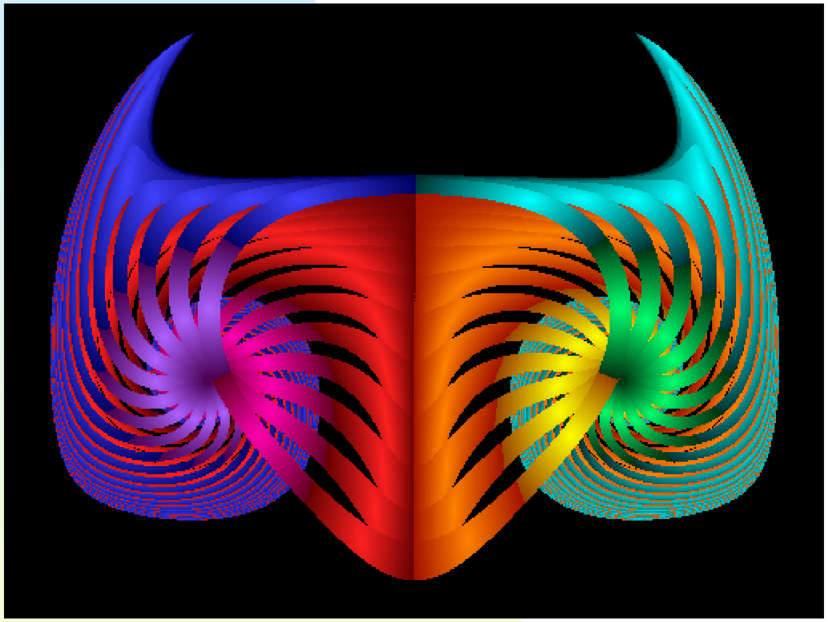
Session overview



- Complex maps and Julia sets
- Reminder: project topics and teams due now on Angel.

Complex maps

- Consider the dynamical system F(z) = z^2, where z is a complex number
- Let's look at the behavior of this system under iteration
  - If $|z_0| < 1$, then the iterates approach 0
  - If $|z_0| > 1$, then the iterates approach ∞
  - If $|z_0| = 1$, then z_0 lies on the unit circle in the complex plane; it lies in the chaotic set
- This chaotic set is called the *Julia set*, after the French mathematician Gaston Julia who first studied it

Example program 1

- Inverse iteration:
  - Apply f'(z)=sqrt(z-c)
    - Choose which square root randomly
  - Generates boundaries

- Program `juliasets.cpp` demonstrates this

Why does this work?

- Points not on the Julia set are repelled to infinity
- Those on it must have preimages in the Julia set.
- Need to invert $u=z^2 + c$.
  - This inverted function is an attractor
  - Just like MRCM

MRCM Example

A problem

- Note the first few points are off the boundary and should really be ignored
- Can we avoid this?
- Can we pick a starting point based on *c* rather than having to input a z_0 and looking at a few extraneous points or putting an unnecessary if statement in the loop?

Repelling fixed points

- Julia showed that repelling fixed points belong to the Julia set
- So, we need to find a repelling fixed point
- Fixed points obey $z = F(z)$, so solve $z = z^2 + c$

Repelling fixed points

- Julia showed that repelling fixed points belong to the Julia set
- So, we need to find a repelling fixed point
- Fixed points obey z = F(z), so solve $z = z^2 + c$

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

Square root of a complex number

- Note that we need the square root of a complex number
- We could convert to polar coordinates like we did earlier, but problems arise when $c = 0$
- So, we implement a square root operation in Cartesian coordinates
- Represent the square root as $x + yi$, square, equate parts, and solve

Round off error

- If the real part of the number whose square root we are taking is large and negative, and imaginary part is small and positive, round off error becomes a major factor
- Need two cases to handle this
- Code is in `juliasets2.cpp`

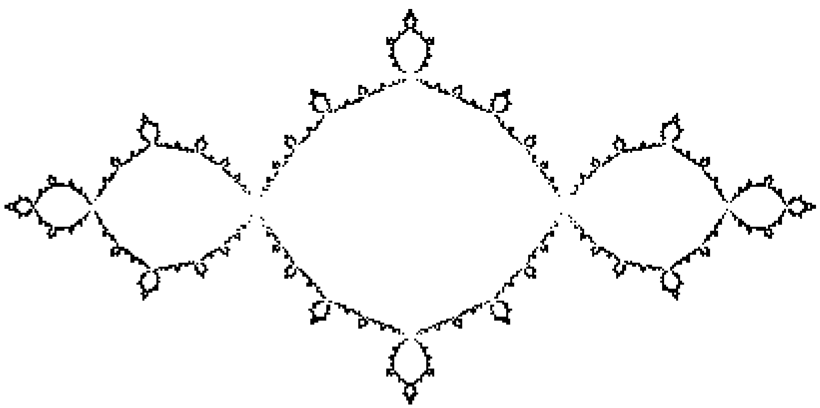
Pick the right one

- We now have the square root of a complex number.
- Recognize there are two of them
- Which one do we start with?
- Need a repelling fixed point
- Recall that a repelling fixed point occurs when the slope of the curve at the fixed point is greater than 1 in magnitude
- So, find $F'(z)$ at z
- This is $2z$, so compute $|2z|$ for each fixed point and take the one that is greater than 1

Example program 2

- Code to do all this is in `juliasets2.cpp`

A boundary



- The Julia set is the boundary of the basin of attraction of a map
- In other words, points within the boundary have bounded orbits while points outside diverge under the map
- The image on the left is the Julia set for *c* = -1

Dynamics of $Q_0(z) = z^2$

- $Q_0^n(z) = z^{2^n}$
- If $|z_0| < 1$, then $z_n \to 0$ as $n \to \infty$
- If $|z_0| > 1$, then $|z_n| \to \infty$ as $n \to \infty$
- If $|z_0| = 1$, then $|z_n| = 1$ for all n
- $|z_0| = 1 \Rightarrow z_0 = e^{i\theta}$ for $0 \leq \theta \leq 2\pi$

Fact 1

- The periodic points of Q_0 are dense on the unit circle, S
- $Q_0^{n}(e^{i\theta}) = e^{i\theta}$ $\Leftrightarrow \left(e^{i\theta}\right)^{2^n} = e^{i\theta}$

  $\Leftrightarrow e^{i2^n\theta} = e^{i\theta}$

  $\Leftrightarrow 2^n\theta = \theta + 2k\pi$

  $\Leftrightarrow \theta(2^n-1) = 2k\pi$

  $\Leftrightarrow \theta = 2k\pi / (2^n-1)$
- For each n, the points given by θ are evenly spaced, so, for n sufficiently large, we can effectively fill up the circle with periodic points

Fact 2

- There is a dense orbit on the unit circle
- This follows from the fact that all of the periodic points are repelling, and yet the orbit is fixed to the circle
- $Q' = 2z \Rightarrow |Q'| = 2 \ \forall \ z$

Fact 3

- $Q_0(z)$ is sensitively dependent to initial conditions
- This follows since any arbitrarily small arc is ultimately mapped over the entire circle, which thus forces two points apart from each other
- So, $Q_0(z) = z^2$ is chaotic on S!

Definitions of Julia sets

- The ***filled Julia set*** of F(z) is the set of all points $z_0 \in \mathbf{C}$ whose orbits are bounded (don't diverge)
- The boundary of the filled Julia set is called the ***Julia set*** of F(z)

Examples

- The filled Julia set for $Q_0(z) = z^2$ is the unit disk, $|z| \leq 1$
- The Julia set for $Q_0(z) = z^2$ is the unit circle, $|z| = 1$

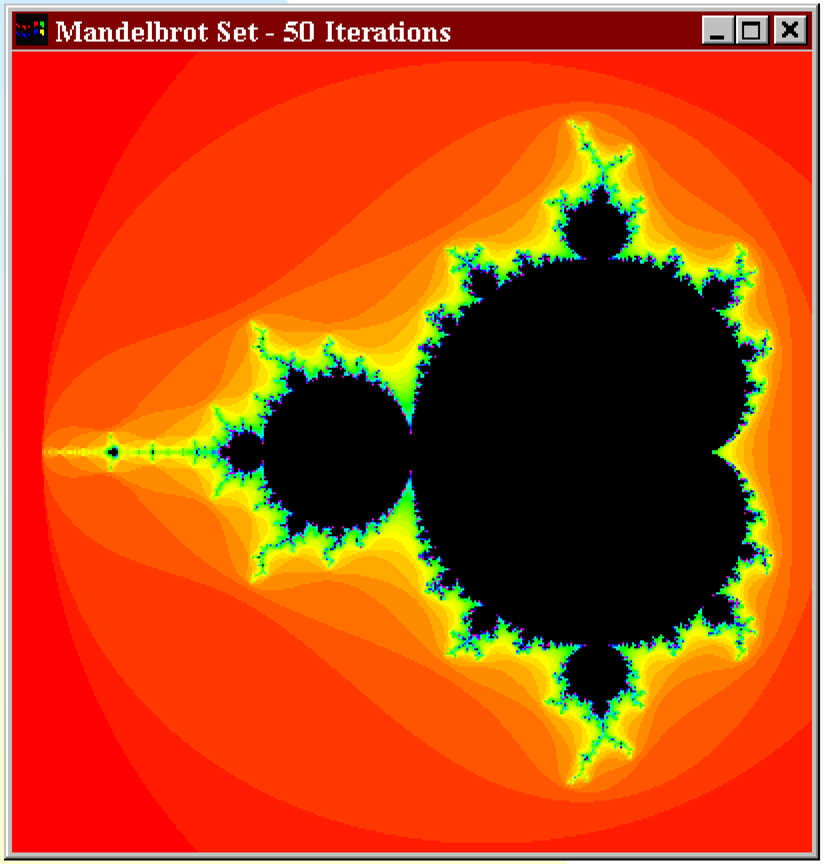
The escape criterion

- Theorem: Suppose $|z_0| \geq |c| > 2$. Then $|z_n| = |Q_c(z_0)| \to \infty$ as $n \to \infty$.
- Proof: $|Q_c(z_0)| = |z_0^2 + c| \geq |z_0|^2 - |c|$ (triangle inequality) $\geq |z_0|^2 - |z_0| = |z_0|(|z_0| - 1) > (\lambda + 1)|z_0|, \lambda > 0 \Rightarrow |Q_c^n(z_0)| > (\lambda + 1)^n|z_0| \Rightarrow |z_n| = |Q_c^n(z_0)| \to \infty$ as $n \to \infty$

Corollary

- If $|c| > 2$, then $Q_c^n(0) \to \infty$
- Proof: $|Q_c(0)| = |c| > 2$. Now apply the previous theorem with $z_0 = c$.

The Mandelbrot set



- Definition: The Mandelbrot set is defined by $\mathbf{M} = \{ c \in \mathbf{C}: |Q_c(0)|$ does not approach $\infty \}$

Coloring the Mandelbrot and Julia sets

- Points inside the boundary are colored black
- Points outside the boundary are colored to indicate rate of escape
- Want colors spread out (so they’re visible)
- Do not want to constantly change the color map
- So, develop one color map and scale based on number of iterations
- Routines in `MandelbrotExplore.c` show this

Observe pictures of Julia set for $c = -1$

- $-2.0 \le x \le 2.0$, $-2.0 \le y \le 2.0$, 25 iterations
- $-2.0 \le x \le 2.0$, $-2.0 \le y \le 2.0$, 100 iterations
- $-2.0 \le x \le 2.0$, $-2.0 \le y \le 2.0$, 1000 iterations
- $-0.75 \le x \le 0.75$, $-0.75 \le y \le 0.75$, 25 iterations
- $-0.75 \le x \le 0.75$, $-0.75 \le y \le 0.75$, 100 iterations
- $-0.75 \le x \le -0.25$, $-0.5 \le y \le 0.5$, 25 iterations
- $-0.4 \le x \le -0.3$, $0.3 \le y \le 0.5$, 25 iterations
- $-0.34 \le x \le -0.32$, $0.35 \le y \le 0.4$, 25 iterations
- $-0.34 \le x \le -0.32$, $0.35 \le y \le 0.4$, 100 iterations
- $-0.34 \le x \le -0.32$, $0.35 \le y \le 0.4$, 500 iterations

Class tomorrow

- Please bring laptops