

Supporting the integration of computation into physics courses

Danny Caballero
Michigan State University



MICHIGAN STATE
UNIVERSITY

PERL
 **@MSU**



MICHIGAN STATE
UNIVERSITY

perl@pa.msu.edu
perl.pa.msu.edu

Mentor Faculty/Affiliates

Danny Caballero
Katie Hinko
Paul Irving
Vashti Sawtelle
David Stroupe (TE)
Brian O'Shea
Stuart Tessmer
Alicia Alonzo (TE)

Postdocs/Scientists

John Aiken
Mashood KK
J.T. Laverty
Angie Little
Bill Martinez

Graduate Students

Kelsey Funkhouser
Brian Hancock
May Lee
Abhilash Nair
Mike Obsniuk
Alanna Pawlak
Tom Finzell (FAST)

Legacy

Leanne Doughty (PD)
Adam Fritsch (FAST)
Steve Wolf (PD)
Paul Hamerski (REU)
Laura Hunter (REU)
Sarah Boyer (REU)

Undergraduates

Justin Gambrell
Kristy Griswold
Nat Hawkins
Bridget Humphries
Maddie Klinkoski
Sam Luba
Sonny Ly
Claire Morrison
Keenan Noyes
Zach Nusbaum
Anna Turnbull



Reasons you might consider computation:

Computational modeling can enhance
(and perhaps improve) instruction

Experience with computational modeling
is expected of our graduates

Computational modeling is
how modern science is done

Classroom use with computational modeling
can provide authentic science experiences

Developing the 21st century scientific workforce

Ivie & Stowe, 2002

Gaps in undergraduate program:

Using scientific software

Using computational modeling

Chonacky & Winch, 2008

Few departments teaching computational modeling

Lack of resources & expertise

Where does it go?

Caballero, Chonacky, Hilborn, & Merner, ~~2017~~

Comprehensive AIP-hosted survey

All departments; random faculty

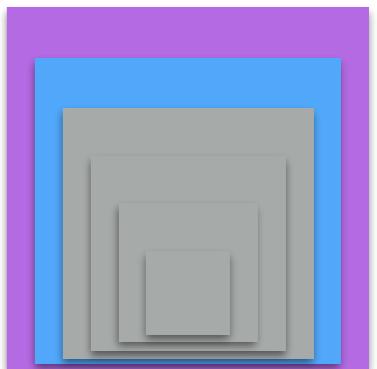
2018

THE AMERICAN ASSOCIATION OF PHYSICS
TEACHERS URGES THAT EVERY PHYSICS AND
ASTRONOMY DEPARTMENT PROVIDE ITS MAJORS
AND POTENTIAL MAJORS WITH APPROPRIATE
INSTRUCTION IN COMPUTATIONAL PHYSICS.

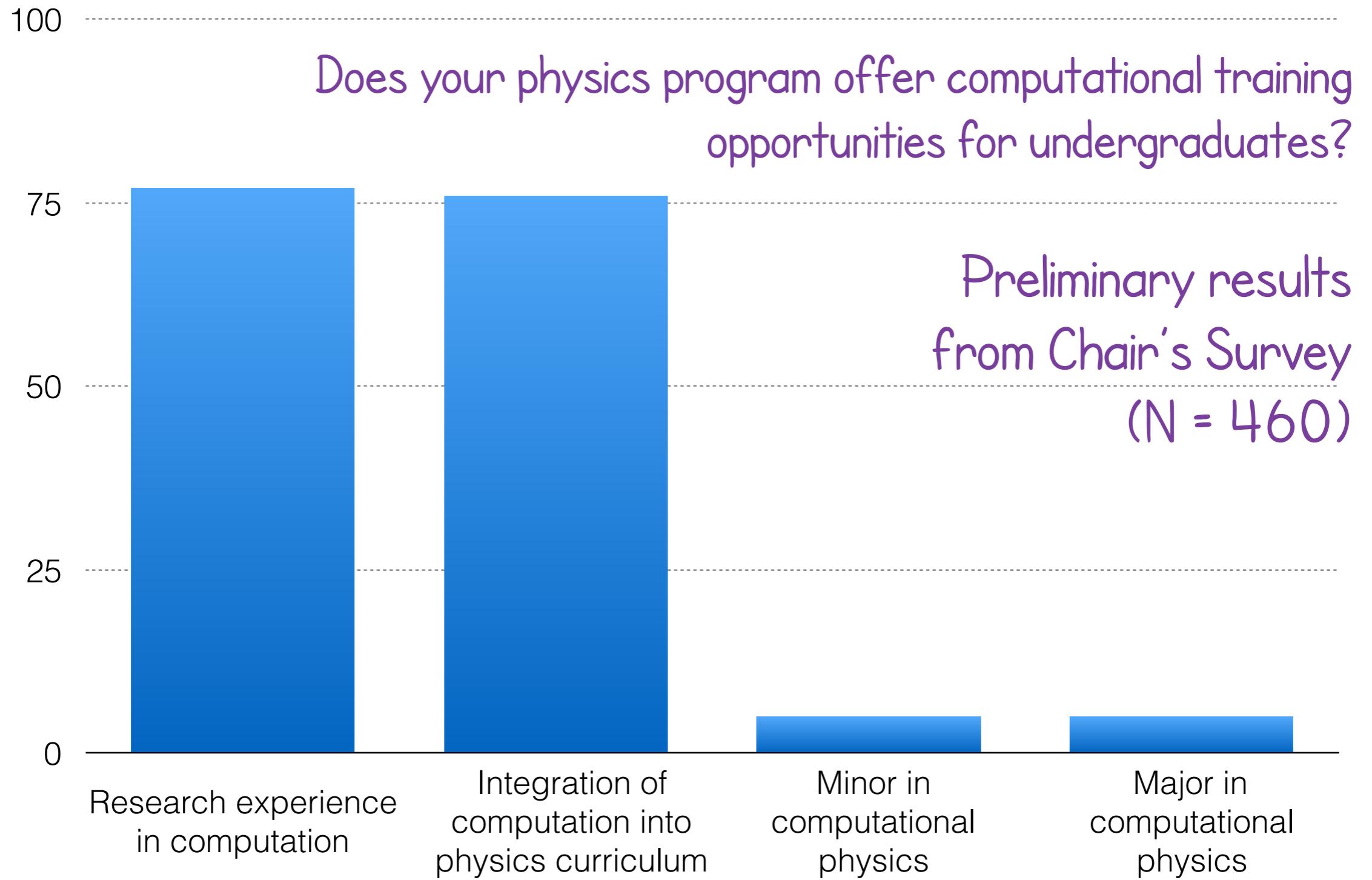
APPROVED BY THE AAPT EXECUTIVE BOARD AT THE SPRING 2011 MEETING.

Surveying the state and implications of computational physics instruction

- Distribute a survey of faculty to investigate the current state of computational physics instruction
- Draw implications for efforts to bolster computational instruction
- Track changes to the state over time



Surveying the state and implications of computational physics instruction



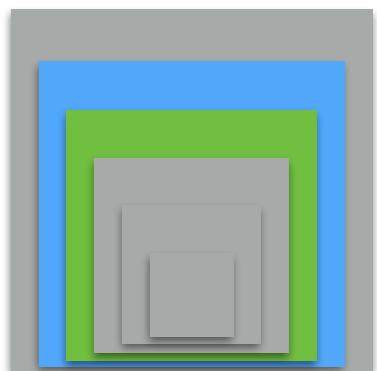
Surveying the state and implications of computational physics instruction

Expected Outcomes:

- Implementation models that interested departments and programs can replicate
- Factors that characterize successful/failed implementations
- Identify departments to study in more detail

Integrating Computation into Undergraduate Physics

- Develop, support, and grow a community of faculty working to integrate computation into their courses
- Co-develop exercises for a wide variety of courses across platforms and implementations
- Study how faculty move into this community



w/ K. Roos, M. Lopez del Puerto,
L. Engelhardt, R. Hilborn, & P. Irving
DUE-1524128, DUE-1525525, DUE-1525062,
DUE-1524493, & DUE-1524963



PICUP

Integrating Computation into Undergraduate Physics

Summer Residential Workshop

PICUP Summer 2016 Workshop Details

Dates: August 1-5, 2016; Participants should plan to arrive on Sunday, July 31.

Location: University of Wisconsin at River Falls

Energize Undergraduates!



Integrating Computation into Undergraduate Physics

The screenshot shows the PICUP website's 'Exercise Sets' page. At the top, there's a red diagonal banner with the text 'BETA - Click to give feedback' and a green circular icon. The main header features the 'PICUP' logo and the text 'Partnership for Integration of Computation into Undergraduate Physics'. Navigation links include 'Home', 'Exercise Sets' (which is the active tab), 'Resources', 'Events', and 'About PICUP'. There are also 'Login | Register', 'About Us | Contact Us', and a 'Feedback' button. The AAPT logo is in the top right corner.

Browse Exercise Sets

Course
Any
 Mechanics
 Electricity & Magnetism
 Waves & Optics
 Thermal & Statistical Physics
 Modern Physics
 Quantum Mechanics
 Mathematical/Numerical Methods
 Experimental Labs
[more](#)

Course Level
Any
 First Year
 Beyond the First Year
 Advanced

Programming Language
Any
 IPython/Jupyter Notebook

19 Exercise Sets Sort by:
Subject

A Rigid Three-bar Pendulum	
Binary Stars with Equivalent One Body Problem	
Falling Sphere with Air Resistance Proportional to v^2	
Lunar Lander	
Simple Hanging Harmonic Oscillator	
Traveling to Mars	
Calculating the magnetic field with the Biot-Savart Law	
Electric Field Due to a Uniformly Charged Ring	
Electric Field Due to a Uniformly Charged Rod	
Equipotentials and Electric Field Lines: Collections of Point Charges and the Method of Images	
Motion of a Charged Particle in a Magnetic Field	
The Wien (E x B) Filter	



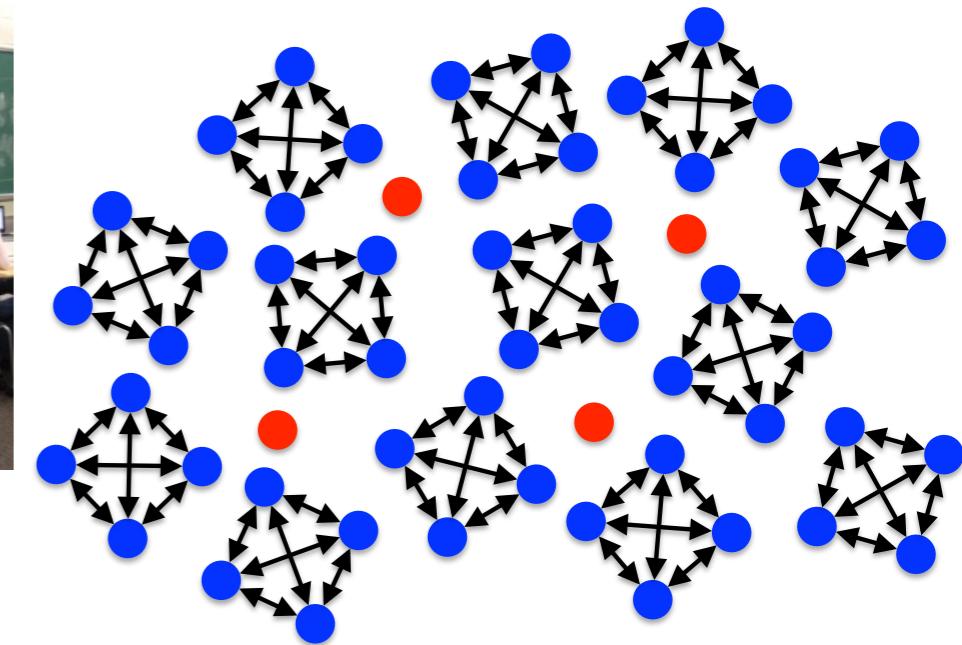
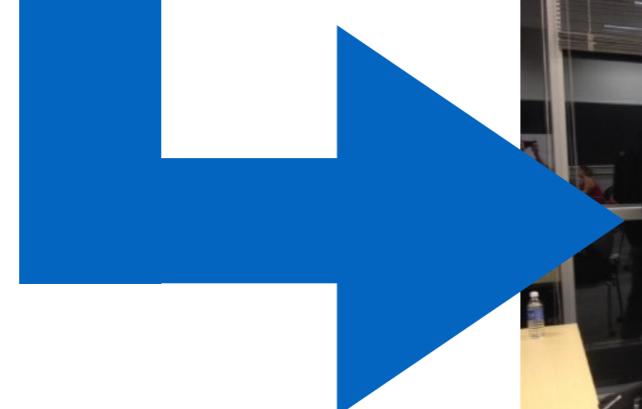
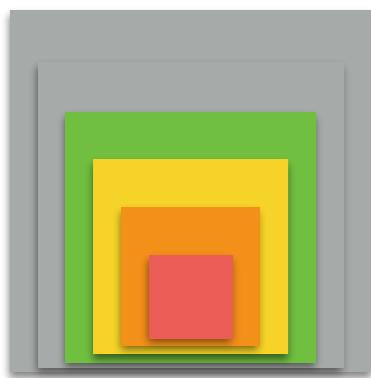
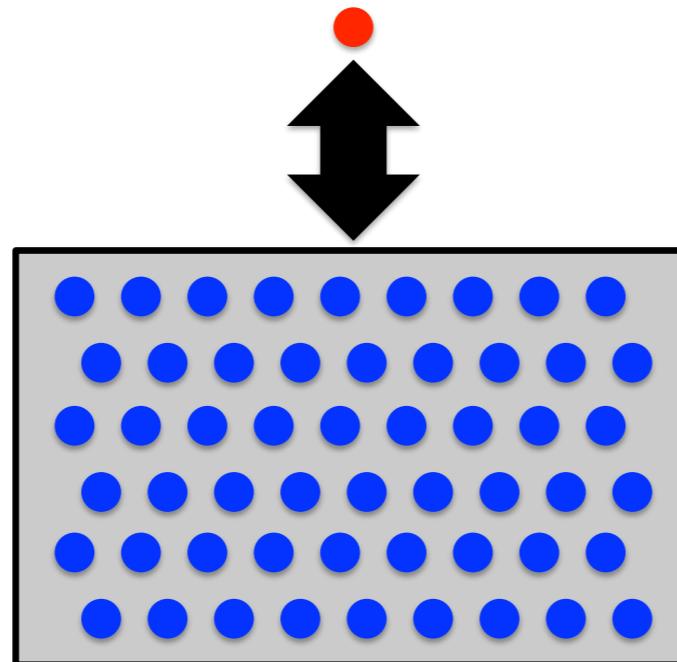
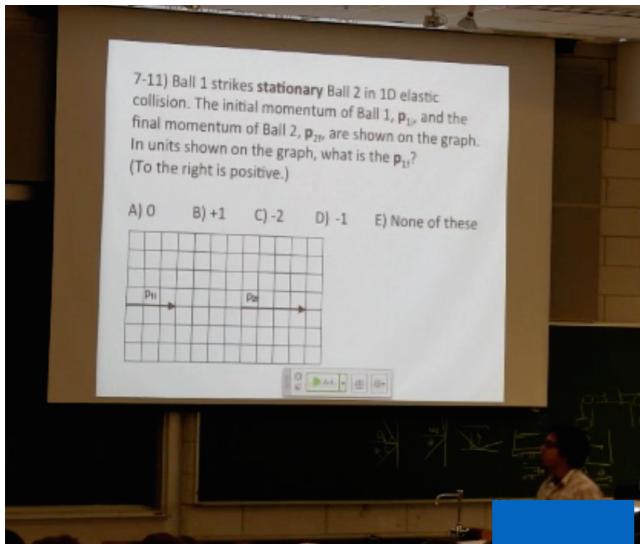
compadre.org/picup brought to you by the tireless efforts of B. Mason & L. Barbato

Studying faculty movement into the community

Expected Outcomes:

- Specific practices and norms of the computational physics education community
- Factors that influence how new members take up and modify those practices and norms
- Structural and community supports needed to bring new members into the community

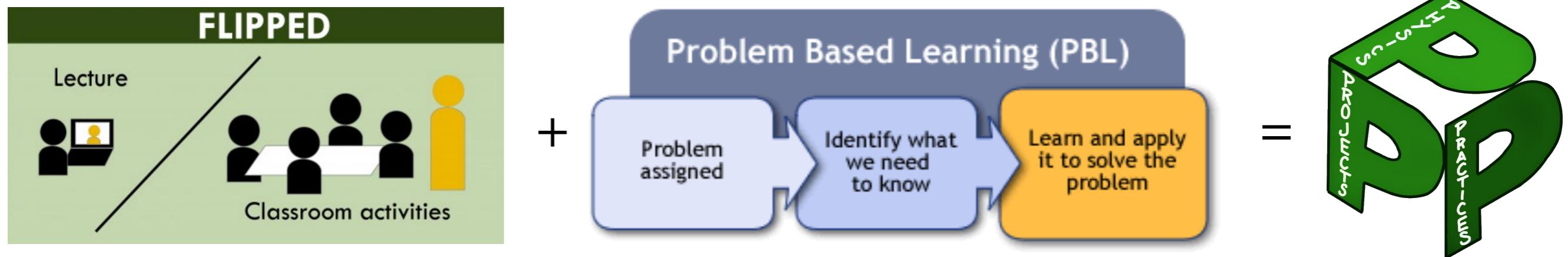
Projects and Practices in Physics (P³)

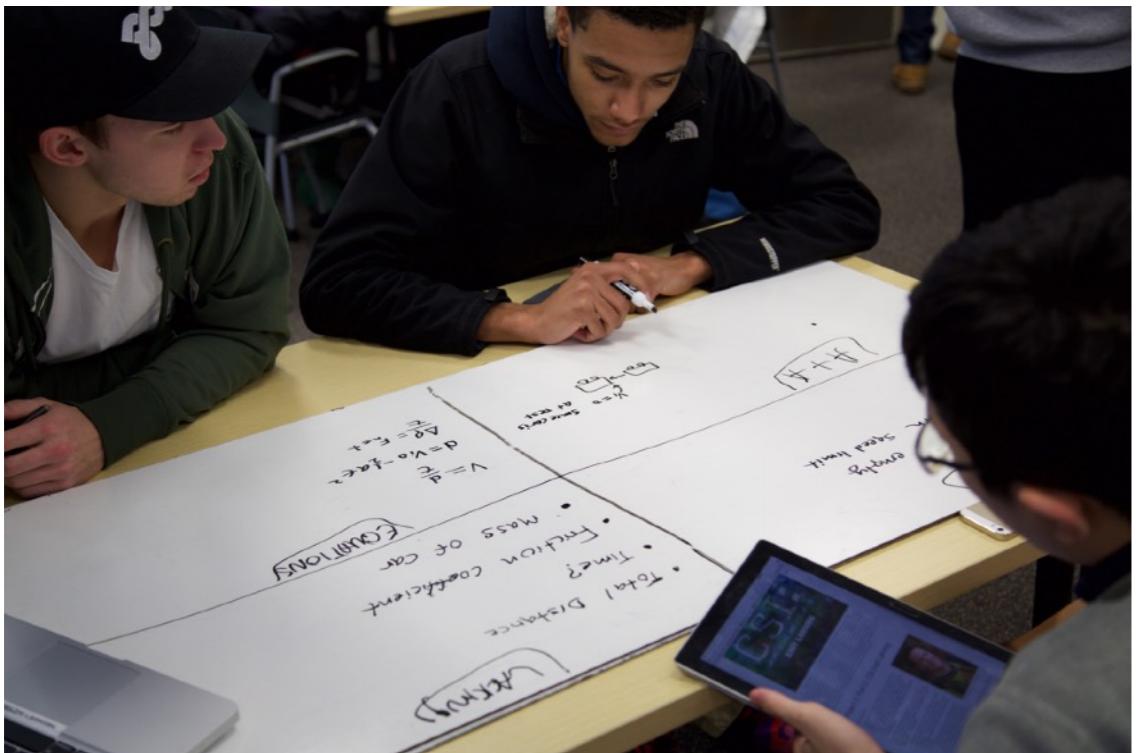


pcubed.pa.msu.edu
<http://arxiv.org/abs/1607.04455>
w/ P. Irving & M. Obsniuk

A Typical Week

- Monday: Pre-class reading/HW
(conceptual clicker-style)
- Tuesday: In-class project (analytical)
- Thursday: In-class project (analytical/computational)
- Sunday: Post-class HW
(standard end-of-chapter + conceptual)





MINIMALLY WORKING PROGRAM

```
from __future__ import division
from visual import *
from visual.graph import *
from physutil import *

# Window setup
scene.width = 1024
scene.height = 760

# Objects
Earth = sphere(pos=vector(0,0,0), radius=6.4e6, material=materials.BlueMarble)
Satellite = sphere(pos=vector(7*Earth.radius, 0,0), radius=1e6, color=color.red, make_trail=True)

# More window setup
scene.range=12*Earth.radius

# Parameters and Initial conditions
mSatellite = 1
pSatellite = vector(0,5000,0)

# Time and time step
deltat = 1
t = 0
tf = 60*60*24

SatelliteMotionMap = MotionMap(Satellite, tf, 20, markerScale=2000, labelMarkerOrder=False)

#Calculation Loop
while t < tf:
    theta = (7.29e-5) * deltat      #      IGNORE THIS LINE
    Earth.rotate(angle=theta, axis=vector(0,0,1), origin=vector(0,0,0))      #      IGNORE THIS LINE
    rate(10000)

    Satellite.pos = Satellite.pos + pSatellite/mSatellite*deltat

    SatelliteMotionMap.update(t, pSatellite/mSatellite)

    t = t + deltat
```

```

from __future__ import division
from visual import *
from visual.graph import *
from physutil import *

# Window setup
scene.width = 1024
scene.height = 760

# Objects
Earth = sphere(pos=vector(0,0,0), radius=6.4e6, material=materials.BlueMarble)
Satellite = sphere(pos=vector(42164e3, 0,0), radius=1e6, color=color.red, make_trail=True)

# More window setup
scene.range=12*Earth.radius

# Parameters and Initial conditions
mSatellite = 15e3
pSatellite = mSatellite*vector(0,3073,0)
G = 6.67e-11
mEarth = 5.97e24

# Time and time step
deltat = 1
t = 0
tf = 60*60*24

SatelliteMotionMap = MotionMap(Satellite, tf, 20, markerScale=2000, labelMarkerOrder=False)
FnetMotionMap = MotionMap(Satellite, tf, 20, markerScale=2000, labelMarkerOrder=False)

sepgraph = gcurve(color=color.red)

#Calculation Loop
while t < tf:
    theta = (7.29e-5) * deltat      #      IGNORE THIS LINE
    Earth.rotate(angle=theta, axis=vector(0,0,1), origin=vector(0,0,0))      #      IGNORE THIS LINE
    rate(10000)

    Fgrav = -G*mSatellite*mEarth*Satellite.pos/(mag(Satellite.pos)**3)
    Fnet = Fgrav

    Satellite.pos = Satellite.pos + pSatellite/mSatellite*deltat
    pSatellite = pSatellite + Fnet*deltat

    SatelliteMotionMap.update(t, pSatellite/mSatellite)
    FnetMotionMap.update(t, Fnet)

    sepgraph.plot(pos=(t,mag(Satellite.pos)))

    t = t +deltat

```

AFTER

“Open-source” Materials*

The Concept of Iterative Prediction

“Iterate” means to “repeat.” In physics, it often means to perform the same calculation repeatedly using information produced by the previous calculation. You might think of this as taking the output of a calculation and using it as the input for the new calculation.

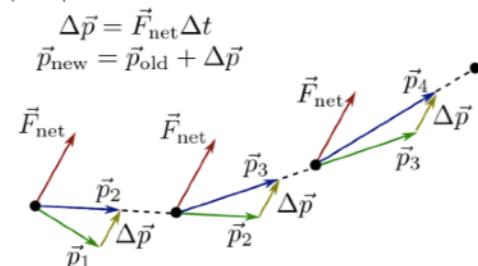
To predict motion iteratively is to apply the [momentum update](#) and [position update](#) formula over small time steps, using their own predictions and the inputs for the next calculation. The steps for iteratively prediction motion are as follows:

- Calculate the (vector) forces acting on the system.
- Update the momentum of the system: $\vec{p}_f = \vec{p}_i + \vec{F}_{\text{net}} \Delta t$.
- Update the position of the system: $\vec{r}_f = \vec{r}_i + \vec{v}_{\text{avg}} \Delta t$.
- Repeat

This process can be used for any system with any type of force. The accuracy of your predictions depend on the length of the time step. By using this method, you assume that the net force and average velocity are roughly constant over the time interval (for each time interval). If you are interested in more details, this method is similar to [Euler-Cromer symplectic integration](#).

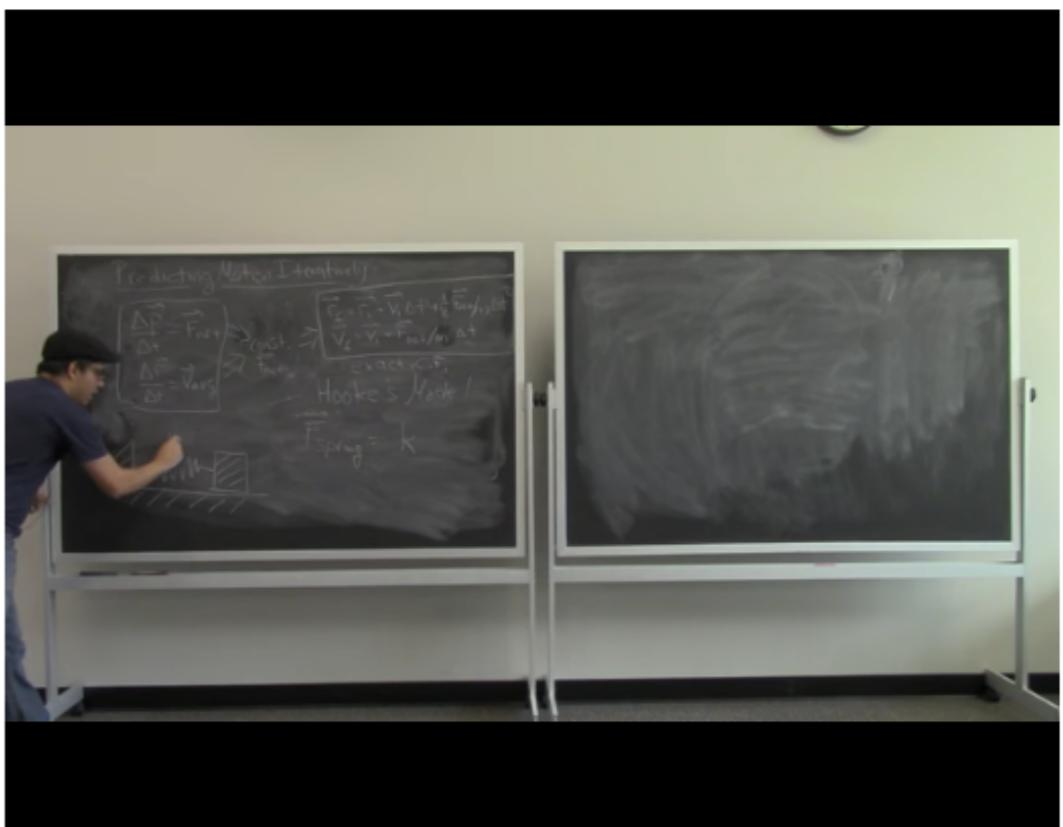
Applying Iterative Prediction

To reiterate, this method is not limited to non-constant forces and can be used to predict the motion in situations where a constant force model can be applied. A visual representation of such an iterative prediction over 3 steps is shown below. In each step, the momentum changes and, thus, the new momentum is calculated. This new momentum is used to determine the new location of the ball. The process is executed again with an updated prediction.



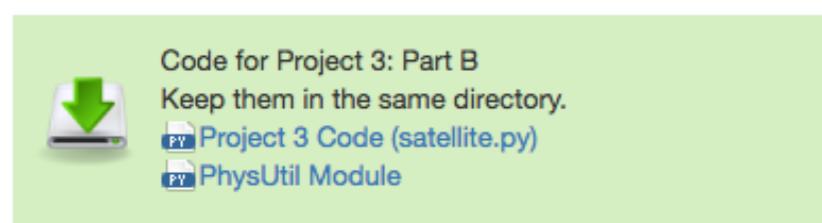
If you were to connect the straight lines in this picture, you would see a trajectory that looks more like moving through a curved trajectory. The time step here is quite long for the motion, but using a shorter time step, the line segments are shorter and more closely produce a curved trajectory.

Lecture Video



Project 3: Part B: Geostationary orbit

Carver is impressed with your work, but remains unconvinced by your predictions. He has asked you to write a simulation that models the orbit of the satellite. To truly convince Carver, the simulation should include representations of the net force acting on the spacecraft, which has a mass of 15×10^3 kg. Your simulation should be generalized enough to model other types of orbits including elliptical ones.



pcubed.pa.msu.edu



*E&M materials this summer/fall

Weekly Feedback

- Students are provided with weekly formative feedback on:
 - their individual understanding
 - their group's understanding
 - their role in facilitating/focusing the group's efforts



Irving, Sawtelle, Caballero 2015 PERC
Irving et. al, TPT (in preparation)

Weekly Feedback

“Liam’s” Feedback:

tendency to jump quickly into a plan of action without evaluating whether it is the right plan

like to see you take on a more guiding role

expect you to encourage another group member to take the lead when working with Python.



Irving, Sawtelle, Caballero 2015 PERC
Irving et. al, TPT (in preparation)

Liam given “just-in-time” feedback



Upper-Level Physics, too!

The screenshot shows a GitHub repository page for `dannycab / phy481msu`. The repository has 2 unwatched users, 0 stars, and 0 forks. The main navigation bar includes links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Pulse, Graphs, and Settings. The current branch is `gh-pages`, and the file path is `phy481msu / jupyter / HW6-MethodOfRelaxation.ipynb`. A commit by `dannycab` titled "homework and relaxtion code updated" was made 2 days ago, with a commit hash of `3f148ee`. The commit message also lists "1 contributor". Below the commit, the file statistics are shown as 245 lines (244 sloc) and 8.94 KB. Action buttons for Raw, Blame, History, and file operations (Copy, Edit, Delete) are available.

Method of Relaxation

In this problem, you will use the method of relaxation to solve for the electric potential in a 2D rectangular box that is 10m by 10m. One side is set to a potential of 10V while the other three sides are set to 0V. Much of the code that sets up this problem has been written, but the core physics and computational steps are missing (see below).

To successfully solve this problem, you will have to understand what different chunks of program do below. Some discussion is provided about each chunk.

Setup and import libraries the needed libraries

Below, we import the relevant libraries needed for our calculations and graphing. We also set a plot style and the size of the figures.

```
In [2]: import matplotlib
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline

plt.style.use('fivethirtyeight')
matplotlib.rcParams['figure.figsize'] = (7.0, 7.0)
```



<https://dannycab.github.io/phy481msu/>

PUNCHLINE: PHYSICS EDUCATION RESEARCH CAN HELP SUPPORT AND FACILITATE THE COMING COMPUTATIONAL REVOLUTION

- Research with students
- Research on activities, pedagogy, curricula
- Research with faculty
- Research with departments & larger systems

Partnerships with "traditional" faculty are incredibly important to this work.

Thank you!
Questions?

caballero@pa.msu.edu

perl.pa.msu.edu

gopicup.org

