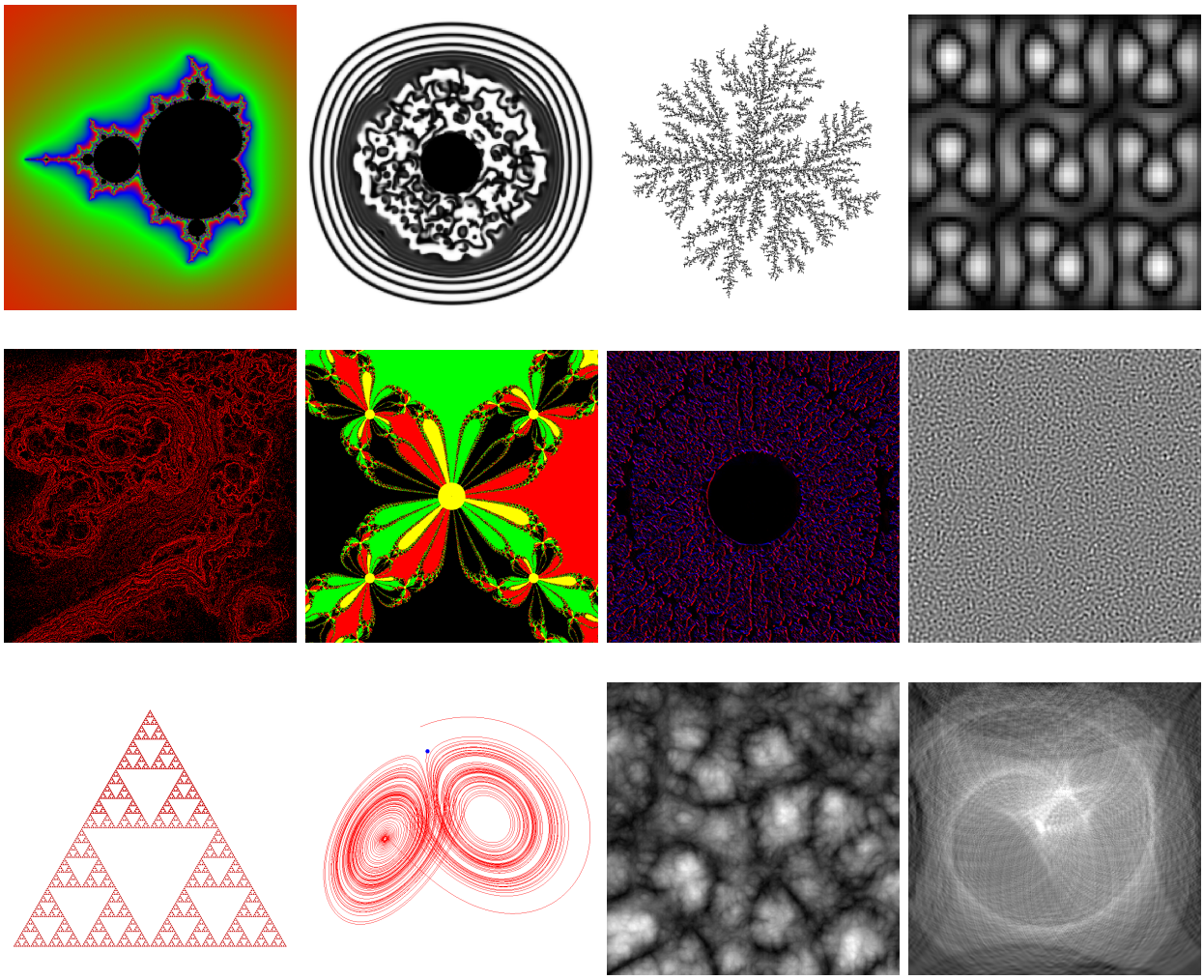
MAT 200C: Digital Media Technology and Engineering

Media Arts and Technology Program, Spring 2012
University of California, Santa Barbara



Course Description

This offering of MAT 200C will focus on mechanisms of pattern formation. Many complex visual patterns can be generated from surprisingly small amounts of code (< 100 lines), or from simple sets of equations. Some examples include the ubiquitous patterns of the Mandelbrot and Julia set fractals, reaction-diffusion systems, Conway's Game of Life, and the Lorenz attractor. We will examine a wide variety of these pattern generation techniques. We will implement these techniques in class, and homework assignments will focus on experimenting with and generating novel patterns. By the end of the class, you will have developed a suite of pattern generation tools.

Tentative Course Topics

- The Mandelbrot and Julia set fractals
- Cellular automata and Conway's Game of Life
- Diffusion limited aggregation (DLA) and Laplacian growth
- Recursive subdivision: Sierpinski's Triangle, Levy's Dragon, the Koch Snowflake
- Isotropic diffusion and Perona-Malik anisotropic diffusion
- Reaction-diffusion systems: Turing, Gray-Scott, Meinhardt, FitzHugh-Nagumo, and Barkley
- The Ising spin lattice: more is different
- Newton-Raphson iteration: the fractal structure of polynomial root finding
- The Fourier transform: the convolution theorem, the slice theorem, and band-limited noise
- Eigenvalues of the Laplacian: can you hear the shape of a drum?
- Spectral differentiation: diffusion using the FFT
- Random sampling, Voronoi diagrams, Delaunay triagulation
- The Lorenz attractor
- The Reynolds bird flocking algorithm
- Procedural texturing: clouds and mountains
- Morphological operators

Instructor:

Theodore Kim, kim@mat.ucsb.edu

Lecture:

Tuesdays and Thursdays, 2-4 PM

Prerequisites:

It will be assumed that you have a basic level of programming experience, and you must have a laptop that you can bring to class. You will be provided with a significant amount of basic substrate code so that you can focus on the task of writing pattern formation code. The substrate code will be written in C++ and designed to be run on a Mac, but it should be generic enough to run on Linux and Cygwin with minor modifications. You are free to implement using any language or platform that you want, with the understanding that we will primarily support C++ on the Mac.

Grading Scheme:

- **In-class exercises, 25%:** The first half of each class will be a lecture on some pattern formation algorithm. The second half of each class will be a lab session during which we will implement the actual algorithm. You will be expected to have made substantial progress on your implementation by the end of lab.
- **Experimentation, 25%:** In between classes, you will be asked to experiment with the pattern formation mechanism developed during lab, and asked to post the most interesting structure you generate to a shared forum.
- **Final Project 50%:** You will have the opportunity to further explore one of the pattern formation mechanisms from class, or a related topic of your own choosing, as a class project.

Textbooks:

There is no required textbook. When appropriate, we will read research papers.