AMS 332/BIO 332: Computational Modeling of Physiological Systems

Spring 2017, Mon-Wed, 4–5.20 PM, HEAVY ENG. 201 WESTCAMPUS

Instructors: David F. Green (david.green@stonybrook.edu), Applied Mathematics & Statistics
Giancarlo La Camera (giancarlo.lacamera@stonybrook.edu), Neurobiology & Behavior

Synopsis: This course will introduce students to the fundamental principles underlying computational modeling of complex physiological systems. A major focus of the course will be on the process by which a model of a biological system is developed. Students will be introduced to the mathematical methods required for the modeling of complex systems (including stochastic processes and both temporal and spatial dynamics) as well as to tools for computational simulation. Roughly one half of the course will focus on models for general cellular physiology, while the remaining half will focus on the neurobiological systems underlying learning and behavior. *NOTE*: AMS/BIO-332 is one of 3 mandatory advanced courses for the *Specialization in Quantitative Biology and Bioinformatics* (QBB), together with AMS 333 ("Mathematical Biology") and BIO 312 ("Bioinformatics and Computational Biology").

Course Objectives: The main objectives of AMS/BIO332 are (1) To introduce mathematical concepts encountered in biology and neuroscience research and more advanced graduate courses. (2) To teach basic programming skills in MATLAB. (3) To teach how to think about biology and neuroscience in a quantitative way. One of the main points of the course is to communicate that simple quantitative models can be very powerful. A more detailed list of learning objectives is available below, following the course syllabus.

Prerequisites: MAT 127 or MAT 132 or a comparable course that provides exposure to elements of Calculus and Linear Algebra, and any one of the following: BIO 202 or BIO 203 or CHE 132 or CHE 331 or PHY 127 or PHY 132, or permission of instructor.

Textbook: There is no official textbook for this course. Extensive sets of lecture notes will be made available to the students. As *background or reserve materials* on Neuroscience, the students can consult the following books (but they are *not* required to buy them):

- P. Dayan and L. Abbott, Theoretical Neuroscience, MIT Press (for Computational Neuroscience)
- J. Nicholls et al, From Neuron to brain, 5th Ed, Sinauer (for General Neuroscience).

Exams and Grading Policy: 1 midterm and 1 final exam. In addition, weekly quizzes (10 min at the beginning of class one day a week) and projects based on Matlab each due roughly every three weeks (see below). *Grading policy:* 10% weekly quizzes; 20% midterm exam; 20% final exam; 50% projects (either 5 projects worth 10% each, or 4 projects worth 12.5% each, at the discretion of the instructors).

Matlab Projects: The course has no formal lab component and prior experience with Matlab is not required. A Matlab primer written by one of the instructors, guiding students through some of the basics of the program, will be provided, and demonstrations of how to use Matlab will be given during some of the lectures. The students are encouraged to cooperate on the projects; the submitted project reports are required to include a statement explaining the nature and extent of collaboration.

Attendance and Make Up Policy: Late work will not be accepted. Attendance at all exams is mandatory. In the case of 1) verifiable illness, 2) verifiable family emergency, 3) University-sanctioned religious holiday, or 4) participation in official University-sponsored events (for documented student athletes only), excuse must be documented on official letterhead (as appropriate) and will be verified by the instructor.

Class protocol: Please try to be punctual and arrive in class within 5 minutes of the beginning of class. Students who will be later than 15 minutes may be asked not to enter class. In the case of conflict with other classes please contact the instructor. All electronic devices are to be turned off during class unless advance permission is given by the instructor.

Class resources: Blackboard (http://blackboard.stonybrook.edu) will be used as the primary means of distribution for readings from the primary literature and/or lecture notes and/or submission of assignments. It is your responsibility to consult regularly the Blackboard listing for this course.

Email and Communications policy: Email and especially email sent via Blackboard (http://blackboard.stonybrook.edu) is one of the ways we will officially communicate with you for this course. It is your responsibility to make sure that you read your email in your official University email account. For most students that is Google Apps for Education (http://www.stonybrook.edu/mycloud) but you may verify your official Electronic Post Office (EPO) address at:
http://it.stonybrook.edu/help/kb/checking-or-changing-your-mail-forwarding-address-in-the-epo
If you choose to forward your official University email to another off campus account, we are not responsible for any undeliverable messages to your alternative personal accounts. You can set up email forwarding using these DoIT-provided instructions found at:
http://it.stonybrook.edu/help/kb/setting-up-mail-forwarding-in-google-mail
If you need technical assistance, please contact Client Support at
(631) 632-9800 or supportteam@stonybrook.edu

***Special Needs* (Americans with Disabilities Act):** If you have a physical, psychological, medical or learning disability that may impact your course work, please contact Disability Support Services, ECC (Educational Communications Center) Building, room128, (631) 632-6748, or online at http://studentaffairs.stonybrook.edu/dss/. They will determine with you what accommodations, if any, are necessary and appropriate. All information and documentation is confidential.
Students who require assistance during emergency evacuation are encouraged to discuss their needs with their instructors and Disability Support Services. For procedures and information go to the following website: http://www.sunysb.edu/ehs/fire/disabilities.shtml

***Academic Integrity*:** Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Faculty are required to report any suspected instances of academic dishonesty to the Academic Judiciary. Faculty in the Health Sciences Center (School of Health Technology & Management, Nursing, Social Welfare, Dental Medicine) and School of Medicine are required to follow their school-specific procedures. For more comprehensive information on academic integrity, including categories of academic dishonesty, please refer to the academic judiciary website at http://www.stonybrook.edu/uaa/academicjudiciary/

***Critical Incident Management*:** Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Judicial Affairs any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn. Faculty in the HSC Schools and the School of Medicine are required to follow their school-specific procedures.

Copyright Notice: Any material prepared for this course, including, but not limited to, this syllabus, the lecture notes, Matlab projects, exercises, quizzes and exams are intended for the sole use of the students enrolled in the course and are not for public distribution. No one is allowed to upload on the World Wide Web, or redistribute in any form and by any means, any of this material without written consent from the instructors, penalty the infringement of copyright law.

Course Syllabus

28 sessions of lectures of 80 minutes each, including review and in-class exams

***NOTE*:** Some of the schedule and/or content of the lectures may change during the course. The instructors will communicate changes (if any) during class and/or via Blackboard.

PART I: MODELING CELLULAR SYSTEMS

1. **1/23**: Introduction to modeling physiological systems
   - Opportunities and challenges of an inter-disciplinary approach
   - Time and length scales in biological systems
2. **1/25**: ODE-based biochemical models I
   - Introduction to biochemical (mass-action) kinetics
   - Systems of ordinary differential equations (ODEs)
3. **1/30**: ODE-based biochemical models II
   - Phase-plane analysis of ODEs; velocity fields, null clines and stationary points
   - The phage-λ lysis/lysogeny decision pathway
4. **2/1**: Stochastic biochemical models I
   - Noise in biological systems; the challenges of small numbers
   - Probability distributions
5. **2/6**: Stochastic biochemical models II
   - Stochastic processes
   - Markov processes and Markov chains
   - Stochastic theory of reaction kinetics
6. **2/8**: Stochastic biochemical models III
   - Chemical master equation
   - The Gillespie algorithm
7. **2/13**: Stochastic biochemical models IV
   - A stochastic analysis of the phage-λ system
   - Stochastic behavior in eukaryotic gene regulation
8. **2/15**: PDE-based biochemical models I
   - Spatial variation in cellular systems
   - Sub-cellular structure in prokaryotes and eukaryotic cells
9. **2/22**: PDE-based biochemical models II
   - Introduction to partial differential equations (PDEs)
   - Diffusion processes; Fick's law
   - Reaction-diffusion systems
10. **2/22**: PDE-based biochemical models III
    - Reaction diffusion in eukaryotic signal transduction
    - Spatial waves of Ca^{2+} ion concentration
11. **2/27**: PDE-based biochemical models IV
    - Stochastic behavior in spatially and temporally varying systems
    - Brownian motion

12. **3/1**: Towards a complete model of the cell
    - Computational challenges and solutions
    - The vexing problem of cellular motion
13. **3/6**: Modeling cellular physiological systems: summary and review
14. **3/8**: Midterm exam

***** March 13-17: SPRING BREAK *****

PART II: MODELING NEUROBIOLOGICAL SYSTEMS

15. **3/20**: Electrical properties of neurons I
    - Introduction to Neurobiology
    - Neurons and synapses
    - Membrane potential, ion channels, ion pumps, reversal potentials
    - Equivalent circuit of one-compartment neuron model
16. **3/22**: Electrical properties of neurons II
    - Kinetic theory of voltage-sensitive (Na^+ and K^+) channels
    - Introduction to the Hodgkin-Huxley model
17. **3/27**: The Hodgkin-Huxley model and the action potential
    - The Hodgkin-Huxley model
    - The action potential
18. **3/29**: Phase-plane analysis of excitable membranes
    - A 2D model of action potential generation (Morris-Lecar model)
    - Phase-plane analysis of the Morris-Lecar model
    - Hopf and saddle-node bifurcations
19. **4/3**: One-dimensional (1D) models of spiking neurons
    - 1D models of spiking neurons: quadratic and leaky integrate-and-fire model
    - The firing rate of the leaky integrate-and-fire neuron
20. **4/5**: Modeling synaptic input
    - Synapses, receptors and synaptic conductances
    - Post-synaptic current and post-synaptic potential
    - Response of the leaky integrate-and-fire neuron to synaptic input
21. **4/10**: Stochastic models of neural activity
    - Cortical spike trains
    - Poisson model of spike trains
    - Response to Poisson spike trains: membrane potential as a stochastic process
    - Diffusion approximation for the membrane potential
    - Sigmoidal response function of the leaky integrate-and-fire neuron
22. **4/12**: Introduction to neural coding
    - Receptive fields and tuning curves of cortical neurons
    - Encoding and decoding
    - Decoding as probabilistic inference: the example of perceptual discrimination
    - Population coding of movement direction

23. **4/17**: Networks of neurons

- Introduction to networks of neurons
- Feedforward and recurrent networks
- Dynamics and stability in recurrent networks
- Memories as persistent states of cortical activity

24. **4/19**: Plasticity and learning I

- Synaptic plasticity: experiments and models
- Unsupervised and Hebbian learning
- Stability properties of Hebbian learning
- The BCM rule

25. **4/24**: Plasticity and learning II

- Learning and classification
- Learning in feed-forward networks: the 'perceptron'
- The method of gradient descent
- Back-propagation: learning with 'hidden' layers

26. **4/26**: Plasticity and learning III

- Introduction to reinforcement learning
- Learning from evaluative feedback
- Reinforcement learning as reward maximization
- Example: stochastic perceptron

27. **5/1**: Plasticity and learning IV

- Reinforcement learning with spiking neurons
- The 'escape noise' model
- Gradient learning of spike trains
- Learning strategies: summary and final remarks

28. **5/3**: Modeling neurobiological systems: summary and review

Learning objectives

General learning goals: upon completion of AMS/BIO 332, the students will be able to

1. Think about biology and neuroscience in a quantitative way.
2. Interpret and use mathematical concepts often encountered in biology and neuroscience research and more advanced graduate courses.
3. Build simple mathematical models of biological systems.
4. Write Matlab computer code to study, explore and analyze their models.
5. Apply/Extend the learned principles of numerical and analytical computation to other computational subjects and/or programming languages.

Specific learning objectives: upon completion of AMS/BIO 332, student will be able to

1. Build a mathematical model of a gene regulatory network based on mass-action kinetics and ordinary differential equations (Lectures 1-2).
2. Interpret the dynamics of a biochemical system using phase-plane analysis (Lecture 3).
3. Explain the difference between a deterministic and stochastic view of biochemical kinetics (Lectures 4-7).
4. Outline Gillespie's algorithm for simulating a stochastic biochemical reaction pathway (Lectures 6-7).
5. Provide an interpretation based on mathematical modeling of how lysis versus lysogeny is determined in bacterial infections (Lectures 3 and 7).
6. Explain the difference between models based on ordinary differential equations and partial differential equations, and to explain when each is appropriate (Lectures 8-12).
7. Describe the reaction-diffusion model (Lectures 9-10).
8. Discuss how a reaction-diffusion model can explain pattern formation in animal skins (Lecture 10).
9. Implement the simulation of a biochemical network in the Matlab programming environment, perform quantitative analyses and present the results graphically (Matlab projects).
10. Describe what makes neurons produce action potentials – in particular, what minimal "ingredients" are required and how this can be formalized in models of reduced complexity and dimensionality (Lectures 15-19)
11. Build a model of synaptic inputs, choose suitable receptor types, and determine how neurons respond to their synaptic inputs under typical cortical conditions (Lectures 20-21)
12. Describe the main concepts and the leading candidate mechanisms involved in neural coding of visual perceptions and planned movements (Lecture 22)
13. Describe the concept of networks of neurons, explain the differences among some canonical network structures, and use one particular structure to provide a model of memory based on the collective behavior of populations of spiking neurons (Lecture 23)
14. Explain the difference in three main learning strategies: unsupervised, supervised and reinforcement learning, and describe some of the influential models of each of these types of learning (Lectures 24-27). More specifically:
15. Explain the principles of Hebbian learning and describe some of the cortical plasticity phenomena providing experimental evidence for it (Lecture 24)
16. Describe the 'perceptron' device and explain the concept of supervised, error-correcting learning of classification tasks (Lecture 25)
17. Explain the principles behind reward-based learning and build a model of reinforcement learning with spiking neurons (Lectures 26-27)
18. Implement the simulation of single neuron and neural networks dynamics in the Matlab programming environment, code simple but widely used tools for the analysis of neural data, perform quantitative analyses and present the results graphically (Matlab projects).