Dynamical modeling in biology: Stochastic models
BI 410/510, University of Oregon

- Syllabus -

Stilianos Louca

Spring 2023

Contents

1 Course description 2

2 Instructors 2

3 Schedule 2

4 Zoom instructions 3

5 Target audience 3

6 Expected learning outcomes 3

7 Prerequisites 4

8 Materials and equipment 5

9 Weekly reading assignments 6

10 Weekly homework exercises 6

11 Estimated student workload 6

12 Grading 7

13 Course policies 7

14 Disabilities and other issues 9

Note: This syllabus accounts for the current regulations and restrictions associated with the COVID-19 pandemic, as of March 20, 2023. Some aspects may be subject to change on short notice as the COVID-19 situation continues to evolve.

1 Course description

This course covers stochastic dynamical models in biology, i.e., mathematical models that describe the behavior of non-deterministic biological systems over time as a result of internal feedback loops, external forcings and random processes. Such models have become a popular alternative to deterministic models throughout the biological sciences. Topics covered will include stochastic iterative maps, discrete-time Markov chains, continuous-time Markov chains, vector autoregression models, and concepts of time series analysis. Examples will cover a broad spectrum of topics in biology, such as neuroscience, microbiology, population dynamics, genetics, molecular evolution and ecology. As the concepts covered are rather general and fundamental, students from other quantitative fields such as physics, chemistry or data sciences will likely also find great use in them.

The course will consist of two in-person lectures per week, one computer lab, weekly reading assignments, and weekly homework exercises. The lectures will cover mostly theoretical/mathematical material, as well as general approaches in scientific programming. The homework exercises will include mathematical questions as well as scientific programming tasks, where students will practice using a programming language to solve modeling questions. The computer lab will provide opportunities to discuss the weekly assignments, to cover additional grounds on the programming side, and to help students resolve programming questions for example related to their assignments. In addition, the GE and the lead instructor will each host 1 office hour per week.

This course is assigned 4 credits.

2 Instructors

Lead instructor: Stilianos Louca, PhD, Assistant Professor.
Email address can be found at www.loucalab.com.
May also be contacted through Canvas, but **not** through Microsoft Teams.
In-person office hours: Fridays at 16:00–17:00, in Onyx Bridge 282.

Teaching assistant: Ananya Kapoor
May be contacted via Canvas.
In-person office hours: Mondays at 15:30–17:30, in McKenzie Hall Room 471.

3 Schedule

In-person lectures: Tuesdays and Thursdays, 16:00–17:20, in Lawrence (LA) 222.
First lecture is on Tuesday, April 4, 2023. Last lecture is on Thursday, June 8, 2023.

In-person computer lab: Fridays, 14:00–15:50, in Esslinger Lab room (ESSL 116).
First lab is on Friday, April 7, 2023. Last lab is on Friday, June 9, 2023.

Mid-term exam: In-person written examination during the Friday computer lab, on

May 12, 2023 at 14:00–15:50, in Esslinger Lab room (ESSL 116).

Final exam: In-person written examination during exam week. Date & time to be determined.

Holidays observed: None.

4 Zoom instructions

Unless mentioned otherwise, all lectures, computer labs, exams and office hours will be held in person. However, as the COVID-19 pandemic continues to evolve, situations might arise where any of these sessions is moved to a remote Zoom meeting (for example, if the lead instructor is forced to self-isolate). Hence, students are encouraged to familiarize themselves with Zoom prior to the course. Instructions for installing Zoom and joining a Zoom meeting can be found at:
https://service.uoregon.edu/TDClient/2030/Portal/KB/ArticleDet?ID=101392

Please join all online lectures and computer labs with video turned on and audio turned off by default. If you'd like to ask a question during the lecture and lab, please behave as in a normal class – either raise your hand and/or discretely draw the instructor's attention by voice (remember, everyone in the meeting can hear you). Please avoid using the Zoom text-chat function during the lecture and lab, since it distracts from the main conversation and may escape the attention of the instructor.

5 Target audience

The course targets senior undergraduate and junior graduate students in biology and related fields, with a strong interest in mathematical and computational approaches, but could also be of interest to students in physics, chemistry, mathematics and environmental sciences. Indeed, students from other quantitative disciplines will find that many of the mathematical techniques taught are also applicable to problems in their own field.

6 Expected learning outcomes

Students will learn to construct stochastic dynamical mathematical models of realistic complexity (i.e., as encountered in realistic applications and in the scientific literature) and use these to analyze realistic datasets using computer programming (including simulations, model fitting and model interpretation). Students will also improve their logical reasoning skills, including conducting simple mathematical deductions (aka. "proofs") in the context of scientific modeling.

The following topics will be covered, roughly in this order:

- Stochastic processes, power spectral density, autocovariance function.
- Analyzing time series (e.g., electroencephalograms) as samples of a stochastic process.
- Stochastic iterative maps (e.g., for population modeling) - general concepts
- Vector Autoregression (VAR) models, simulation, forecasting, fitting models to data,

autocovariance function, power spectral density.

- Discrete-time Markov chains, stationary distributions, irreducibility, simulations and fitting models to data (e.g., for DNA evolution).
- Continuous-time Markov chains, stationary distributions, irreducibility, simulations and fitting models to data (e.g., for gene expression dynamics).

Simple mathematical proofs as well as scientific programming are common components of the course. Here "proof" simply means that a rigorous mathematical justification is given for a particular statement. This has a two-fold educational rationale. First, seeing the *reason* for a specific mathematical statement facilitates its apprehension, as the task shifts from that of *memorization* to one of *understanding*. Second, a central aspect of modeling is to make quantitative predictions through deductive reasoning that can then be tested experimentally. When an experiment contradicts the prediction of a model it is essential to know whether the experiment really contradicts the model, or whether the model itself is accurate but the prediction was erroneously deducted through sloppy (aka. "handwavy") arguments. Being accustomed to the art of mathematical proof (that is, clear deductive reasoning) is a skill that any serious modeler must have.

Application of the material taught will be demonstrated using examples from across the field of biology, including neuroscience, microbiology, population dynamics, genetics, molecular evolution and ecology.

All programming examples will be taught in MATLAB, which is a widely used programming environment in the sciences and in engineering, however concepts and methods will be taught in a sufficiently general manner as to be easily applicable using alternative programming languages.

Learning outcomes will be evaluated through homework assignments and exams (details in Section 12), as well as through discussion in class.

7 Prerequisites

Recommended prerequisites for this course are:

- Familiarity with probability theory and statistics (multivariate distributions, conditional probabilities, independent variables, correlations, linear regression).
- Familiarity with multivariate calculus (e.g., partial derivatives).
- Familiarity with differential equations and basic matrix algebra.
- Basic experience with computer programming (any language).
- Basic knowledge of biology (e.g., high-school level).

Students lacking some of this background can still succeed in this course with additional effort. Optional supporting materials covering many of the above prerequisites will be provided to all students at the beginning of the course. Students lacking most of this background will probably find this course to be challenging. Prerequisite courses listed on the university website may be omitted upon instructor approval.

8 Materials and equipment

Canvas: Unless otherwise mentioned, material (reading assignments, homework assignments, announcements etc) will be posted on the UO Canvas course page. Students should check for announcements and assignments posted on Canvas at least every 2-3 days; students may also choose to set up automated email notifications in Canvas.

Textbook: The full reading material for this course will be provided for free as a textbook on Canvas. Students should download the latest version of this document from Canvas at least once a week, since material might be added or modified throughout the course. You must not share this textbook outside of the course! Students lacking some of the prerequisites mentioned earlier will need to look into catching up on that material prior to the course, e.g. through wikipedia, books etc., although some brief introductions are also provided as appendixes in the textbook.

Software: The course includes examples and homework assignments in MATLAB, a widely used commercial software for scientific computations. The provided code examples assume MATLAB 2019b or later. Macs at the Price Science Commons & Research Library have MATLAB installed, and students can use those to work on their homework assignments depending on current UO restrictions due to COVID-19. Students will also have access to computers with a MATLAB installation during the weekly computer labs, however note that the lab time will almost certainly not be enough to finish the homework assignments. Students are thus strongly encouraged to install MATLAB on their own computers. It is also recommended to bring a small USB thumb drive to the labs for backing up your work at the end of each session (the lab computers have USB-A ports). Students are free to use their own laptops during the weekly computer labs, instead of those provided in the lab. The University of Oregon provides MATLAB licenses for students at: software.uoregon.edu. During installation, make sure to include the "Optimization", "Econometrics" and "Curve fitting" toolboxes (other toolboxes might also be useful if you have space).

Students may in principle choose to use an alternative software to MATLAB for their homework assignments (e.g., R or python), however this is discouraged and no advice/instructions will be provided specific to that alternative software. There is also no guarantee that the provided lab computers will have any of those other programming languages installed. In past years many students that chose to use an alternative programming language tended to have more difficulties with the homework than their peers, likely because all course examples were given in MATLAB. Students should not be discouraged by the fact that they may currently be less familiar with MATLAB than, say, R or python. The programming tasks for this course will be sufficiently generic that MATLAB solutions can easily be applied in other languages such as R or python — the algorithmic/mathematical reasoning remains essentially the same across languages.

Copyright and privacy protection: Do not share or post online any of the course material, including the textbook, lecture notes, assignments and official solutions, exams, or recordings of online Zoom meetings.

9 Weekly reading assignments

Students will be expected to complete weekly reading assignments from the freely available textbook mentioned in Section 8. These assignments will be posted on the course's Canvas webpage prior to the week in which they are supposed to be completed. Reading assignments will largely cover the material discussed in class during the same week. By having reading assignments that overlap with the lecture material, students will be better prepared to ask questions and participate in discussions during the lectures. Reading assignments for graduate students will sometimes include additional material (such as mathematical proofs) compared to undergraduates, to satisfy the higher expected workload/credit ratio.

10 Weekly homework exercises

Students will be expected to complete weekly homework exercises, which will generally include mathematical calculations or simple proofs as well as scientific programming tasks (such as simulating a model and visualizing the outcomes). Exercises will be based on previously covered material and will be graded. Weekly exercise assignments will be posted on Canvas each Thursday, and will be **due on the following Wednesday at 23:59**. Solutions must be uploaded to Canvas in PDF file format prior to the deadline. Only PDF files will be accepted; please do not upload your assignments in Word document format. Solutions in paper format will not be accepted. Students are encouraged to learn and use LaTeX for their assignments, a free and widely used software for writing scientific (especially mathematical) documents. An alternative is to use Jupyter notebooks. Students may also complete their assignments on paper and upload scanned PDFs to Canvas, provided that these are clearly legible and combined into a single PDF (although students are encouraged to type their solutions on a computer).

Graduate students will generally need to complete more assignments than undergraduates, to satisfy the higher workload/credit ratio expected for graduate students. Students are expected to complete most of the assignments on their own. Consultation and collaboration with peers is permitted, but every student should fully understand the solutions produced and must individually write up their computer code and report. Assignment solutions are graded individually. Solutions must be clearly legible and must adhere to common professional standards (e.g., proper axes labeling, thorough reasoning, proper writing, reasonable structure, legibility, etc). Solutions to selected exercises will be subsequently discussed in class during the weekly computer session. Computer code itself will generally not be graded nor evaluated - only the results of the computer code will be evaluated.

11 Estimated student workload

In addition to attending the lectures and computer labs, students will be expected to complete weekly reading assignments from a textbook (details in Section 8) and weekly exercise assignments (details in Section 10). A summary of the typical workload needed to succeed in this course is given below:

Lectures: 20 lectures × 1.5 hours = 30 hours

Computer lab (including discussions): 10 labs × 2 hours = 20 hours

Assignments (undergraduates): 10 weeks × 7 hours per week = 70 hours

Assignments (graduate students): 10 weeks × 11 hours per week = 110 hours

Total hours (undergraduates): 120

Total hours (graduate students): 160

12 Grading

All homework and exam scores are counted on a percentage scale. The final grade G will be calculated on a percentage scale using the following formula: $G = 0.4 \cdot H + 0.6 \cdot E$, where H is the average score from the homework assignments (averaged over all assignments) and E is determined by the mid-term and final examination scores, as follows. If the final exam score is greater than the mid-term exam score, then E is equal to the final exam score. If the final exam score is equal to or less than the mid-term exam score, then E is equal to a weighted average of the two exam scores, with the mid-term exam weighting 30% and the final exam weighting 70%. The motivation for this grading scheme is that it gives an opportunity to the student to fully make up for a bad mid-term score at the finals, thus encouraging effort to improve even if a bad score is achieved in the mid-term. There will be no opportunities for extra credit, no grade bumps and no grading on a curve.

Letter grade: The following rubric will be used to convert the final grade G from a percentage scale to a letter scale.

Percentage (P)	Letter
$97\% \leq P \leq 100\%$	A+
$93\% \leq P < 97\%$	A
$90\% \leq P < 93\%$	A-
$87\% \leq P < 90\%$	B+
$83\% \leq P < 87\%$	B
$80\% \leq P < 83\%$	B-
$77\% \leq P < 80\%$	C+
$73\% \leq P < 77\%$	C
$70\% \leq P < 73\%$	C-
$67\% \leq P < 70\%$	D+
$63\% \leq P < 67\%$	D
$60\% \leq P < 63\%$	D-
$0\% \leq P < 60\%$	F

13 Course policies

Attendance: Students are strongly encouraged to attend all lectures and computer labs. However, attendance will not be taken into account for the grade. It is ultimately the student's responsibility to stay up to date with the material taught and homework assignments. Students that miss a class will be able to download the covered course materials from Canvas, as part of the freely provided textbook. Assignment solutions and all important announcements will also be posted on Canvas.

Late homework submissions: Late delivery of solutions to homework exercises will generally not be approved, with the exception of disability-related accommodations established in coordination with the Accessible Education Center (AEC). Non-approved late homework submissions incur a score penalty of 10%. In any case, homework submitted more than 24 hours past the deadline will not be accepted regardless of reason, including disability-related reasons; this is because homework solutions will be shared with the class shortly after their deadline to facilitate students' timely learning. Note that due to UO senate-mandated equitable policy on course attendance and engagement, the instructor is not allowed to make special exceptions, other than for disability-related reasons (via the AEC). The aforementioned late-submission penalty (10% if the delay is no more than 24 hours) is mild enough so that if an unexpected event does cause a brief delay in the student's homework submission, the cost incurred in terms of grade is small. Further, students are encouraged to start working on their homework early on and not postpone it until the last minute, to avoid risking missing the deadline due to unexpected circumstances. Students with an AEC-approved flexibility in assignment deadlines due to a disability should keep in mind the following:

- Students must only request a deadline extension if there is an acute, unpredictable, disability-related event that coincides with the deadline, even if a general flexibility with deadlines was previously approved by AEC.

- Deadline extensions will still be limited to 24 hours, for reasons stated above.

- Students should request a deadline extension in writing (via email), and only assume that an extension has been granted if this was confirmed by the instructor in writing.

- Students must CC their AEC advisor every time they request a deadline extension.

- The instructor reserves the right to first reach out to AEC (e.g., the advisor that has been CC'd on the student's email) with any concerns, and only grant the extension after hearing back from the AEC advisor that the extension is justified, and only if it does not remove essential elements of the assignment or interfere with the course's objectives, on a case by case basis.

Sharing materials: Students may share their personal lecture notes and may discuss their homework assignments with each other. However, students **must not** share exams, exam solutions, or solutions to the weekly exercise assignments online (including through Canvas). Students must not share audio or video recordings of the lectures, computer labs or office hours without prior permission from the head instructor.

Arriving on time: Please arrive in class on time. Late arrivals distract the instructor and the other students. Please turn off or silence your cell phones during classes. Ask questions if you did not hear or understand something. Students may use laptops or tablets during class (e.g., to take notes), but must do so in a way that does not distract other class participants (e.g., don't display distractive videos or play user interface sounds).

Academic integrity: Students are expected to adhere to the University's guidelines on academic integrity as outlined in the University of Oregon Student Conduct Code: https://policies.uoregon.edu/vol-3-administration-student-affairs/ch-1-conduct/student-conduct-code

14 Disabilities and other issues

If there are aspects of this course that result in unfair barriers to your participation, please let us know as early as possible. You may also contact Accessible Education Services in 164 Oregon Hall, by phone at 541-346-1155 or via email at uoaec@uoregon.edu.

Students may request to borrow a Chromebook laptop from the University of Oregon, as described here:
https://service.uoregon.edu/TDClient/2030/Portal/Requests/ServiceDet?ID=42589

Students may also access the UO virtual computer lab, which provides various Windows software, including MATLAB, remotely:
https://service.uoregon.edu/TDClient/2030/Portal/Requests/ServiceDet?ID=42572

Students having trouble installing MATLAB on their computer should contact CASIT as soon as possible (https://casit.uoregon.edu).