MTH 337: Introduction to Scientific and Mathematical Computing. Fall 2016.

Instructor:	Adam Cunningham	**Lecture:**	MWF	1:00-1:50pm	Math 250
	Math Building 106	**Lab:**	M	10:00-10:50am	Math 150
	adamcunn@buffalo.edu	**Office hours:**	MW	4:00-6:00pm	

Course Description

This course provides a broad introduction to computing in the sciences and in both abstract and applied mathematics. It is accessible to students early in their undergraduate program, thereby opening the door to the profitable use of computation throughout the junior and senior years.

The course covers the following areas:

- Programming using Python, the scientific computing package NumPy, and the plotting library Matplotlib.
- Applications of scientific computing in number theory, linear regression, dynamical systems, initial value problems, random number generation and optimization.
- Using computers to explore topics in the mathematical and natural sciences.
- Presentation of experiments, observations and conclusions in the form of written reports.

Prerequisities: MTH 141 and MTH 142 or MTH 154. No prior programming experience is needed.

Course Materials

Textbook: There is no required textbook for this class. We will be using online resources and extensive class notes. If you would like a book for reference, then either "Scientific Computation" by Bruce E. Shapiro (Sherwood Forest Books, 2015) or "Python for Scientists" by John M. Stewart (Cambridge University Press, 2014) is recommended.

Python: We will be using Python 3.5. It is recommended that you use the Anaconda distribution, which is available free on Windows, Mac and Linux and contains all the packages we need (NumPy, SciPy, Jupyter, Matplotlib).

Jupyter Notebook: Weekly reports will be written in Jupyter Notebook. This is included as part of the Anaconda distribution.

UBlearns: The course page on UBlearns will be used for submitting assignments and posting grades.

Website: `http://www.acsu.buffalo.edu/~adamcunn/MTH337.html`

Course Requirements

Laptop/Tablet: Classes take place in the Math Building and involve programming, so expect to bring your laptop/tablet to every class.

Class participation: Aspects include attending class, asking and answering questions in class and Piazza, and involvement in any group activities.

Quizzes: Short quizzes will be held frequently during both class and recitation.

Reports: Weekly reports will be submitted via UBLearns. The submission deadline will usually be 9am on a Tuesday. You are free to discuss the homework with others, but the work you turn in should be written by you alone.

Report Portfolio: You will generate a final portfolio collating all your reports into a single document.

Grading Policy

Course Grades: will be determined by averaging the grades with the weightings:

Weekly reports and final portfolio	70%
Quizzes	20%
Participation	10%

4-point Grading Scale: Reports will be assigned a letter grade on a four point grading scale:

A+ = 4.33	A = 4.00	A- = 3.67
B+ = 3.33	B = 3.00	B- = 2.67
C+ = 2.33	C = 2.00	C- = 1.67
D+ = 1.33	D = 1.00	D- = 0.67

+/- grades will be used in the computation of your overall grade. Note that the university does not permit A+ or D- grades.

Lowest Report Dropped: The lowest assignment grade will be dropped when computing your overall grade.

Extra Credit: Up to one full grade of extra credit (e.g. B → A) is available on every assignment for extra or unusual work or insight.

Late Assignment Policy: Assignments are usually due by 9am on a Tuesday. Late assignments will be accepted up to three days late, with a penalty of one full grade (e.g. A → B) for each day late.

Incompletes: Incompletes will be given only under extraordinary circumstances (such as surgery during the last week of class).

Expectations for this class

Attendance: You are expected to attend every class. If you miss a class you are responsible for getting the homework, lecture notes and any other in-class information or materials from a classmate.

Independent research: We cannot cover every single option available in Python in the limited time we have in class. It is expected therefore that you make use of the online resources to do research for youself. If there is something that you want to do, the chances are that Python already has a way to do it - you just need to find it.

Absolutely no non-class-related computer or phone use during class: (e.g. texting, Facebook, emailing): the penalty will be one third of a letter grade off overall course grade for each instance.

Academic Honesty: Students are expected to adhere to the university policy on academic honesty. Note particularly that

> **Written works must be the complete and original effort of the student, with all ideas and words of others duly attributed.**

Cheating, plagiarism, or misrepresentation of your work will result in formal charges.

How to succeed: Sustained steady effort, starting from day one. Make full use of the resources available to you: the classnotes, website, online documentation, Piazza, and each other. See me in office hours, email me, or make an appointment to see me at some other time. Start homework assignments early - programming and debugging often take more time than anticipated.

Other

Important dates:

Tue Sep 6	Last day to drop the course - no record appears on transcript.
Fri Nov 11	Last day to resign from the course - an R appears on the transcript.

Students with Disabilities: If you have a diagnosed disability (physical, learning or psychological) which will make it difficult for you to carry out the course work as outlined, or requires accommodations such as recruiting note takers, readers or extended time on quizzes and/or assignments, please advise me during the first two weeks of the course so that we may review possible arrangements for reasonable accommodations.

Student Learning Outcomes

Assessment measures: weekly programming assignments, in-class quizzes.

At the end of this course, a student will be able to:	Assessment
- use Jupyter Notebooks to write computer programs in the Python language. - use Markdown to format text within Jupyter Notebooks. - use MathTex to format mathematical expressions within Jupyter Notebooks.	All reports
- communicate their work in formal research-style reports.	All reports
- define and use Python integer, float, complex and boolean data types. - define and use Python list, string, tuple, set and dictionary data structures. - understand and use Python list comprehensions and generators.	All reports
- use the Python print and format commands to format and display results.	All reports
- use the Python control constructs if, else and elif. - use Python for and while loops, and break and continue statements. - define functions in Python using positional and keyword arguments.	All reports
- display numerical data using several Matplotlib plotting commands. - fully and accurately label Matplotlib graphs. - display multiple plots using the Matplotlib subplot command.	All reports
- use NumPy array creation operations. - understand the concepts of vectorization and array operations. - access NumPy arrays using indexing and slicing. - use NumPy boolean and integer array indexing.	Reports 3 - 11
- load numerical data from files using NumPy's loadtxt command.	Report 3
- time the execution of code using the IPython %timeit and %%timeit magics.	Reports 3 - 11
- understand how the RGB color model can be used to represent colors. - create color images using NumPy multidimensional arrays. - load, save and display images using imread, imsave and imshow.	Reports 5 - 6
- generate animations using the Matplotlib animation library.	Report 6
- understand how the WAVE file format is used to store audio data. - read and write sound files using the scipy.io.wavfile module.	Report 7
- load and write text data to and from files.	Report 11

- understand and implement Euclid's algorithm for the greatest common divisor. - test conjectures in number theory by finding examples or counterexamples. - search for patterns in the integers by making appropriate graphics.	Reports 1 - 2
- find the "best-fit" to real-world data using linear and quadratic models. - apply regression diagnostics to explore the residual errors of these models.	Report 3, 7
- understand the principles and details of the IEEE 754 floating point arithmetic standard. - recognize and avoid pitfalls of floating-point arithmetic using algebraic reformulation and Taylor-series approximations.	Report 4
- explore the dynamics of discrete dynamical systems in the form of cellular automata. - use color to make visual representations of the dynamics.	Report 5
- understand the concept of a system of ordinary differential equations, solve them using Euler's method and Heun's method, and make visual representations of the solutions. - formulate the differential equations for the motion of a magnetic pendulum, and simulate that motion.	Report 6
- acquire and process audio data from sensors on a smartphone. - analyze audio data to determine how well a theoretical model of inelastic collisions matches the experimental data.	Report 7
- understand the need for random numbers in mathematical modeling of physical, economic and social systems. - use a linear congruential generator to generate uniform pseudo-random numbers. - generate non-uniform random numbers with a specfied distribution, including Gaussian. - apply Monte-Carlo methods to area computation, surname extinction and other applications.	Report 8
- understand the method of golden section search and apply it to find the extremum of a unimodal function of a single variable. - formulate multivariable nonlinear optimization problems and solve them using the Nelder-Mead downhill simplex method and a simple version of the method of simulated annealing. - understand the principles of the Global Positioning System and solve a simplified version of this system.	Report 9
- understand linear optimization (linear programming) problems and solve them using the simplex method.	Report 10
- write computer programs to manipulate and analyze text. - explore and implement an n-gram probabilistic language model based on the word frequency distributions in works of literature.	Report 11

The table below indicates to what extent this course reflects each of the learning objectives of the undergraduate mathematics program. A description of learning objectives is available online.

Computational Skills:	**Analytical Skills:**	**Practical Problem Solving:**	**Research Skills:**	**Communication Skills:**
extensively	moderately	extensively	moderately	extensively

Class Schedule

Week	Programming Topics	Mathematical Topics
1	Introduction to Jupyter Notebook Numbers, variables, loops, functions	The Euclidean algorithm
2	Lists, namespaces Matplotlib basic plotting	Pythagorean triples
3	Report 1: Primitive Pythagorean Triples, due 9/13	
	Break, continue, list comprehensions NumPy arrays and vectorization	Prime number generation
4	Report 2: Prime Spirals, due 9/20	
	NumPy array indexing and slicing Loading numerical data from files	Linear least squares Floating point numbers
5	Report 3: Mauna Loa CO_2 Levels, due 9/27	
	RGB color representation Image creation and display	Hazards of floating point arithmetic
6	Report 4: Floating Point Numbers, due 10/4	
	Timing code execution NumPy fancy indexing	Deterministic dynamical systems Cellular automata
7	Report 5: Cellular Automata, due 10/11	
	Matplotlib animations	Continuous dynamical systems Initial value problems
8	Matplotlib subplots	Solving differential equations
9	Report 6: The Magnetic Pendulum, due 10/25	
	Recording and analyzing WAVE files	Data acquisition and analysis
10	Report 7: The Bouncing Ping-Pong Ball, due 11/1	
	NumPy random number generation	Random number generators Monte Carlo integration
11	Matplotlib histograms Matplotlib polar plots	Non-uniform random numbers
12	Report 8: The Generation and Use of Random Numbers, due 11/15	
	Matplotlib contour plots	Optimization
13	Report 9: Global Positioning System, due 11/22	
		Thanksgiving
14	Linear programming using SciPy	Linear programming
15	Report 10: Linear Programming, due 12/6	
	Python dictionaries String manipulation functions	Computing with text
	Report 11: Probabilistic Language Models, due 12/17	
	Portfolio, Due 12/18	