BMES 678: Computational Methods in Biosimulation[1]

Instructor: Ah-met Saçan <ahmet.sacan@drexel.edu> (Office: Bossone702)
Preferred course contact email of instructor and TAs: bmes678staff[at]umich.edu

Course Description:
This course focuses on computational methods used to simulate and analyse dynamical systems in biological systems. Solutions of ordinary differential equations (ODEs) using both symbolic and numerical methods and parameter estimation from experimental data are discussed. 3D modelling and simulation are introduced. Graphical tools to design and simulate models are demonstrated.

Course Learning Objectives:
1. Learn to design and simulate biomedical dynamical systems using differential equations and graphical modeling tools.
2. Learn methods for numerical and symbolic solutions of ordinary differential equations (ODEs).
3. Learn methods for estimating model parameters from experimental data.
4. Be introduced to 3D modeling and simulation methods.

Course Materials:
- Advanced Engineering Mathematics, 10th Edition, Erwin Kreyszig, 2011, Wiley. ISBN 978-0-470-45836-5. Available online at: http://instructor.sdu.edu.kz/~merey/Advanced%20Engineering%20Mathematics%2010th%20Edition.pdf
- Dynamical Systems with Applications Using Matlab, Stephen Lynch, Springer, ISBN 9780817643218, 2004. Available online at: https://archive.org/details/springer_10.1007-978-0-8176-8156-2

Course Outline:

Week1. Analytical solutions to dynamic systems, symbolic math
Assignment: Collection of calculus problems, differential equations, Irreversible Receptor-Ligand binding kinetics

Week2. Numerical solutions . First Order ODEs, Numerical Solutions, Taylor Series, Euler's Method.
Assignment: Lotka-Volterra predator-prey model, Euler's method

Week3. Numerics for first-order ODEs. Euler's method and Runge-Kutta 4th method, Matlab odeXX functions.
Assignment: Approximating sin(x) and simulating Lotka-Volterra predator-prey model using Improved Euler's Method and Runge-Kutta Method.

Week4. Applications of numerical ODE solutions.

[1] The syllabus is subject to change by the instructor. Any changes will be announced in class and/or on the course discussion forum.

Assignment: Glycolytic Oscillations, Three-node repressive network as a biochemical oscillator.

Week5. Stability, Bistability
Assignment: Fish Population Dynamics with Holling's Type II Harvesting

Week6. Simbiology, Graphical programming of dynamical systems.
Assignment: Fish Population Dynamics with Holling's Type II Harvesting
Assignment: Lactose transport & LacY expression

Week7. Modeling & Simulation using Simulink.
Assignment: Mechanical Model of Muscle Spindle

Week8. Parameter Estimation
Assignment: Glycolytic Oscillations

Week9. In-class final exam.
Advanced Topics: 3D Modeling, Finite Element Method, and Molecular Dynamics

Week10. Student project presentations
Example projects: Lactose Operon, DNA damage & repair, Virus dynamics & therapy, HIV infection, Biological Clock, Cell migration.

Grading:

- ***Blackboard Quizzes*****:** 20%
  - *Blackboard Quizzes are multiple choice, short answer, True/False, or matching type questions. On Blackboard, these are indicated with "- quiz" in the title of a test.*
- ***Programming Assignments*****:** 40%
  - *These are weekly or biweekly programming assignments that require you to upload/submit your program files to Blackboard. On Blackboard, these are indicated with "- programming" in the title of an assignment.*
- ***Final Exam*****:** 20%
  - *Final Exam will be held synchronously (all students need to take the final exam during the scheduled time).*
- ***Projects*****:** 20%
  - *Project topics are pre-assigned or selected by students. Each project team will contain 3 students (adjustments to the group memberships may be made by instructor). There are milestones you need to complete for the project; these milestones are graded assignments. Your overall project grade is the sum of all project-related assignments.*

Blackboard Quiz Questions:
Blackboard quizzes are intended to reinforce the weekly material. The questions may be in multiple-choice, short-answer, and term-matching format. An example question is given below:

Solve the initial value problem y'=3y, y(0)=5. What is the value of y(0.1) ? Provide your final answer rounded to 4 digits after the decimal point.

Programming Assignment Questions:
Each week, you will be working on solving small modelling and simulation problems that require writing computer programs.

- Unless otherwise noted, you may complete the programming assignments in Matlab, Python, or R. If you would like to use other programming languages (Java), you must obtain permission from the instructor.
- You must work (from start to finish) within a Dropbox folder dedicated to your programming assignments. You are asked to share that Dropbox folder with the instructor (see instructions on Blackboard).
- Your programs are graded for correctness as well as programming style.
- Your solution must only answer the assigned questions and should not have any extraneous code or output. A common source of grading deductions is using instructor-provided demonstrations that do not directly address the assignment questions.

Expected Competencies: Students are expected to know at least one of Matlab, Python, or R programming languages and be prepared to complete the programming assignments in any of these programming languages. Students must know the following basic programming concepts: selection statements, loops, functions, structs, cell arrays, vectorized code (ie., performing a mathematical operation on each element of a vector), logical indexing, data import, basic statistics (finding average or standard deviation), and creating plots. Lecture and instruction material may be given in any programming language and the students may need to translate code from one programming language to another.
An example programming problem you are expected to be able to solve (in any programming language of your choice) is given below. Check how much time it takes you to independently solve that example problem. You can expect to spend 5 to 10 times as much time on the weekly programming assignments in this course.

Programming Competency Check:
Write a function xls_columnaverage(file, columnname) that takes as input arguments: the name of an Excel file and the name of a column of interest. Assume that the first row of the Excel file contains the field names (describing what each column contains), and the remaining rows contain numerical data. Your function should load the data contained in the Excel file and calculate the average of the column whose name is contained in columnname. The column average should be returned from the function. Your function should work when it is called with any Exccel file and column name, assuming the Excel file satisfies the constraints described above.

To test your function, create two example Excel files called sample1.xlsx and sample2.xlsx. Add mock patient information to these Excel files. Make sample2 columns be in different order than sample1. Include columns representing name, age, height, and weight. Add several rows of example patients to each of these files.

Test your function (in Matlab command window – or adapt these test case codes to your programming language) with the following test cases:

```
>> xls_columnaverage('sample1.xlsx','age')

>>  xls_columnaverage('sample2.xlsx','weight')
```

Double-check the answers from your function and make sure they match with what you get from manual calculations of the data.

Weekly Plan[2]:

- by **Monday** morning: Topic material, quizzes, and programming assignments become available. There may be modifications to the programming assignment during the lecture time.
- by **Wednesday** night: Complete reviewing the lecture material (read the weekly material and watch the lecture videos). Complete any pre-lecture preparatory quizzes.
- **Thursday** lecture: I will present the weekly topic and review the programming assignments.
- **Friday** morning: Lecture recording becomes available (email the instructor if it does not appear on Blackboard).
- **Saturday** night: Blackboard quizzes due. **No late submission accepted for Blackboard quizzes.** Your grade for the Blackboard quizzes as well as the correct answers will be available after the deadline.
- **Sunday** night: Weekly programming assignments due. **0% late penalty for the first day late. Additional 10% per day for each additional day late. Late penalties are not pro-rated.**
- **Office Hours:** weekly days and times to be announced later. Office hours will be held in-person and/or on Zoom.

Late Submission Policy:

- **For Blackboard quizzes/tests (composed of multiple choice and short answer questions), late submission is not allowed.** You must submit them before the deadline.
- **For programming assignments:** there will be 10% per day penalty for each day late. Late penalties are **not** pro-rated to the time of day.
- Individual assignments may have their own specific late penalty policy.
- Late penalties cannot be waived, unless arranged with the instructor prior to the due date or in the case of an emergency.
- Late submissions are not accepted after the last day of classes.

Due dates:

[2] There may be changes to the quiz and assignment deadlines due to late course registrations, changes in weekly progress, and holidays. In such cases, the deadline listed on Blackboard for the quiz or assignment will be the effective one.

Typical due dates are provided in the Weekly Plan above. Some assignments and in-class exercises may have different due dates. See Blackboard for specific due dates.

Class Attendance:
Attendance is not graded. Students registered to the in-person course section are expected to attend class in-person. Students registered to the online section are strongly recommended to follow the lecture session synchronously via Zoom. A recording of the lecture session will become available in the morning following the lecture session; online section students may follow the course asynchronously through these recordings. In-person and online students will have the same deadlines for the assignments.

Grading Scale:

Score %	**<64**	**64**	**67**	**70**	**74**	**77**	**80**	**84**	**87**	**90**	**94**	**97**
Letter Grade	**F**	**D**	**D+**	**C-**	**C**	**C+**	**B-**	**B**	**B+**	**A-**	**A**	**A+**

The grading scale is subject to alterations depending on student performance in the class.

Use of External Resources:

- We expect you to utilize our office hours to get help on the assignments. If you are using an external tutor (either an online service or an in-person tutor), you must document the tutoring activity in your submitted work.
- Searching or using solutions to the assignments that may be available on various "homework help" sites (e.g., Chegg) is prohibited.
- Your use of other online knowledgebases (such as documentation and existing forum messages), and automated response tools (e.g., ChatGPT) must serve the learning goals of the course. For example, you may use these resources to learn more about the programming concepts, but you may not use an existing solution to an assignment.
    - Example of **acceptable** of use ChatGPT: Asking ChatGPT to explain how Matlab's indexing works by asking: "Can you explain indexing in Matlab to me?"
    - Example of **acceptable** of use ChatGPT: Entering your code that has problems into ChatGPT and asking it to find mistakes or improvements. You need to use the responses to enhance your understanding of the material.
    - Example of **un-acceptable** of use ChatGPT: Entering an assignment text asking it to solve it for you.
- If you use an external resource for constructing or improving your work on an assignment, you must document your use in your submitted work. E.g., if you use ChatGPT, save the history of questions and answers for an assignment in the folder for that assignment as an html, docx, or pdf file. If you use StackOverflow (or other FAQ websites), include the URL as comments in your code and/or describe your use in a docx file.

Additional Course policies:

- All submissions must be the submitting student's own work. For assignments where the use of external source code is allowed, references should be provided to such source code.
- All coursework must be performed within a Dropbox folder shared with the instructor.

- You are responsible for following the discussion forum and checking your Drexel email account for any announcements on a daily basis.
- All items in this syllabus are subject to change as the course progresses. You will be notified in class and/or via email of any changes in policies or content.

- All students are expected to abide by Drexel University's policies. If an act of academic dishonesty is determined to have occurred, for a first offense, one of the following sanctions will be imposed, depending on the severity of the offense:
    - Reduction of a course grade
    - An "F" for the assignment or exam
    - Failure for the entire course with the inability to withdraw.
    - Other action deemed appropriate by the faculty member. Examples include, but are not limited to, requiring the student to re-take the exam, re-complete an assignment, or complete an assigned exercise. The decision of the faculty member and the department head shall be reported to the Office of Judicial Affairs, which is responsible for maintaining student conduct records. The incident will result in an official disciplinary record for the student(s).
- Any academic honesty infraction beyond a first offense is subject to the sanctions described above, as well as to disciplinary sanctions that may be imposed through the University judicial process, administered through the Division for Student Life and Administrative Services/Office of Judicial Affairs. These sanctions may include suspension or expulsion from the University.

Appropriate Use of Course Materials:
It is important to recognize that some or all of the course materials provided to you may be the intellectual property of Drexel University, the course instructor, or others. Use of this intellectual property is governed by Drexel University policies, including the IT-1 policy found here: https://drexel.edu/it/about/policies/policies/01-Acceptable-Use/
Briefly, this policy states that course materials (including recordings and assignments), provided by the course instructor may not be copied, reproduced, distributed or re-posted. Doing so may be considered a breach of this policy and will be investigated and addressed as possible academic dishonesty, among other potential violations. Improper use of such materials may also constitute a violation of the University's Code of Conduct found here: https://drexel.edu/cpo/policies/cpo-1/ and will be investigated as such.
Please refer to the following links regarding important University policies.

Academic Integrity:
http://www.drexel.edu/provost/policies/academic_dishonesty.asp
http://www.drexel.edu/studentlife/judicial/honesty.html

Disability Accommodation:
http://www.drexel.edu/oed/disabilityResources/

Adding a Course:
http://www.drexel.edu/provost/policies/course_add.asp

Dropping a Course:
http://www.drexel.edu/provost/policies/course_drop.asp

Withdrawing from a Course:
http://www.drexel.edu/provost/policies/course_withdrawal_policy.asp