

Nonlinear Physics:
Modeling Chaos and Complexity
Spring 2008
Syllabus

Instructor: Prof. Jim Crutchfield (chaos@cse.ucdavis.edu; <http://cse.ucdavis.edu/~chaos>)

Assistant: Benny Brown (brown@cse.ucdavis.edu)

Time: 3:10 - 4:30 AM TuTh

Location: 158 Roesler Hall (Tu) and 2118 Mathematical Sciences (Th)

WWW: <http://cse.ucdavis.edu/~chaos/courses/nlp/>

Parallel Theme I: Forms of Randomness, Order, and Intrinsic Instability

1. Qualitative Dynamics
2. Continuous-time ODEs and discrete-time maps
3. Bifurcations
4. Stability, Instability, and Chaos
5. Quantifying (In)Stability

Parallel Theme II: Tools for Exploring Chaos and Complexity

1. Modeling methods
2. Graphics
3. Simulation
4. Interaction
5. Programming

Prerequisites:

- Interest in modeling some dynamical phenomenon
- Vector calculus
- Linear algebra
- Lower division Math, Physics, or CS courses
- Programming: C/C++, Java, or Python (We will use Python.)
- Laptop with Python version 2.5 running

Readings:

- *NDAC* textbook: *Nonlinear Dynamics and Chaos*, Strogatz, 2001 Printing
- *Python* textbook: *Learning Python*, Lutz & Ascher, Third Edition, 2008
- Course Lecture Notes (available via course website).

Contents

1 Qualitative Dynamics

2

2	Bifurcations	4
3	Visualizing and Quantifying Unpredictability	5
4	Finish Projects and Present	7

First Lecture (1 April, Tuesday): Overview

Readings:

- *Chaos*, JP Crutchfield, JD Farmer, NH Packard, RS Shaw, Scientific American **255** (1986) 46–57.
- *Odds*, Stanislaw Lem, New Yorker **54** (1978) 38–54.

Topics:

1. Introduction and motivations
2. Three parts: Dynamics, Bifurcations, Chaos
3. Survey interests, background, and abilities
4. Course logistics
5. Exams and projects
6. Software and program development

Homework Week 0: Everyday unpredictability; see website. Write-up due one week from Thursday, but be prepared to discuss at next Tuesday meeting.

Programming Lab 1 (3 April, Thursday): Python and Its Environment

Reading: *Python* Part I (Chapters 1-5).

Topics:

1. Modeling: Simulation, interaction, and graphics programming
2. Python language (Ch. 1)
3. Python and scientific computing packages installed and running (Ch. 2)
4. Developing and running Python using iPython (Ch. 3)
5. Python as a calculator (WWW)

1 Qualitative Dynamics

Theory Lecture 1 (8 April, Tuesday): The Big Picture

Reading: *NDAC*, Chapters 1 and 2.

Topics:

1. Pendulum demo
2. Discuss *Chaos* and *Odds* readings and homework
3. Qualitative Dynamics: A geometric view of behavior
4. State space
5. Flows
6. Attractors
7. Basins
8. Submanifolds
9. Concrete, but simple example: One-dimensional flows

Programming Lab 2 (10 April, Thursday): Python, the Language

Reading: *Python* Part II (Chapters 6-9).

Topics:

1. Objects and Assignments (Ch. 8 & 9)
2. Text files (Chs. 6 & 7)

Homework: Collect Week 0's, assign Week 1's.

Theory Lecture 2 (15 April): Example Dynamical Systems

Reading: *NDAC*, Sections 6.0-6.7, 7.0-7.3, and 9.0-9.4.

Topics: Continuous-time ODEs

1. 2D Flows: Fixed points (Sec. 6.0-6.4)
2. 2D Flows: Limit cycles (Sec. 7.0-7.3)
3. 3D Flows: Chaos in Lorenz (Sec. 9.0-9.4)
4. Simulation demo
5. From continuous to discrete time (Sec. 9.4)
 - (a) Poincaré Maps and Sections
 - (b) Lorenz ODE to cusp map
 - (c) Rössler ODE to logistic map (pp. 376–379)
 - (d) Discrete-time maps
6. Dimensional classification of behavior
7. Example dynamical systems

Programming Lab 3 (17 April): Arrays, Functions, and Modularity

Reading: *Python* Part II (12, 13, and 15) & course website.

Topics:

1. Arrays and Functions (Chs. 12 & 13)
2. Modules (Ch. 15)
3. Command line control

Homework: Collect Week 1's, assign Week 2's.

2 Bifurcations

Theory Lecture 3 (22 April): The Big, Big Picture (Catastrophes)

Reading: *NDAC*, Chapter 3.

Topics:

1. Qualitative Dynamics: Space of all dynamical systems
2. Example: Bifurcations of one-dimensional flows
 - (a) Saddle Node
 - (b) Transcritical
 - (c) Pitchfork
3. Catastrophes: Fixed point to fixed point bifurcation
4. Example: Cusp Catastrophe
5. Catastrophe theory classification of fixed point bifurcations

Programming Lab 4 (24 April): Higher Python & Graphical Python

Reading: *WWW* & *Python* Chapters 19-21 and 24-26.

Topics:

1. Statistics (*WWW*)
2. Linear Algebra (*WWW*)
3. Plotting (*WWW*)
4. Data Types
5. OOP (Ch. 19)
6. Classes (Chs. 20 & 21) & Attributes
7. Error Handling (Chs. 24-26).
8. Visual Python 3D graphics

Homework: Collect Week 2's, assign this week's (3).

Theory Lecture 4 (29 April): The Big, Big Picture (Bifurcations)

Reading: *NDAC*, Chapter 8 and Sec. 10.0-10.4.

Topics:

1. Logistic map
2. Fixed point to limit cycle
3. Phenomenon and calculation
4. Limit cycle to limit cycle
5. Phenomenon and calculation
6. Routes to chaos: Period-doubling cascade
7. Phenomenon and calculation
8. Band-merging
9. Periodic windows and intermittency
10. Simulation demo
11. Bifurcations in ODEs:
 - (a) Hopf bifurcation
 - (b) Limit cycle to torus
 - (c) Torus to chaos
 - (d) Chaos to chaos

Programming Lab 5 (1 May): Simulation Methods, Fixed Points and their Stability

Reading: *WWW & Python* Chapter 28.

Topics:

1. Graphical Users Interfaces (*WWW & Ch. 28*)
2. Euler method for ODEs
3. Runge-Kutta method for ODEs
4. Discrete-time maps
5. 1D: 1D state v. time
6. 2D phase portrait and 2D v. time
7. Stable & unstable manifolds of fixed points
8. Bifurcations

Homework: Collect Week 3's, assign this week's (4).

Project: Pick project. Write up project proposal.

3 Visualizing and Quantifying Unpredictability

Theory Lecture 5 (6 May): Mechanism of Chaos

Reading: *NDAC*, Sec. 12.0-12.3, 9.3, and 10.5.

Topics:

1. Chaotic mechanisms: Stretch and fold
2. Baker's map
3. Cat map (and stretch demo)
4. Henon map: stretch-fold and self-similarity
5. Roessler attractor branched manifold

Programming Lab 6 (8 May): Limit Cycles

Topics:

1. 2D phase portrait and 2D v. time
2. Stable & unstable manifolds of fixed points
3. Bifurcations: Hopf bifurcation
4. Limit cycle to torus

Homework: Collect Week 4's, assign this week's (5).

Project: Project should be chosen and designed.

Theory Lecture 6 (13 May): Quantifying Chaos

Reading: *NDAC*, Sec. 12.0-12.3, 9.3, and 10.5.

Topics:

1. Dot spreading: Roessler and Lorenz ODEs
2. Lyapunov characteristic exponents (LCEs)
3. Time to unpredictability
4. Dissipation rate
5. Attractor LCE classification
6. Chaos defined

Programming Lab 7 (15 May): Chaos

Topics:

1. ODE Examples: Lorenz, Roessler, Pendulum, and more

2. Map Examples: Baker's map, Henon map, Standard Map, and more
3. Exponential separation: Dot-spreading
4. Fractal attractors: Dissipative Baker's map and Henon map
5. Multiple basins of attraction: Simple and fractal separatrices

Homework: Collect Week 5's, assign this week's (6).

Theory Lecture 7 (20 May): Analyzing Chaotic Maps & Routes to Chaos

Reading: *NDAC*, Chapter 10.

Topics:

1. Shift Map
2. LCEs for Maps
3. Tent Map
4. Logistic Map
5. LCE view of period-doubling route to chaos
6. Period-doubling self-similarity
7. Renormalization group analysis of scaling

Programming Lab 8 (22 May): Measuring Chaos

Topics:

1. Lyapunov Characteristic Exponents
2. For 1D Maps
3. For 2D Maps
4. For 3D Flows

Homework: Collect Week 6's, assign this week's (7).

Programming Lab 9 (27 May): Immersive Visualization

Tour of KeckCAVES sensory immersive environment: keckcaves.org.

4 Finish Projects and Present

1. Class, 29 May: Projects; collect Week 7's homework.
2. Class, 3 June: Projects
3. Class, 5 June: Projects

Note: Project write-ups due at the end of the last week of classes, which is Friday 6 June.