University of Montana
ScholarWorks at University of Montana

University of Montana Course Syllabi | Open Educational Resources (OER)

Spring 2-1-2017

CSCI 477.01: Computer Simulation and Modeling

Jesse Johnson
University of Montana - Missoula, jesse.johnson@umontana.edu

Follow this and additional works at: https://scholarworks.umt.edu/syllabi
Let us know how access to this document benefits you.

Recommended Citation
Johnson, Jesse, "CSCI 477.01: Computer Simulation and Modeling" (2017). *University of Montana Course Syllabi*. 4748.
https://scholarworks.umt.edu/syllabi/4748

This Syllabus is brought to you for free and open access by the Open Educational Resources (OER) at ScholarWorks at University of Montana. It has been accepted for inclusion in University of Montana Course Syllabi by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.

Computer Simulation and Modeling
CSCI 477 and 577
Spring 2017 Syllabus

> The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work – that is, correctly to describe phenomena from a reasonably wide area.
>
> –John von Neumann

Instructor Details

Name:	Jesse Johnson
Office:	406A Interdisciplinary Science Building
Telephone:	(406) 243-2356
Email:	`jesse.johnson@umontana.edu`
Web:	http://hs.umt.edu/hs/faculty-list/faculty-details.php?id=540
Office Hours:	T 15:00–16:00 and Th 10:00-12:00 , Interdisciplinary Science Building 406A *Or, by appointment.*

Prerequisites

Students taking this course are expected to have:

- familiarity with computer file systems, software installation, and prior experience programming in a structured language,
- understanding of symbolic differentiation and integration (calculus I and II) and some exposure to linear algebra,
- college level science courses (Physics, Chemistry, Geology, Biology),
- intellectual maturity and a willingness to work.

Course Objectives

After taking this course, students will be able to:

- write numerically efficient Python programs of intermediate complexity,
- construct algorithms that capture the physical characteristics of a system,
- solve both ordinary and partial differential equations with computers,
- verify and validate computer simulations,
- manipulate the fundamental data structures of numerical computing; matrices and vectors, and
- represent and understand datasets graphically.

Textbook

This semester I'll be using the following text book. You'll need to purchase a copy at the UM Bookstore, or online.

Computational Physics by *Mark Newman* (2013)

Supplemental Text

Some students appreciate a concise, printed reference for Python programming. There are many good online tutorials, but this is a good, inexpensive ($9.10) reference.

Python Pocket Reference by *Mark Lutz*, 2014 O'Reilly Media

Online Resources

Please bookmark the following online resources immediately:

- with the exception of the textbook, all course material will be made available online, through the University of Montana's Moodle system,
- the textbook has a web site, and
- there is a course wiki.

Software

We will be using Python, as well as a number of numerical libraries and some programming tools. The easiest way to obtain all of what you need is by installing Anaconda Python. If you are using Linux and prefer to install individual packages, begin with `numpy`, `ipython`, `scipy`, `matplotlib`, and `jupyter`.

Course Format

This course has the following pedagogical features to help you master what is challenging material for most:

- in-class lecture and problem solving sessions. The lectures will be interspersed with problem solving sessions to provide "*just-in-time*" delivery of information. Rapid development of a basis for understanding will be achieved through in-class group programming.
- Most problem solving will be done with Jupyter notebooks. These notebooks allow me to motivate the problem and provide hints. They also allow me to break down larger problems into a series of steps.
- Lectures, problems, and projects will mostly be drawn from the text "*Computational Physics*", by Mark Newman.
- The course has a wiki, that will be added to by students and the instructor.
- We will be using a source code repository for our work.

Meeting Times/Place

Times: Monday, Wednesday 16:30–17:20
Place: Social Science 362

Final Exam Time and Place

15:20–17:20 Monday, May 9, 2017
Social Science 362

Grading Policy

Grading scale

A	94-100
A-	90-93
B+	87-89
B	83-86
B-	80-82
C+	77-79
C	73-76
C-	70-82
D+	67-69
D	63-76
D-	60-62
F	0-59

Students taking the course pass/no pass are required to earn a grade of D or better in order to pass.

477 Assessments and weights

The following assessments will be used and weighted according to the values in the table to determine final grades for 477 students.

Component	Description	Number	Weight
In-class problems	Problems worked on in the classroom, by the instructor and groups of students. $\frac{1}{3}$ of the grade will be attendance, $\frac{1}{3}$ assessment from classmates, and $\frac{1}{3}$ correctness of solution. All group members may submit the same work for these assignments.	12	30%
Homework	Assessment of individual student performance on the assigned problems. These are to be worked by students outside of the classroom. Students are encouraged to discuss solutions with their groups, but each submitted assignment must represent the student's own work and be unique.	6	30 %
Projects	In depth application of programming techniques to questions arising in the sciences. These are also to be completed individually.	2	20%
Exams	Tests of your knowledge of basic programming and applications to science.	2	20%

577 Assessments and weights

The following assessments will be used and weighted according to the values in the table to determine final grades for 577 students.

Component	Description	Number	Weight
In-class problems	Problems worked on in the classroom, by the instructor and groups of students. $\frac{1}{3}$ of the grade will be attendance, $\frac{1}{3}$ assessment from classmates, and $\frac{1}{3}$ correctness of solution. All group members may submit the same work for these assignments.	12	20%
Homework	Assessment of individual student performance on the assigned problems. These are to be worked by students outside of the classroom. Students are encouraged to discuss solutions with their groups, but each submitted assignment must represent the student's own work and be unique.	6	20 %
Projects	In depth application of programming techniques to questions arising in the sciences. These are also to be completed individually.	2	15%
Graduate Project	An advanced project taken directly from modern scientific literature. Undertaken by the graduate student and tracked through a series of milestones detailed below.	1	30%
Exams	Tests of your knowledge of basic programming and applications to science.	2	15%

577 Milestones for Graduate Project

Past experience has shown that even graduate students struggle to pace themselves on the final projects. For that reason, the following set of milestones are in place.

1. **Planning** Students must identify important theoretical material by week 5 of the course. Students must have a clear project in mind and at least 3 papers from the relevant literature. Students are encouraged to use this project to complement their graduate research. Students that are at a loss for what to pursue can look at the text book author's web site (http://sip.clarku.edu /projects.html) or speak to the instructor about ideas for good projects. Note that Physical Review articles are very easy to obtain online. I am interested in encouraging collaboration between science and computer science graduate students. Hence, group projects are a possibility, but should be discussed with me in advance.

2. **Implementation** In week 10 the students will have a working program that reproduces some essential behaviors suggested in the papers. Additionally, students will have to submit a clear outline of the proposed project.

3. **Interpretation** At the end of the semester the student will have completed the project in a satisfactory way. The results of the computer simulation will be appropriately reduced and interpreted.

The final report on the project will be a formal typed report, in a style specified on the course wiki. Students will also be required to deliver a 10–15 minute presentation on their project. There is no page requirement on the final project, but it should be a substantial piece of work. The final project will be graded based on the ”Project and Final Project Rubric” that appears on the course wiki. The final project should be graduate research quality; something that could lead to a professional paper or presentation.

Co-convening course

Special accommodations must be made for the fact that this course co-convenes, or involves both graduate and undergraduate students. Graduate students are expected to provide leadership and additional explanations to the undergraduates that are taking the course. Undergraduates are expected to engage in the challenging material that is presented, and work towards mastery. The course is organized in such a way that these interactions should be more natural, through the groups. More specifically, in this course

- “group assignments” will promote interaction between graduate and undergraduate group members.
- use of a wiki will allow collaborative content generation that works independently of student’s schedules.
- time for interaction at the computer will allow groups to interact in a structured format.
- graduate projects done outside of class will allow undergraduates to focus on course material.

Tentative Schedule:

MONDAY	WEDNESDAY
Jan 23rd **1** Python Programming: basic syntax, lists and arrays	25th **2** Graphics and Visualization: Scatter plots, density plots, 3D graphics, and animation
30th **3** Accuracy and Speed: Variables and ranges	Feb 1st **4** Accuracy and Speed: Numerical error and program speed
6th **5** Integrals and Derivatives: trapezoidal rule, Simpson’s rule, errors, Romberg integration, Gaussian quadrature	8th **6** Integrals and Derivatives: forward and backward differences, errors, central differences, second derivatives
13th **7** Solution to Linear and nonlinear equations: Gaussian elimination, back-substitution, pivoting	15th **8** Solution to Linear and nonlinear equations: LU decomposition, inverses, tri-diagonal systems
20th *Presidents Day*	22nd **9** Solution to Linear and nonlinear equations: Relaxation method and binary search

MONDAY	WEDNESDAY
27th **10** Solution to Linear and nonlinear equations: Newton's method, golden ratio search	Mar 1st **11** Fourier transforms: Discrete Fourier transform
6th **12** Fourier transforms: Fast Fourier transform	8th **13** **Midterm Exam**
13th **14** Ordinary Differential Equations: Euler's method	15th **15** Ordinary Differential Equations: Runge-Kutta method
20th *Spring Break*	22nd *Spring Break*
27th **16** Ordinary Differential Equations: leapfrog methods	29th **17** Ordinary Differential Equations: time reversal, energy conservation and the Verlet method
Apr 3rd **18** Ordinary Differential Equations: midpoint method, Bulirsch-Stoer method	5th **19** Ordinary Differential Equations: boundary value problems
10th **20** Partial Differential Equations: relaxation methods and boundary value problems	12th **21** Partial Differential Equations: overrelaxation and Gauss-Seidel methods
17th **22** Partial Differential Equations: initial value problems	19th **23** Partial Differential Equations: FTCS method
24th **24** Partial Differential Equations: numerical stability	26th **25** Partial Differential Equations: Crank-Nicolson methods, spectral methods
May 1st **26** Additional topics as time permits	3rd **27** Additional topics as time permits

Attendance Policy

Attendance is required and enters your grade as part of the in class assessment (10% of total grade). The policy for excusing absences is identical to that of late assignments.

Late Assignments

Other than in in exceptional circumstances, such as family or medical emergencies *late homework will not be accepted* unless an extension was agreed upon *well in advance* of the due date.

Academic Integrity

All students must practice academic honesty. Academic misconduct is subject to an academic penalty by the course instructor and/or a disciplinary sanction by the University. All students need to be familiar with the Student Conduct Code. I will follow the guidelines given there. In cases of academic dishonesy, I will seek out the maximum allowable penalty. If you have questions about which behaviors are acceptable, especially regarding use of code found on the internet or shared by your peers, please ask me.

Disabilities

Students with disabilities may request reasonable modifications by contacting me. The University of Montana assures equal access to instruction through collaboration between students with disabilities, instructors, and Disability Services for Students. Reasonable means the University permits no fundamental alterations of academic standards or retroactive modifications.