MAT 128B: Project I: Using iteration methods to understand fractal geometry

Wednesday, Feb. 14th

To get full credit show all your work and explain in your words the results you obtain

Names:
[100 pts]

In this project your team will implement computer programs that use iteration methods to generate fractal figures on the plane. You will learn some basic concepts in fractal geometry through the project.

This is a team project and requires collaboration. Describe in detail how your team is organized and how the tasks are being addressed. I strongly recommend you to use Github for this collaboration. If you do so please provide evidence.

Filled Julia sets, Julia sets and Mandelbrot sets

- **i.- An introduction to fractals** Pages 98–101 in Greenbaum and Chartier (attached) give an introduction to fractals. Some key concepts that you will need to understand are: orbit, Filled Julia sets, Julia set and Mandelbrot set (discuss these concepts with your team). An important result states that if the $orb(0)$ is bounded then the Julia set is connected (that is any two points in the set can be joined by a line). Implement the program given in the background reading to generate the Filled Julia set in Figure 4.13 and to convince yourself that the Filled Julia set of $\phi(z) = z^2$ is the unit disk (denoted by D^2).

- **ii.- Generate (and plot) other examples changing the value of c in the function $\phi(z) = z^2 + c$.** Try examples like $c = 0.36 + 0.1i$ or $c = -0.123 - 0.745i$. Describe what happens when $|z| > 2$ or when the initial z_0 values (chosen in the program provided) are changed.

- **iii.- Constructing the Julia set**. Remember the Julia set is the boundary of the Filled Julia set. We here are going to give an algorithm to generate the Julia set. It is called the Inverse Iteration method: Given a complex number $z = x + iy = r(\cos\theta + i\sin\theta)$ with $r = \sqrt{x^2 + y^2}$ and $\theta = tan^{-1}\frac{y}{x}$ when $x > 0$ (adding π if $x < 0$) and $\sqrt{z} = \sqrt{r\cos\frac{\theta}{2}} + i\sqrt{r\sin\frac{\theta}{2}}$. Use these expressions to develop an iteration method whose function is $\psi = \pm\sqrt{w - c}$. For each number you can randomly pick the positive or negative value of the expression for ψ. Verify your program using the values you used before.

- **iv.- Computing the Fractal dimension** The paper Bisoi and Mishra 2001 (see references below) gives several algorithms to compute the fractal dimension. Explain with your own words what the fractal dimension is and implement one of those algorithms to compute the fractal dimension of the Julia sets generated above. Check that your algorithm is correct by computing, for instance, the dimension of the boundary of the disk generated in i.

- **v.- Connectivity of the Julia set** Write a program that computes $orb(0)$ and determines whether a Julia set is connected or not. Assume divergence occurs when $|z| > 100$

- **vi.- Coloring divergent orbits** Write a program that assigns a color to the divergent orbits in the Julia set. The coloring is assigned according to the time the orbits take to diverge (i.e. $|z| > 100$). Two orbits, one that gets $|z| \approx 100$ at iteration n and another at iteration $n+1$ have similar colors.

- **vii.- Newton's method in the complex plane** Using the algorithms developed above. Can you write a program that uses the Newton method on the complex plane (as an iteration method)? Looking at the colored orbits. Can you identify the roots of the polynomial? Choose examples of the form $\phi(z) = z^n - 1$ for a fixed n.

- **viii.- The Mandelbrot set** Generate the Mandelbrot set associated with $\phi(z) = z^2+c$. This is going to be achieved by changing c. Color the values of c according to the number of iterations that it took for the orbit of 0 to diverge. This was the cover of the journal Scientific American in 1985.

References

- Bisoi AK and Mishra J. 2001 On calculation of fractal dimension of images *Pattern Recognition Letters* Volume 22, Issues 6–7, Pages 631-637

- Devaney RL and Keen L 1988 Chaos an fractals. The mathematics behind the computer graphics. American Mathematical Society.