PHYS220/MATH220/CPSC220 : Scientific Computing I

Fall 2018

Instructor:	Dr. Justin Dressel	**Time:**	Tue/Thur 10:00am – 11:15am
Email:	dressel@chapman.edu	**Place:**	Leatherby Library B14
Office:	Keck Center 353		

Office Hours: 9:00am – 9:50am in KC353, or by appointment.

Course Page: `https://github.com/chapman-phys220-2018f`

Course Description: This example-driven course introduces computation as a tool for scientific exploration. Topics include manuscript preparation with LaTeX and Jupyter, test-driven development, numerical methods with arrays/dataframes, and symbolic computation. Modern languages like Python, MATLAB, Sage, and Julia are emphasized.

Prerequisites: CPSC230

Course Textbook:

Effective Computation in Physics: Field Guide to Research with Python, Scopatz & Huff, 2015.
`http://physics.codes`

Supplementary Python/SciPy Web Resources:

`http://github.com/chapman-phys220-2018f/info`
`http://www.scipy-lectures.org`
`http://lectures.quantecon.org/py/learning_python.html`
`http://numba.pydata.org/numba-doc/0.12.2/quickstart.html` (Numba Quickstart)
`https://youtu.be/eKlNEJlSjuE` (Introduction to Numba)
`https://youtu.be/6oXedk2tGfk` (Numba for Accelerating Scientific Computation)

Grading Policy:

Homework (30%), In-class work (20%), Midterm Project (20%), Final Project (30%).

Organization:

This course is intended to develop a professional skill set aimed at getting jobs in physics, mathematics, or computer science after college. The tools we will use are not watered down—they are exactly the tools becoming increasingly used in industry and academia for scientific computing and data science. *Practice is essential.* As such, it is your responsibility to arrive prepared to complete the in-class group projects in an efficient and productive manner. You are expected to turn in professional-quality work that is documented according to industry standards. Your work should be bug-free, well-tested, and elegantly written. Half of your job is solving interesting problems; the other half of your job is proudly presenting your results as a professional.

The bulk of in-class time will be spent in groups solving interesting physical problems using a computer. You will put into practice what you have read from the textbook and free web resources, with guidance from the instructor and peers. Homework will be a more challenging extension of what is covered in class to provide additional practice. The midterm and final projects will be more substantial and completed individually. They will require a synthesis of all topics learned in the course to complete successfully.

CoCalc (formerly SageMathCloud)**:**

To guarantee software compatibility, we will be using `http://cocalc.com` as a cloud-based computing solution that you can access with any browser. Your first task is to ensure that you have received an account invitation to your Chapman email, and that you can access your course project. Please choose a username such that your real name is clear. Your account will give you access to a virtual Linux machine running Ubuntu, complete with an accessible bash terminal, text editors like `vim` or `nano` or `emacs`, the document preparation language LaTeX, interactive Jupyter notebooks, Numeric Python, Scientific Python, Pandas, Numba, C, C++, Julia, Octave, and many other useful tools: `http://tutorial.cocalc.com/students/`

Laptop:

You are welcome to use software installed on your own laptop in addition to *CoCalc*, but it is your responsibility to manage that software and ensure compatibility with *CoCalc*. The software you need is almost entirely contained in the python distribution *Anaconda* (`http://www.anaconda.com/download`), which includes Python, Numeric Python, Scientific Python, Pandas, Numba, Julia, and many other packages. You will also need access to a bash terminal and the `git` program: both are automatically available in Linux or Mac, and can be installed in Windows as part of the *Git for Windows* (`http://git-scm.com/download/win`) application.

GitHub:

Homeworks, classworks, and exams will be assigned and collected on the industry-standard cloud code-sharing site GitHub (`http://www.github.com`), using the `git` change-control tool. Your second task will be to ensure that you have an account, and know how to use it. The GitHub Organization for the course will be `http://github.com/chapman-phys220-2018s`.

Slack:

Group discussion and contact with the professor will be facilitated by the industry-standard chat application Slack, at `http://scststudents.slack.com`. Your second task will be to ensure that you have an account. Please notify the instructor if you need to be invited. The channel for this course will be `#phys220-2018f` and is set to auto-notify the instructor. Note that the class channel is a public forum, but private chats are also available as needed.

Course Learning Outcomes:

1. Create professional reports in LaTeX, including proper section structuring, figures, and references. Augment these static reports with dynamic Jupyter notebooks that are targeted for web publication as supplementary information.

2. Demonstrate industry-standard software engineering principles, including modular design, test-driven development, change control (git), and the efficient use of console editors (nano/vim) for code development.

3. Write simulation code involving precision numerical methods and array-based numerical processing in Python and MATLAB, as well as dataframe manipulation and symbolic computation in Python. Demonstrate the use of vectorized code and compiled libraries for efficiency, as well as a familiarity with just-in-time compilation.

4. Produce accurate simulations of challenging physical systems, with the explicit goal of using those simulations to answer interesting scientific questions.

Physics Program Learning Objectives:
Upon graduation, students will:

1. Demonstrate knowledge and understanding of basic mathematics and physical principles used to model natural phenomena.

2. Demonstrate ability to convey physical concepts with mathematical expressions and/or computation, and effectively derive quantitative predictions from a model through mathematical/computational analysis.

3. Demonstrate competency in using computer tools.

4. Demonstrate the ability to apply advanced knowledge of advanced mechanics, electromagnetism, thermodynamics and quantum physics to the solution of problems in physics.

5. Demonstrate the ability to effectively communicate information, scientific or otherwise, in both written and verbal form.

6. Demonstrate the ability to write clear, organized and illustrated technical reports with proper references to previous work in the area.

7. Demonstrate the skills and motivation for continued self-education.

Chapman University Academic Integrity Policy:

Chapman University is a community of scholars which emphasizes the mutual responsibility of all members to seek knowledge honestly and in good faith. Students are responsible for doing their own work, and academic dishonesty of any kind will not be tolerated anywhere in the university. At their discretion the faculty may submit work to plagiarism detection software, such as www.turnitin.com for review.

Chapman University Students with Disabilities Policy:

In compliance with ADA guidelines, students who have any condition, either permanent or temporary, that might affect their ability to perform in this class are encouraged to contact the Office of Disability Services. If you will need to utilize your approved accommodations in this class, please follow the proper notification procedure for informing your professor(s). This notification process must occur more than a week before any accommodation can be utilized. Please contact Disability Services at (714) 516-4520 or http://www.chapman.edu/students/studenthealth-services/disability-services if you have questions regarding this procedure, or for information and to make an appointment to discuss and/or request potential accommodations based on documentation of your disability. Once formal approval of your need for an accommodation has been granted, you are encouraged to talk with your professor(s) about your accommodation options. The granting of any accommodation will not be retroactive and cannot jeopardize the academic standards or integrity of the course.

Chapman University Equity and Diversity Policy:

Chapman University is committed to ensuring equality and valuing diversity. Students and professors are reminded to show respect at all times as outlined in Chapmans Harassment and Discrimination Policy: http://ow.ly/XEwTu Any violations of this policy should be discussed with the professor, the Dean of Students and/or otherwise reported in accordance with this policy.

Student Support at Chapman University:

Over the course of the semester, you may experience a range of challenges that interfere with your learning, such as problems with friend, family, and or significant other relationships; substance use; concerns about personal adequacy; feeling overwhelmed; or feeling sad or anxious without knowing why. These mental health concerns or stressful events may diminish your academic performance and/or reduce your ability to participate in daily activities. You can learn more about the resources available through Chapman University's Student Psychological Counseling Services here: http://www.chapman.edu/students/health-and-safety/psychological-counseling/

Approximate Course Outline:

Week	Tuesday	Thursday	Topics
1: 8/28, 8/30	Orientation	Read: Ch. 1	CoCalc, `bash`, Jupyter
2: 9/4, 9/6	Read: Ch. 15	Read: Ch. 16	`git`, GitHub
3: 9/11, 9/13	Read: Ch. 2	Read: Ch. 3, 4	Python : modules, containers, logic
4: 9/18, 9/20	Read: Ch. 5	Read: Ch. 17	Python : functions, generators, debugging
5: 9/25, 9/27	Read: Ch. 6	Read: Ch. 18	Python : classes, testing
6: 10/2, 10/4	Read: Ch. 7		Python : visualization (`matplotlib`)
7: 10/9, 10/11	Read: Ch. 9		Efficient python: arrays and `numpy` **(Midterm project assigned)**
8: 10/16, 10/18	Read: Ch. 10		Efficient python: managing data (HDF5)
9: 10/23, 10/25	Read: Ch. 11		Efficient python: data structures **(Midterm project due)**
10: 10/30, 11/1	Read: Ch. 12		Efficient python: parallelization
11: 11/6, 11/8	Numba Quickstart	Numba Videos	Efficient python: compiling with `numba`
12: 11/13, 11/15	Read: Ch. 20		Publication: LaTeX and Jupyter
13: 11/20, 11/22	*No Class*	*No Class*	**Fall Break**
14: 11/27, 11/29			From `scipy` to MATLAB **(Final project assigned)**
15: 12/4, 12/6			Extras: `sympy`, Julia, and Sage **(Work on final project)**
16: 12/11	10:45am–1:15pm	*No Class*	**Finals Week** (Final project due)