

Learning through Computation

in Upper-Division Physics

David Roundy

Oregon State University

National Science Foundation

DUE-1141330



No templates needed

- students write their programs from scratch
- they google for help

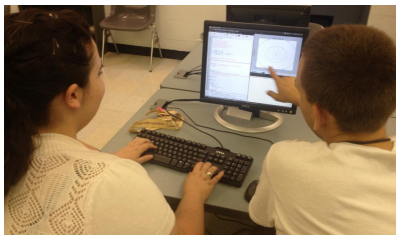
Pair programming

- students work in pairs: a *drive* and a *navigator*
- roles are swapped every 30 minutes or so
- “show and tell” when projects are done



A computational lab for traditional courses

- students have a very busy schedule: just one credit
- reinforce learning in traditional courses
- save time by not having to introduce the physics



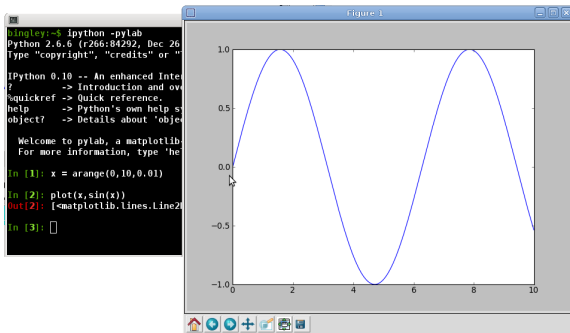
All work is done in the lab

- Today *all* physicists need to program
- Struggling students make little progress outside of class
- These are the students who *need* a computational course

Python and matplotlib

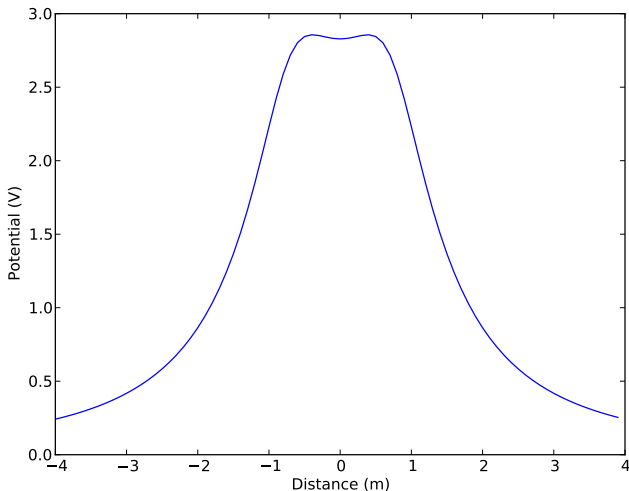
Reasoning

- free software, readily available to students
- ease of use and power comparable to Matlab
- used by professional scientists
- tutorials and help readily available on web



Electrostatics (student work)

Day 1: 4 point charges

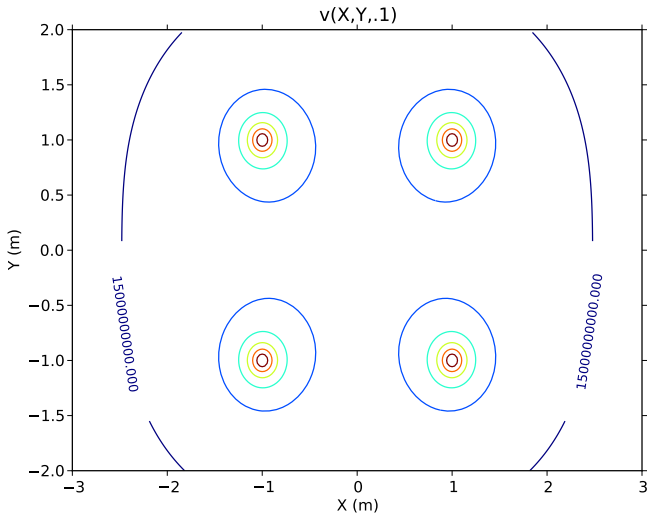


● how to compute distances

● how to plot

Electrostatics (student work)

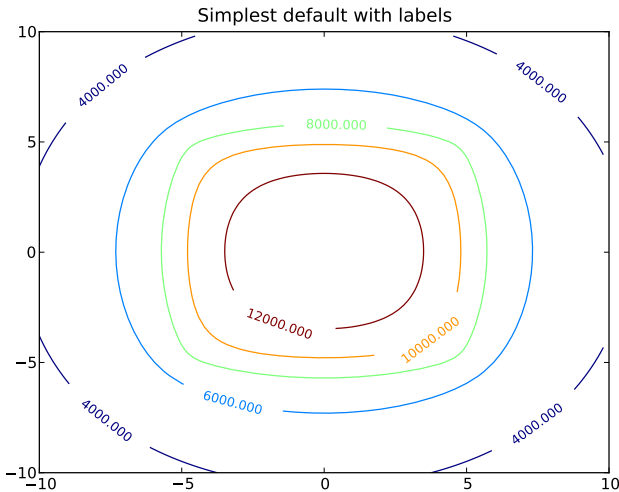
Day 2: 4 point charges



- plotting with slices

Electrostatics (student work)

Day 3: Square of charge

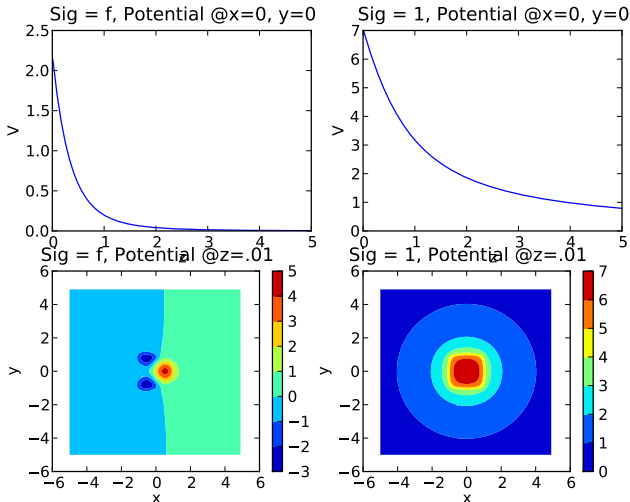


● chopping and adding

● googling for help

Electrostatics (student work)

Day 4: Square of charge (with varying density on left)



- visualizing in multiple dimensions

The physics continues...

- spheres, cylinders and hemispheres
 - curvilinear coordinates
 - distances in curvilinear coordinates
- Rigid pendulum
 - Verlet's method
 - first integral of the motion
 - Fourier transforms
- ...

Conclusions (or keys to success)

- no templates required
- alongside traditional course
- all-in-class
- pair programming
- python/matplotlib