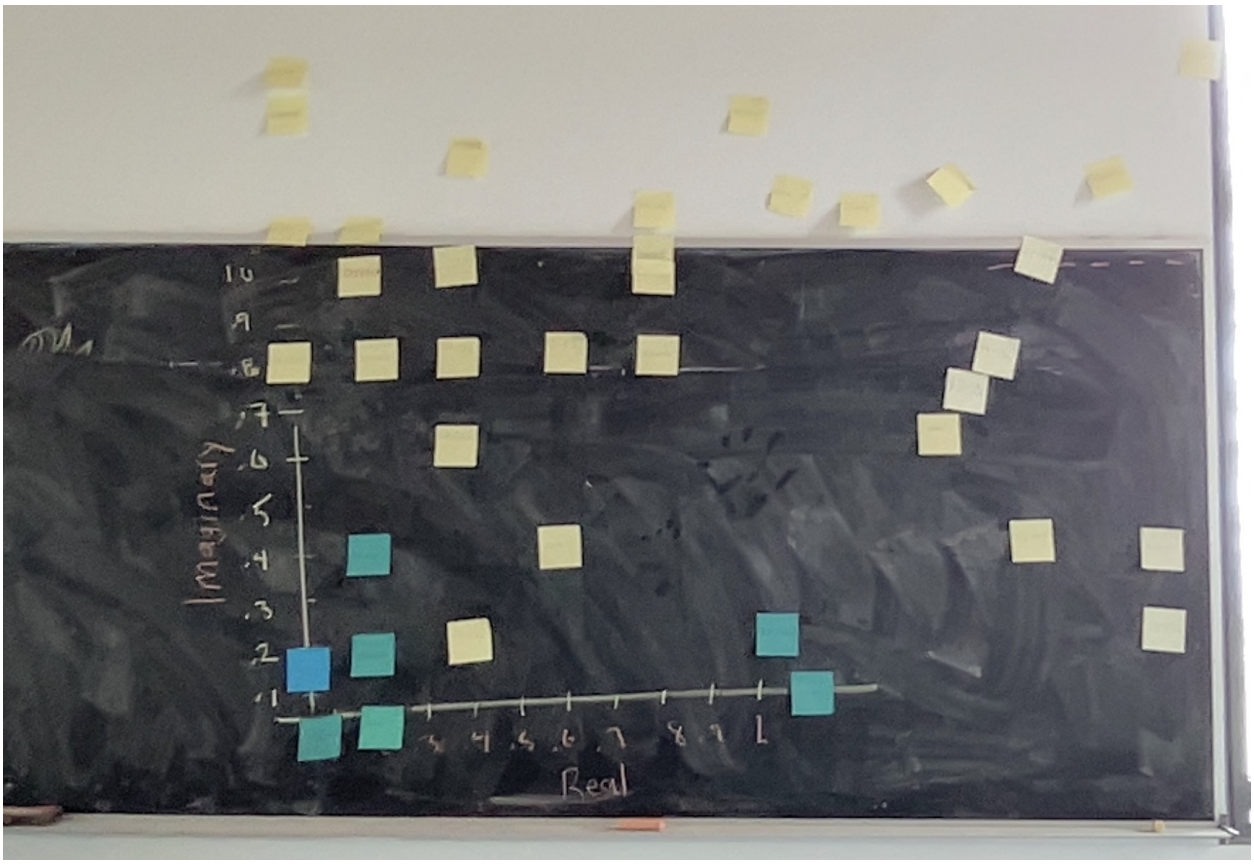
Worksheet to accompany

David Feldman, *Chaos and Fractals: An Elementary Introduction*, Oxford University Press, 2012

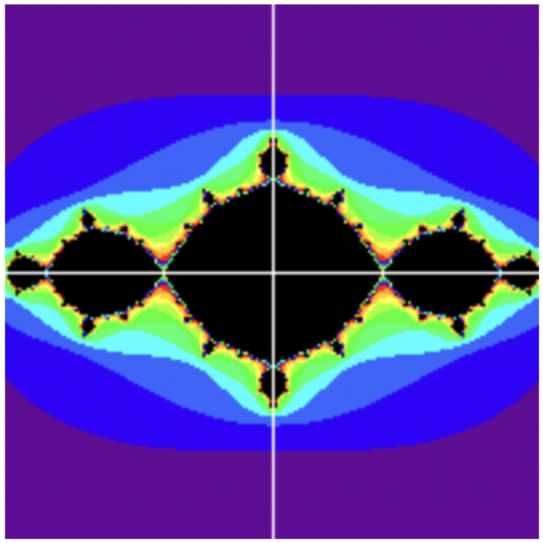
Consider the function $f(z) = z^2 - 1$.

1. I will give you an initial condition z_0.
2. Using a complex calculator, calculate the first few iterates of z_0.
3. If the iterates blow up (tend toward infinity), write your initial condition on a yellow post-it. If your iterates do not blow up, write your initial condition on a non-yellow post-it.
4. Stick your post-it with the initial condition on the appropriate spot on the complex plane on the blackboard.
5. Go to step one.

For some values of Z_0, iterating the function will cause the iterates to blow up to infinity.



Let's ask a computer to do this for us. We get something like this, where the colour coding represents the amount of time it takes for an orbit to blow up to infinity, and black means the orbit does not blow up to infinity:



In the in-class example, we were plotting only the upper-right quadrant.

We're now ready to talk about Julia Sets.

Julia Sets

A Julia set is the set of all initial conditions Z_0 that do not blow up to infinity when iterated through a given function. It could be any function. Julia sets are also called prisoner sets. One of the most common function types for generating Julia Sets are $Z^2 + C$, where C and Z are complex numbers (including purely real and purely imaginary numbers, $Z = 0 + 2i$, and $C = 3 + 0i$ both count). (Some Julia Sets are known as fractal cacti)

For a given initial condition, we can use a program to plot the corresponding orbit of the function on the complex plane like so:

Complex number function iterator:

```
options(scipen = 100)

n <- 15
z_0 <- -0.2+0.4i
c <- -1
funky <- function(z){return(z^2+c)}

z_n <- z_0
for(i in 1:(n-1)){
  z_n <- c(z_n, funky(tail(z_n, 1)))
}
z_ends <- shift(z_n, -1)[1:(n-1)]
z_ends <- append(z_ends, NA)
names <- paste0("Z_", 1:n)

segments <- tibble(real=Re(z_n), imaginary=Im(z_n), realends=Re(z_ends), imaginaryends=Im(z_en
```

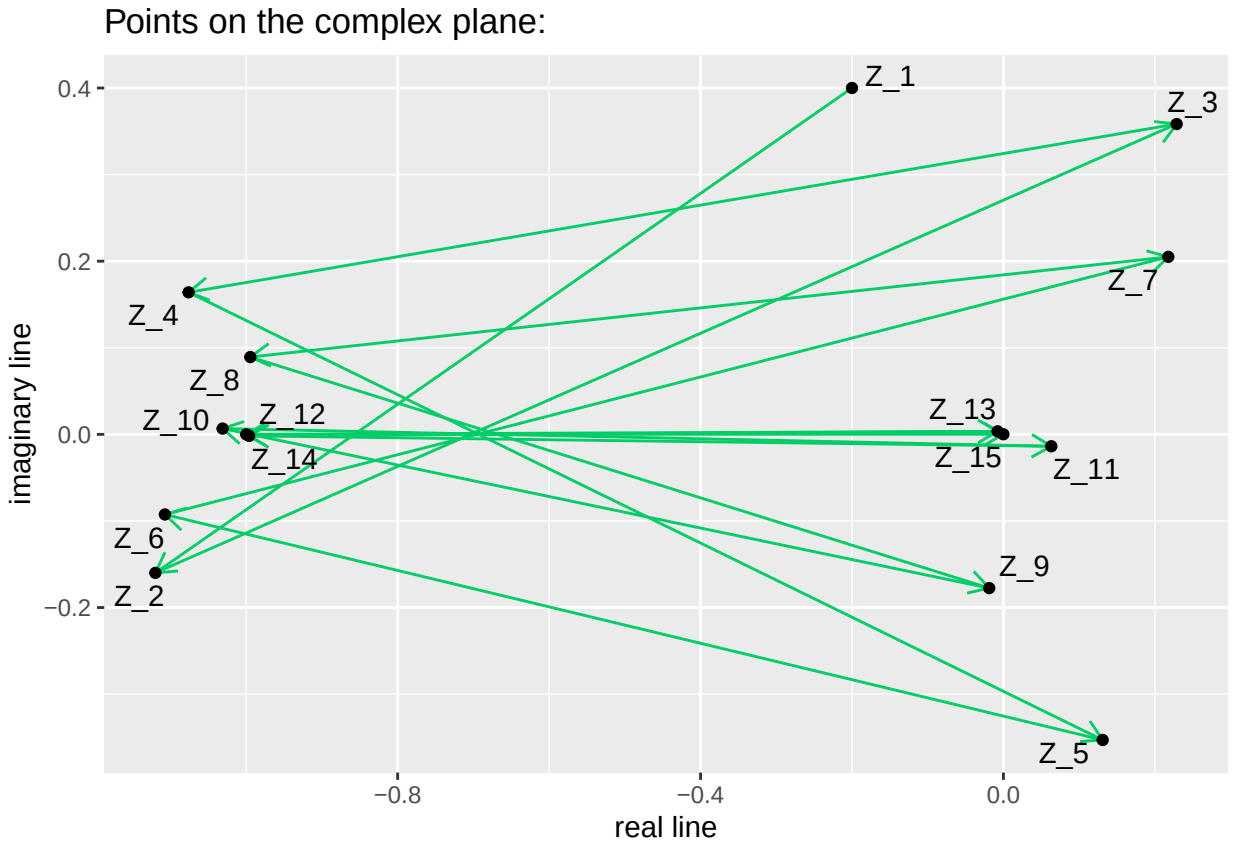
```r
data <- tibble(real=Re(z_n), imaginary=Im(z_n), name=names)

data
```

```
## # A tibble: 15 x 3
##          real  imaginary name
##         <dbl>      <dbl> <chr>
##  1 -0.2        0.4       Z_1
##  2 -1.12      -0.16      Z_2
##  3  0.229      0.358     Z_3
##  4 -1.08       0.164     Z_4
##  5  0.131     -0.353     Z_5
##  6 -1.11      -0.0925    Z_6
##  7  0.218      0.205     Z_7
##  8 -0.995      0.0893    Z_8
##  9 -0.0188    -0.178     Z_9
## 10 -1.03       0.00666   Z_10
## 11  0.0633    -0.0137    Z_11
## 12 -0.996     -0.00174   Z_12
## 13 -0.00762    0.00347   Z_13
## 14 -1.00      -0.0000529 Z_14
## 15 -0.0000922  0.000106  Z_15
```

```r
data %>% ggplot(aes(real, imaginary)) +
  geom_segment(data=segments, aes(x=real, y=imaginary, xend=realends, yend=imaginaryends),
               arrow = arrow(length = unit(0.3, "cm")),
               colour = "springgreen3") +
  geom_point() +
  geom_text_repel(aes(label=name), hjust = .5, vjust = .5) +
  labs(title= "Points on the complex plane:", x="real line", y = "imaginary line") # +
```

```
## Warning: Removed 1 rows containing missing values (geom_segment).
```




```
# coord_equal()
```

Today, I don't have the time to code a a Julia set plotter, but maybe a little further down the line it will appear in my notes.

In the meantime, here's a version that someone else made: *Julia set plotter*