Physics 514

Computational Physics

September 5, 2017

Welcome

- Emanuel Gull
- egull@umich.edu,
- 2253 Randall (office hours at 10 AM on Thursdays)
- 4246 Randall (TA & Discussion section, 5 PM on Wednesdays)
- http://www-personal.umich.edu/~egull/
- Check canvas site for up to date lecture information and lecture notes of previous year

Welcome

- Please register!!!
- If you cannot register please go to student services and tell them they should register you.
- If you are still waitlisted please go to student services and tell them to help you.
- If it still does not work please come to me and let me know. Thursday 10 AM, 2253 Randall

No classes / Classes taught by substitute

- Tuesday Nov 14 / 16
- Thanksgiving, Nov 23
- Fall break, Tue Oct 17

Nov 14/16 is yet to be decided.

Extra classes

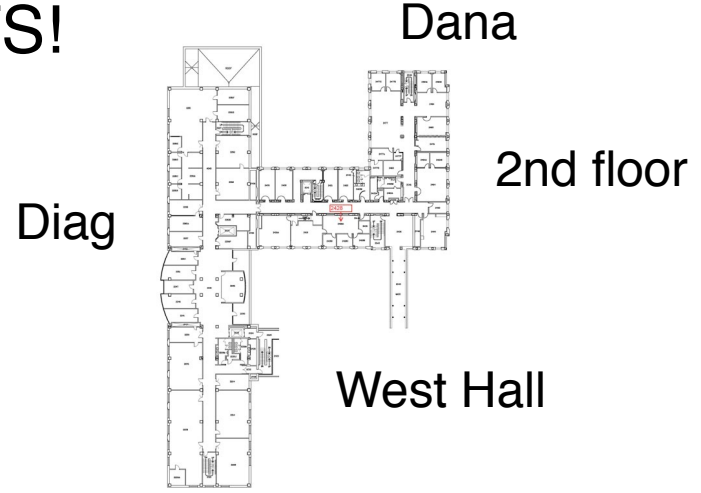
- Next week: Python workshop, open for all: Four special lectures
- Location: 335 West Hall, across from 340 (colloquium room)
- Will teach you how to program in Python
- 5:00 PM to 6:30 PM
- Everybody welcome

Class schedule

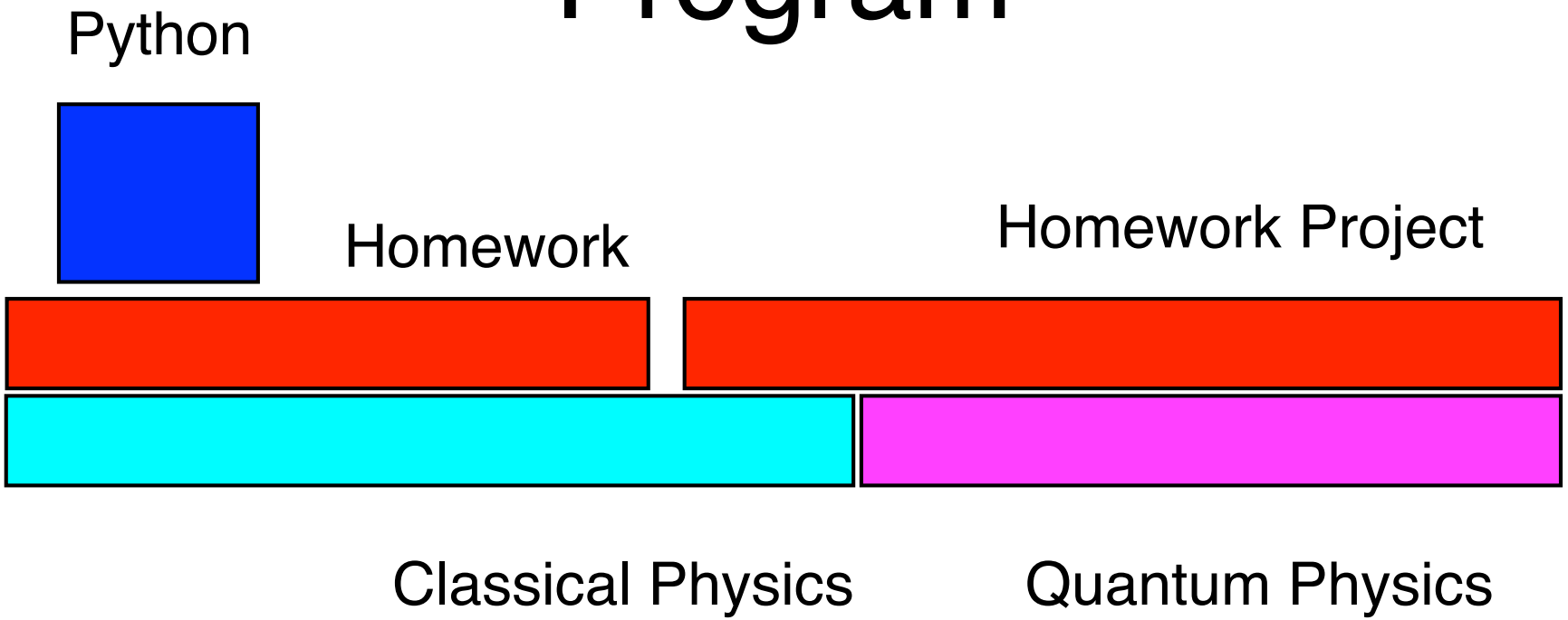
- Please go to canvas and have a look at the syllabus
- Homework handed in BY 10 AM on Thursday. Exceptions only after discussion with Shang
- Helpful tool: subscribe to ical feed / online calendar online to stay up to date on topics.
- Contact: preferred through canvas

Welcome! (admin stuff)

- We meet Tuesdays & Thursdays 8:40 - 10:00 AM
- Python mini course: we meet Monday/Tuesday/ Wednesday/Monday NEXT WEEK 5:30-7:00 in 340 and 335 West Hall
- Please bring a laptop (first two weeks only, hands on training). Loner laptops are available at CATS
- Please install python (at least for the first two weeks). You will need python for Stat Mech and E&M. Python is also available in the computer rooms
- Computer problems: Talk to CATS!



Program



- Main grade from homework project & **presentation/ report**. Presentations in the last week of classes. Details yet to be decided but last day for project selection is October 12

Homework

- Simulation project: Phase transitions in classical spin systems (think 2d Ising, Heisenberg, etc)
- Typical case: Write a Monte Carlo simulation for a system with a classical phase transition.
- Debug it, test it! (…Shang and I are here to help… go to office hours and set up appointments)
- Implement cluster updates, compute observables (specific heat, susceptibility, etc)
- Perform correct error analysis and error propagation
- Perform finite size scaling
- Find the critical exponents, plot them
- Present a brief report that documents what you've been doing
- Give a 10 minute presentation on your project to the rest of the class (last week of the term, potentially add. meetings)

This class…

- …Is a grad level physics class!
- Undergrad standing is OK but good grasp of upper level undergrad classes is required. Historical average grade for undergrads is B+
- …traditionally has a mix of physicists, computer scientists, chemists, applied physicists, electrical engineers, etc.
- …is not a programming class. We will implement algorithms but most of the class will not deal with programming questions
- …counts for the ARC / MICDE PhD in Scientific Computing
- …counts for the ARC / MICDE Graduate Certificate in Computational Discovery and Engineering

PhD in scientific computing

http://micde.umich.edu/ph-d-in-scientific-computing/

OVERVIEW

This program is intended for students who will make extensive use of large-scale computation, computational methods, or algorithms for advanced computer architectures in their doctoral studies. A firm knowledge of the scientific discipline is essential.

This is not a stand-alone degree; it is a joint degree program. Students must be accepted into the PhD program of a home department. The actual degree name will have "…and Scientific Computing" appended to the the normal title, e.g., "Ph.D. Degree in Aerospace Engineering and Scientific Computing."

Students may apply to the program after having completed one term, **but prior** to being promoted to candidacy status.

GRADUATION REQUIREMENTS

Students are expected to complete the normal doctoral requirements of their home department as well as meet additional requirements in the area of scientific computation. The specific requirements will vary with the choice of home department; however, there are a few general course requirements for all students in this program. The student must elect:

- nine (9) credit hours in numerical analysis, and
- nine (9) credit hours in computer science and applications in scientific computation outside the home department.

The latter category will typically include courses in computer science, parallel algorithms, advanced computer architectures, computational fluid dynamics, or other courses in scientific computation not offered by the home department. Students are expected to work closely with their advisors, who will help to determine their courses.

If you consider this please set up an appointment with an adviser EARLY so that the program does not delay your candidacy!

Certificate in scientific comp.

http://micde.umich.edu/certificate/

OVERVIEW

The Graduate Certificate in Computational Discovery and Engineering trains participants to conduct computationally intensive research, and prepares them to function effectively in interdisciplinary research and product development settings that employ high-performance computing.

Advances in computational algorithms and HPC hardware have transformed CDE into a predictive science that yields quantitative insights into the behavior of realistic systems. CDE is now a fundamental tool in scientific research and modern design and manufacturing practices. The Graduate Certificate in CDE recognizes students' competence in this field, and prepares them for success in the following areas, among others:

- aerospace engineering
- material physics and biophysics
- biomedical engineering
- material science
- data mining
- parallel computing
- nuclear engineering
- electronic and communication engineering
- structural engineering
- climate and space science modeling
- industrial and operations engineering.

The program is open to all academic units and all students pursuing M.S. or Ph.D. degrees at the University of Michigan. Academic requirements include coursework in approved classes, an experiential component such as internships, and attendance at MICDE events. Master students have the opportunity to participate in the Multidisciplinary Design Program.

Classical Physics

- Partial Differential Equations
- Classical N-body problems
- Integration methods
- Percolation and classical spin problems

There will be some overlap with Physics 411. If you are an undergrad, please take Physics 411 instead.

Quantum Physics

- Exact Diagonalization
- Hartree Fock and related methods
- Lattice Monte Carlo
- DMRG
- Continuous-time quantum Monte Carlo

We'll see how far we get – possibly we'll do less / different topics depending on interest. Preliminary schedule is online.

Who here:

- Has previous programming experience?
- Is a good programmer?
- Knows Python?
- Has used Python before?
- knows/has used before/heard of: C, C++, D, fortran, Perl, Oberon, Pascal, Eiffel, Modula, Cobol, Basic, VisualBasic, C#, Matlab (octave), Mathematica, Maple?

Python facts

- Interpreted language (…well… it's complicated…)
- Modern concepts: object oriented, dynamically typed, garbage collection, exceptions, free and open source
- Developed in the 1980s, popular & in wide use since around 2000
- Comparatively easy to use
- Distributed with most modern operating systems (Microsoft is the big exception)
- Large collection of useful libraries (see week 2 for some of them)
- language of many every-day programs and used in major software companies:
  - Internet: YouTube,Google BitTorrent,Cisco, IBM, …
  - Science: NASA, Los Alamos, JPL, Fermilab, …
  - Banking: Chase, UBS, …

Python mini course

http://mirlyn.lib.umich.edu/Record/004188099

[other versions of the book available, max 7 concurrent views]

- Install python, if you have not done so already (choose version 2.6 or 2.7, anything above 2.4 should do).
- get the e-book from the library and follow along (see hp for more information and links)!
- If it goes too fast: tell me, this is supposed to be for everyone / non-experts
- If it goes too slow: follow the e-book by yourself, read through some of the more advanced chapters.
- By the end of the first course you will be able to use python as a pocket calculator and write small functions.
- Second course: Numbers, Strings, Lists, Dictionaries, Tuples, Functions, Modules, Control flow
- Third course: Useful Python libraries (part I): numpy and scipy; matplotlib
- Fourth course: lots of useful exercises