Lec6 – More fractals

- Dimensions ...
- Julia and Mandelbrot fractals
- Event driven programming
- Arrays

Next 2 weeks

- I will be away for the next 2 weeks and Prof. Jen Schwarz will be taking over.

- She is an expert in the physics of `percolation theory` which is closely related to fractals. Teach you some of this.

- The course will function the same way - lectures on Tuesdays, lab on Thursdays with a Thursday homework.

Calculating the dimension of regular fractals

eg **Koch curve:**
Start from line; add triangular bump
then add add bump to all sublines etc
At each stage number of line segments needed to cover goes up by 4
scale distance goes down by factor of 3
Box counting dimension is just:
Thus $d_F = \frac{\log 4}{\log 3}$
Sierpinski similar:
Each iteration needs 3 more triangles to cover object
scale length down by factor of 2 $- d_F = \frac{log 3}{log 2}$

Julia sets

Consider the very simple non-linear map

$$x_{n+1} = x_n^2 - \frac{3}{4}$$

For most starting values of x the final $x = \infty$!
In fact x only remains *bounded* if $|x| < 3/2$
See by drawing graphs of $y = x$ and $y = x^2 - \frac{3}{4}$

- The boundary between the two regions in x (diverging and non-diverging) is called the **Julia set** of the map and contains seemingly 2 points $x = -\frac{3}{2}$ and $x = \frac{3}{2}$

- Things *much* more interesting if we allow ourselves to consider *complex numbers*

Complex numbers

Summary:

- Complex number has 2 parts – real and imaginary

$$z = x + iy$$

- Needed to give answer to question: what is square root of a negative number.

- Add/subtract by adding/subtracting corresponding parts

- Multiply out using usual rules and collect terms *together with* the simple rule $i^2 = -1$

- Magnitude $|z| = \sqrt{(x^2 + y^2)}$

Complex in python

- Python contains an intrinsic complex type `complex`
- eg. `c=complex(0.1,0.2)`
- `c=a+b` etc works transparently
- Also, `conjugate`, `c.real,c.imag` etc
- And `abs(z)`. Coding complex maps trivial.

Maps of complex numbers

Consider previous map for complex numbers

$$z_{n+1} = z_n * z_n - \frac{3}{4}$$

- What is now the region in which $|z|$ diverges under iteration of the map ?

- The region of convergence is called the *filled-in Julia set B* and the boundary between between diverging and non-diverging sets is the *Julia* set of the map.

- Remarkably it is a fractal !!

- The boundary is *rough* on all scales.

Arrays in python

- Similar to lists eg `x[2]` is 2nd element of an array `x`

- Restricted to reals and integers. eg

$$x = array([2, 2], Float)$$

$$x = zeros(5, Int)$$

- Allows one to use powerful fast functions to do array processing eg.

$$x = y + 0.1$$

adds 0.1 to all elements of arrays `x` and `y` which must (obviously) be of the same size and shape.

Other Julia sets

Try general maps $f(z) = z^2 + a$ with

- $a = -0.85 + 0.18i$

- $a = -1.24 + 0.15i$

- $a = -0.16 + 0.74i$

In practice if `abs(z)>2` iterations will always explode.
Try fixed number. Close to boundary need very large number to know .... Useful to zoom ...

Zooming ...

- Julia sets as seen from the complex plane look quite fractal – that is they exhibit lots of structure at small scales. One way to see this more clearly would be examine some portion under high magnification.

- Would be nice to be able to zoom in ....

- This can be done picking out some point and recomputing the Julia set on a small region around that point.

- This point can be selected by clicking with the mouse.

- Requires new event programming techniques.

Event programming

```
#event loop
while(1):
#look for mouse clicks/drags
    if scene.mouse.events:
       my_event=scene.mouse.getevent()
# if left click increase scale factor
       if my_event.press=='left'
          scale_factor=zoom
#get screen coordinates
       target=my_event.pos
       ...
```

More events

- Can also collect keyboard events, textbox events, scrollbar events etc

- Very important in designing graphical user interfaces GUI.

- Allows one to build interactive programs which can be modified while running.

- In this course only used occasionally ... however see the zooming code `juliazoom.py`

More fractals – Mandelbrot

- Can play a new game. Set initial `z=0` and iterate the Julia set map with variable "constant " `a`.

- New famous fractal – the Mandelbrot fractal