DAMOP 2016 - May 25, 2016

Teaching Computation Across the Curriculum

Danny Caballero
Michigan State University

Roadmap

- Why Computational Modeling?
- Implementation Examples
- Research
- Parting Thoughts

Computational modeling is…

Computational modeling is…

Using the computer as a tool
to solve, to simulate, and / or
to visualize a physical problem.

Computational modeling is…

Using the computer as a tool
to solve, to simulate, and / or
to visualize a physical problem.

High-level computing languages +
Powerful computers

Computational modeling is…

Using the computer as a tool
to solve, to simulate, and / or
to visualize a physical problem.

High-level computing languages +
Powerful computers

Some programming is necessary.

Computational modeling is…

Using the computer as a tool
to solve, to simulate, and / or
to visualize a physical problem.

High-level computing languages +
Powerful computers

Some programming is necessary.
Can be limited as **you** see fit.

Reasons you might consider computation:

Reasons you might consider computation:

Computational modeling can enhance (and perhaps improve) instruction

Reasons you might consider computation:

Computational modeling can enhance
(and perhaps improve) instruction

Experience with computational modeling
is expected of our graduates

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

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

Implementation Examples*

Specific goals will depend on your needs and constraints

*Your mileage will vary.

Implementation Spectrum*

*Not to scale

Implementation Spectrum*

"Slight"
Modifications to
Existing Course

Upper-Level
Physics

*Not to scale

Implementation Spectrum*

“Slight” Modifications to Existing Course

Upper-Level Physics

Implementing & Adapting Existing Curricula

Introductory & HS Physics

*Not to scale

Implementation Spectrum*

"Slight" Modifications to Existing Course	Implementing & Adapting Existing Curricula	Soup-to-Nuts Design & Implementation
Upper-Level Physics	Introductory & HS Physics	Introductory Physics

*Not to scale

Implementation Spectrum*

"Slight" Modifications to Existing Course	Implementing & Adapting Existing Curricula	Soup-to-Nuts Design & Implementation
Upper-Level Physics	Introductory & HS Physics	Introductory Physics

Goals, Instruction, & Assessment

*Not to scale

Context:

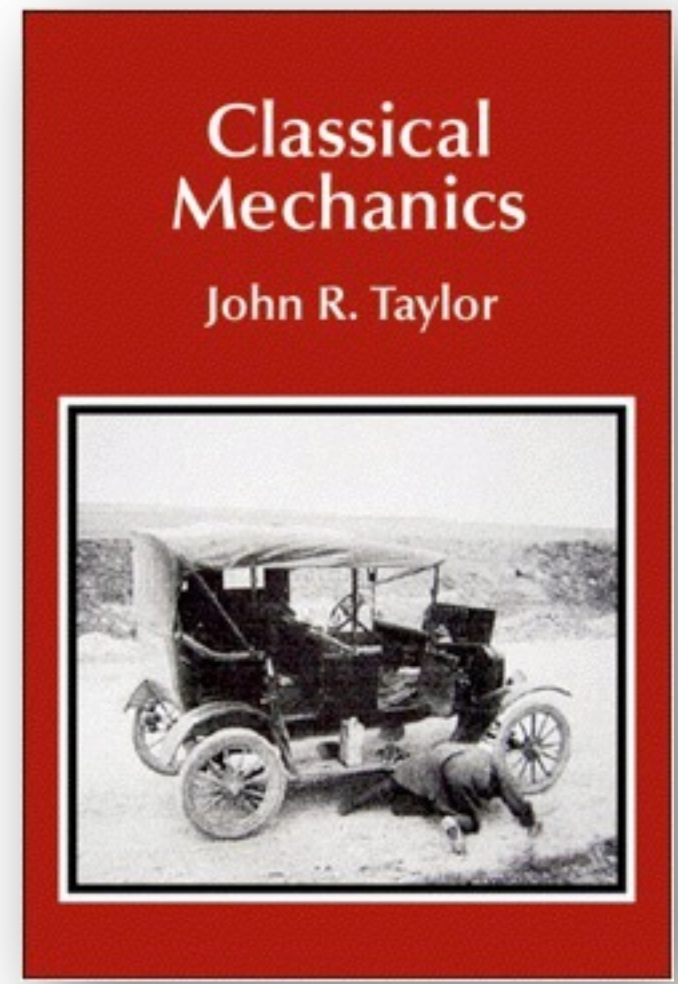
Classical Mechanics

w/ S. Pollock

Context:

Classical Mechanics

- Sophomore-level course
  40-80 students/semester



w/ S. Pollock

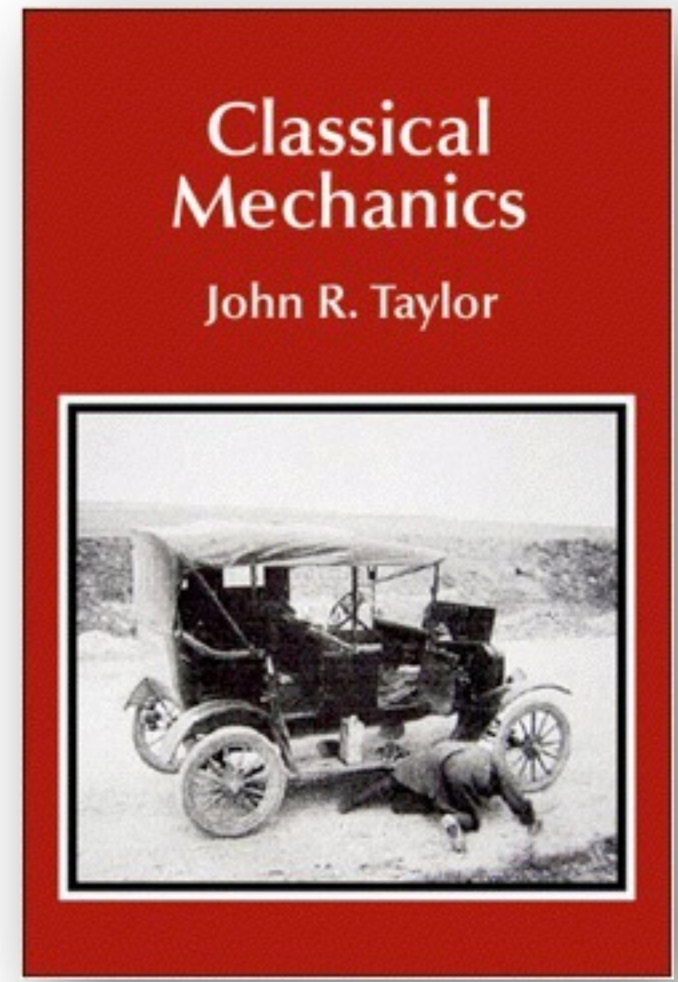
Pepper, et al., PERC Proc. 1413, 291 (2011)
Pollock, et al., PERC Proc. 1413, 303 (2011)

Context:

Classical Mechanics

- Sophomore-level course
  40-80 students/semester

- Pedagogically transformed
  JiTT, Clickers, Tutorials



w/ S. Pollock

Pepper, et al., PERC Proc. 1413, 291 (2011)
Pollock, et al., PERC Proc. 1413, 303 (2011)

Computational Learning Goals

Students should be able to...

Computational Learning Goals

Students should be able to...

Map Physical Situations

Computational Learning Goals

Students should be able to...

Map Physical Situations

Identify Appropriate Tools

Computational Learning Goals

Students should be able to...

Map Physical Situations

Identify Appropriate Tools

Implement Computational Methods

Computational Learning Goals

Students should be able to...

Map Physical Situations

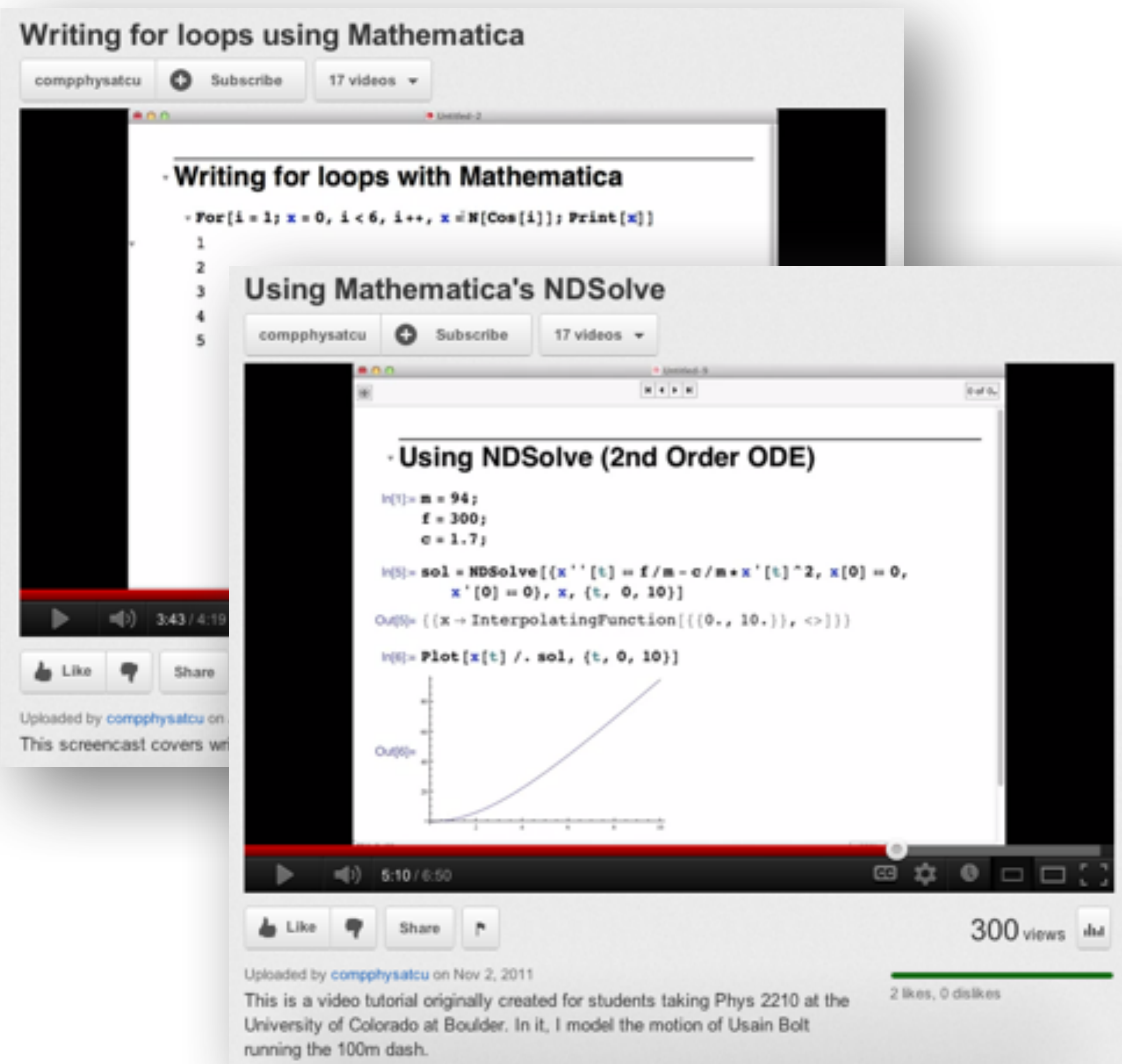
Identify Appropriate Tools

Implement Computational Methods

Reflect on Output

Modular and On-demand Instruction

Modular and On-demand Instruction



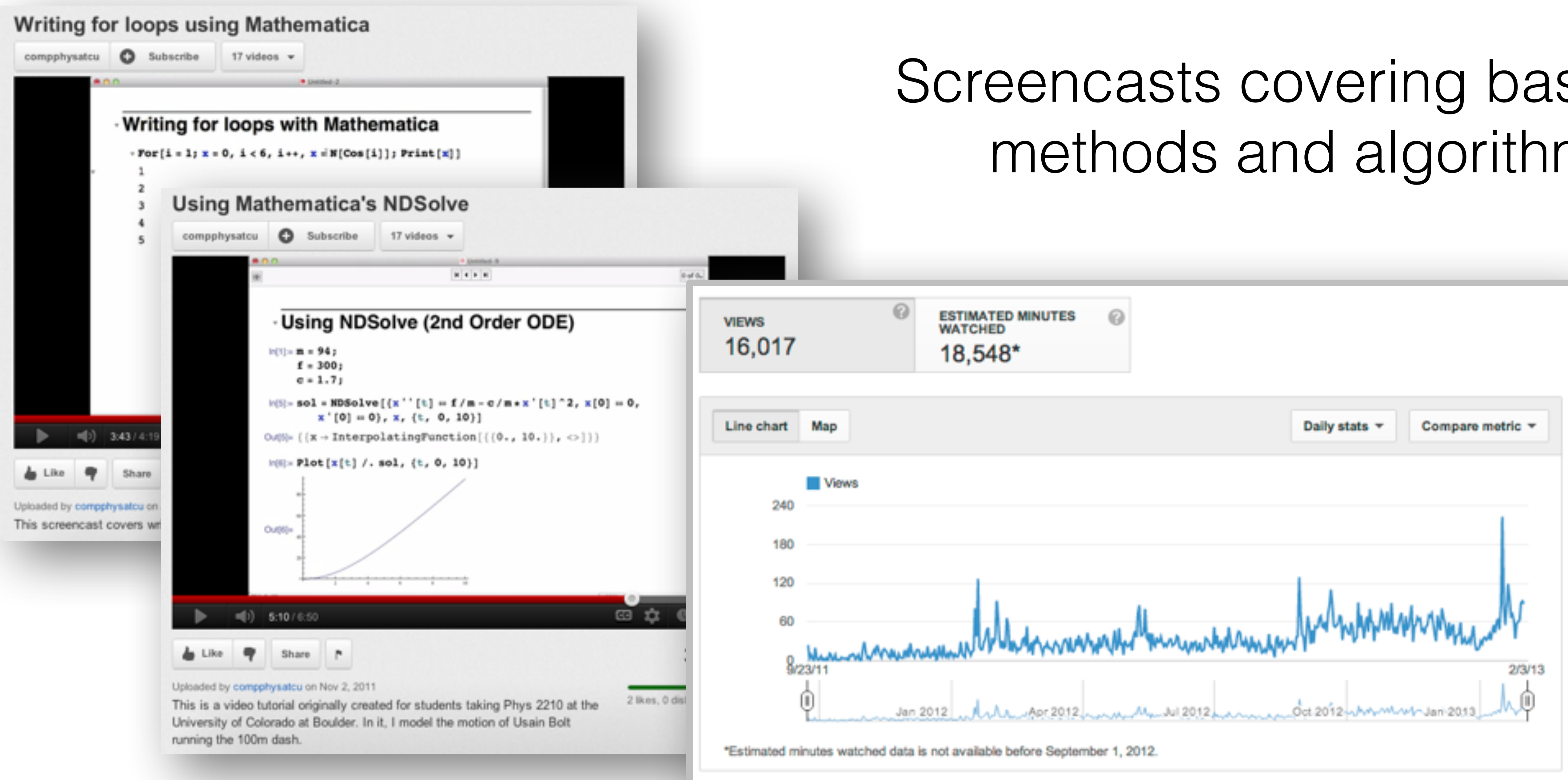
Screencasts covering basic methods and algorithms

Online Resources: www.youtube.com/compphysatcu

w/ B. Zwickl

Modular and On-demand Instruction

Screencasts covering basic methods and algorithms



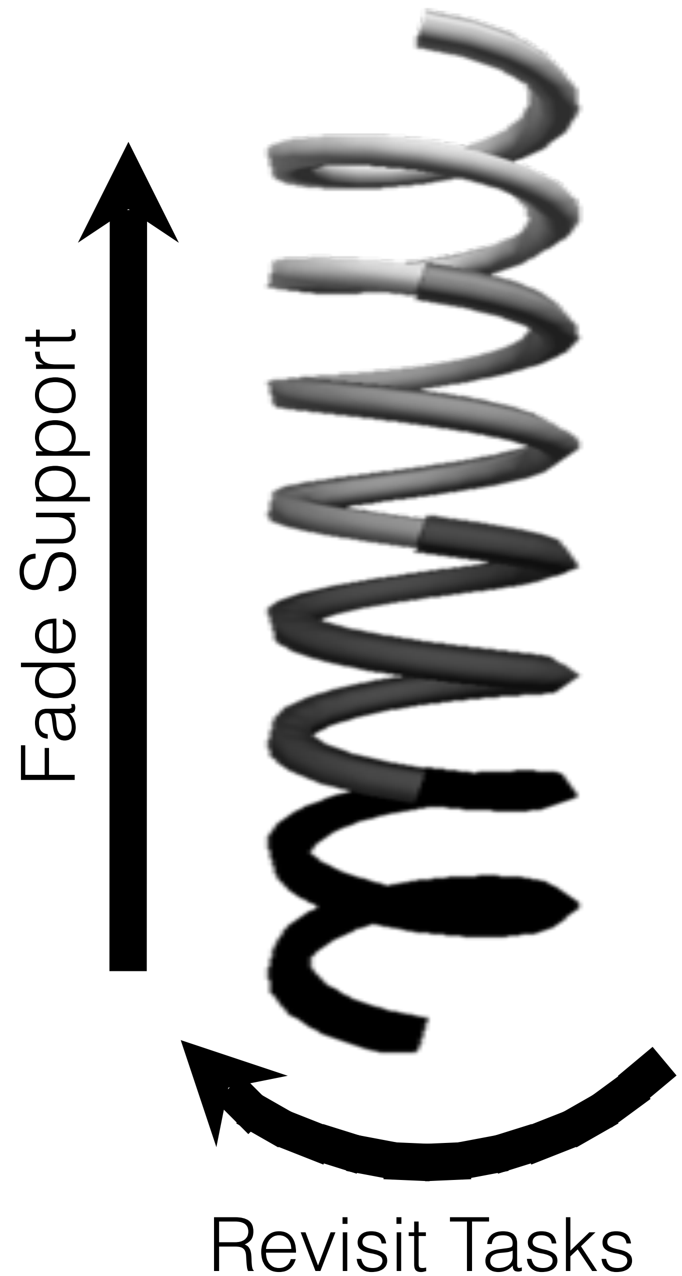
Online Resources: www.youtube.com/compphysatcu

w/ B. Zwickl

Computational Homework

Goal: Model equations of motion numerically
Week 11 - Extract initial conditions from video
Compare model to collected data.

Computational Homework

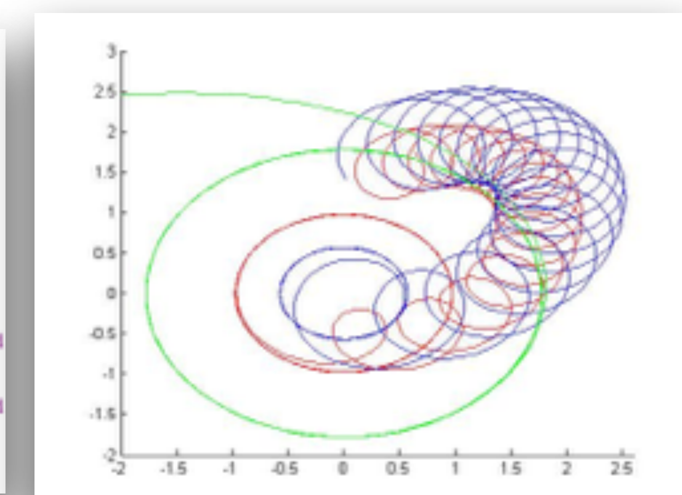
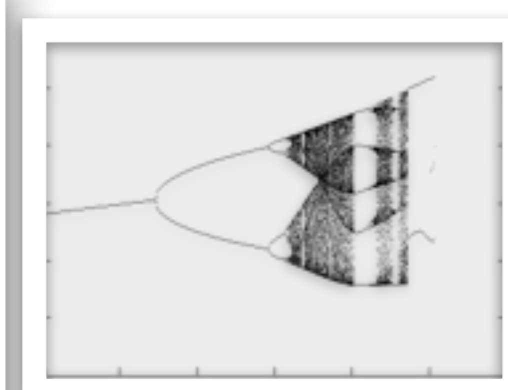


Goal: Model equations of motion numerically
Week 11 - Extract initial conditions from video
Compare model to collected data.

Caballero & Pollock, Am. J. Phys., 82, 231 (2014)

Final Computational Project

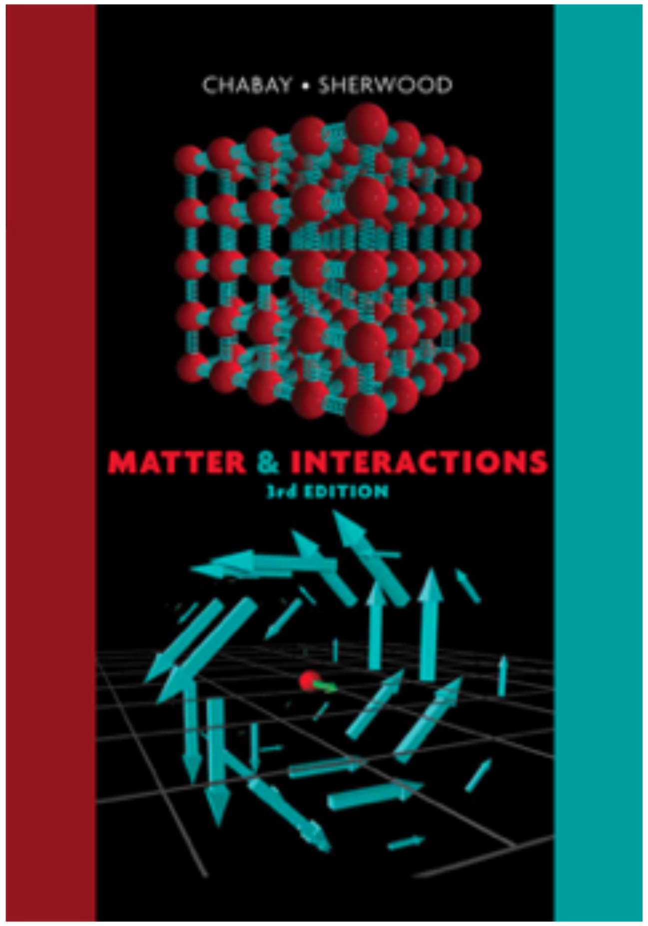
- Project + Paper in Groups of 2 or 3
- Proposal (4 weeks out) - Feedback
- Update + Calculations (2 weeks out) - Feedback
- Topics selected by students


Much more info: http://tinyurl.com/cu-comp-project

Context:

Calculus-Based Physics

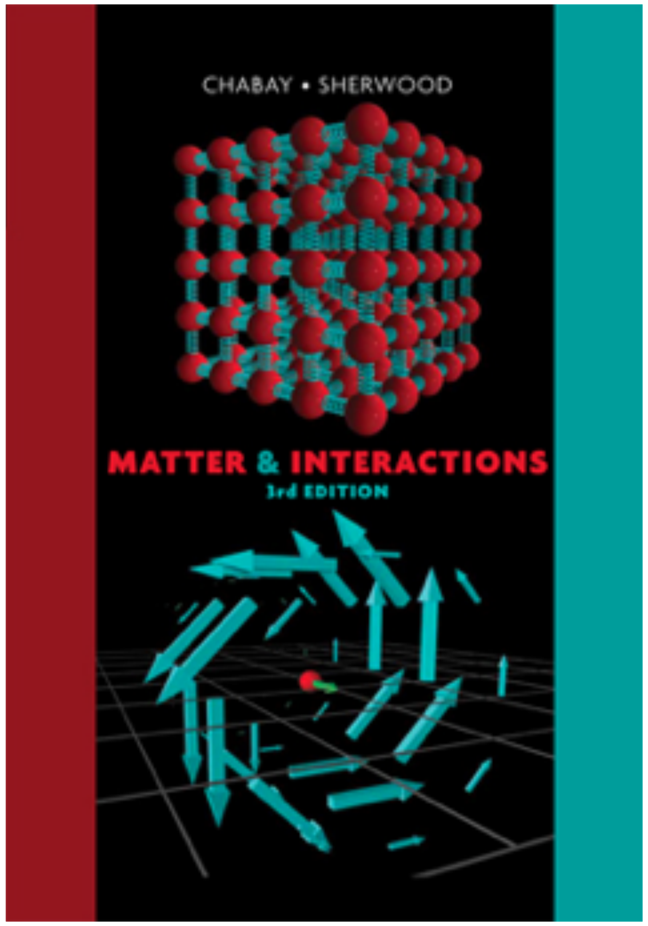


www.matterandinteractions.org

Context:

Calculus-Based Physics

- Matter and Interactions (M&I)
  R. Chabay & B. Sherwood
  Wiley, 2010



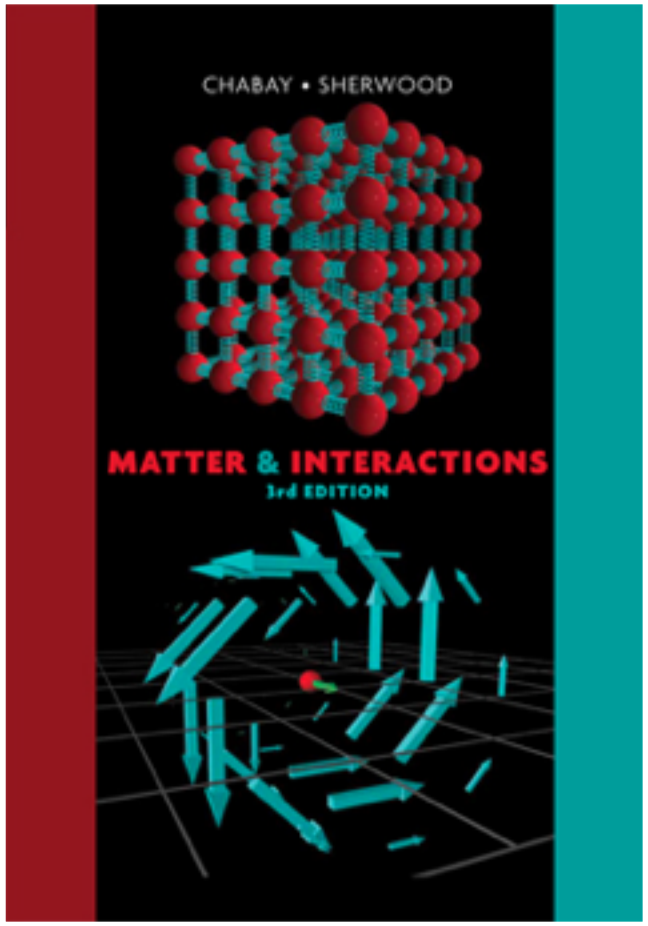
www.matterandinteractions.org

Context:

Calculus-Based Physics

- Matter and Interactions (M&I)
  R. Chabay & B. Sherwood
  Wiley, 2010

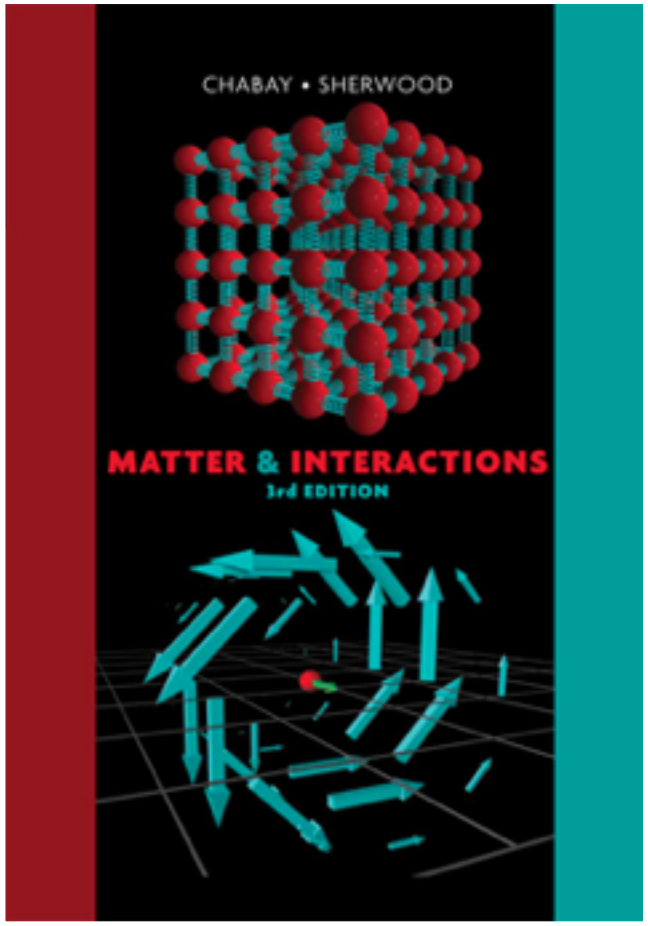
- Computer modeling (VPython)
  as part of laboratory component



www.matterandinteractions.org

Context:

Calculus-Based Physics



- Matter and Interactions (M&I)
  R. Chabay & B. Sherwood
  Wiley, 2010
- Computer modeling (VPython) as part of laboratory component
- Goals: Modeling phenomenon, Developing conceptual understanding, and Identifying important physics in code

www.matterandinteractions.org

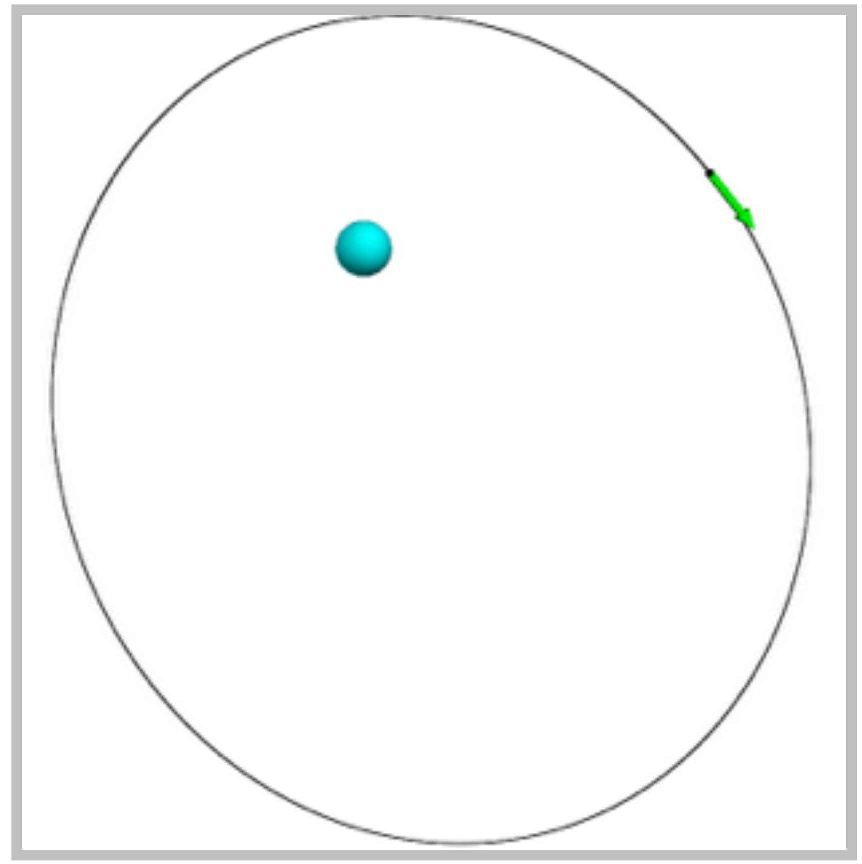
Computational Modeling Homework

Caballero, Kohlmyer, Schatz, PERC Proc. 1413, 15 (2012)
Caballero, Kohlmyer, Schatz, PRST-PER 8, 020106 (2012)

Computational Modeling Homework

Students create a model of an elliptical orbit

```
while t < tf:

        r = craft.pos - Earth.pos
        Fgrav = -G*mEarth*mcraft*r/mag(r)**3

        pcraft = pcraft + Fgrav*deltat
        craft.pos = craft.pos + pcraft/mcraft*deltat

        parr.pos = craft.pos
        parr.axis = pscale*pcraft

        ## check if the spacecraft has crashed into Earth
        ## if so, get out of the calculation
        if mag(r) < Earth.radius:
                break

        trail.append(pos=craft.pos)
        t = t+deltat

print 'Calculations finished after ', t,'seconds.'
```

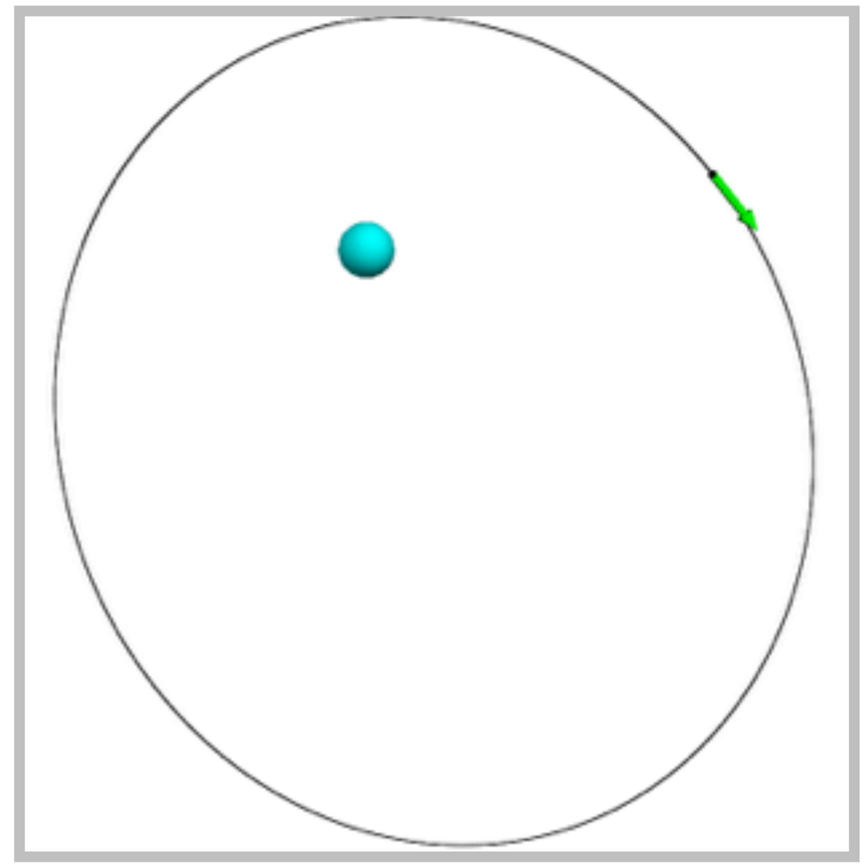


Caballero, Kohlmyer, Schatz, PERC Proc. 1413, 15 (2012)
Caballero, Kohlmyer, Schatz, PRST-PER 8, 020106 (2012)

Computational Modeling Homework

Students create a model of an elliptical orbit

```
while t < tf:

        r = craft.pos - Earth.pos
        Fgrav = -G*mEarth*mcraft*r/mag(r)**3

        pcraft = pcraft + Fgrav*deltat
        craft.pos = craft.pos + pcraft/mcraft*deltat

        parr.pos = craft.pos
        parr.axis = pscale*pcraft

        ## check if the spacecraft has crashed into Earth
        ## if so, get out of the calculation
        if mag(r) < Earth.radius:
                break

        trail.append(pos=craft.pos)
        t = t+deltat

print 'Calculations finished after ', t,'seconds.'
```



Adapt model to solve weekly homework problems

Caballero, Kohlmyer, Schatz, PERC Proc. 1413, 15 (2012)
Caballero, Kohlmyer, Schatz, PRST-PER 8, 020106 (2012)

How proficient are they?

New Model: Central Force

- Assign initial conditions
- Compute force
- Update velocity

```
blueObject = sphere(pos=vector(5,4,0), radius=0.25, color=c
redObject = sphere(pos=vector(-3,-2,0), radius=0.25, color=
trail = curve(color = redObject.color)

k = 0.3
redObject.m = 5e-3

redObject.p = redObject.m*vector(800,800,0)

t = 0
deltat = 5e-6

while t < 1:

    r = redObject.pos - blueObject.pos
    rhat = r/mag(r)
    F = -k/mag(r)**4*rhat

    redObject.p = redObject.p + F*deltat

    redObject.pos = redObject.pos + redObject.p/redObject.m*deltat

    trail.append(pos = redObject.pos)
    t = t + deltat
```

Initial Conditions (identify and assign)

Force Calculation (contextualize)

Newton's Second Law (implement)

Caballero, Kohlmyer, Schatz, PERC Proc. 1413, 15 (2012)
Caballero, Kohlmyer, Schatz, PRST-PER 8, 020106 (2012)

How proficient are they?

New Model: Central Force

- Assign initial conditions
- Compute force
- Update velocity

approx. 1300 students

```
blueObject = sphere(pos=vector(5,4,0), radius=0.25, color=c
redObject = sphere(pos=vector(-3,-2,0), radius=0.25, color=
trail = curve(color = redObject.color)

k = 0.3
redObject.m = 5e-3

redObject.p = redObject.m*vector(800,800,0)

t = 0
deltat = 5e-6

while t < 1:

    r = redObject.pos - blueObject.pos
    rhat = r/mag(r)
    F = -k/mag(r)**4*rhat

    redObject.p = redObject.p + F*deltat

    redObject.pos = redObject.pos + redObject.p/redObject.m*deltat

    trail.append(pos = redObject.pos)
    t = t + deltat
```

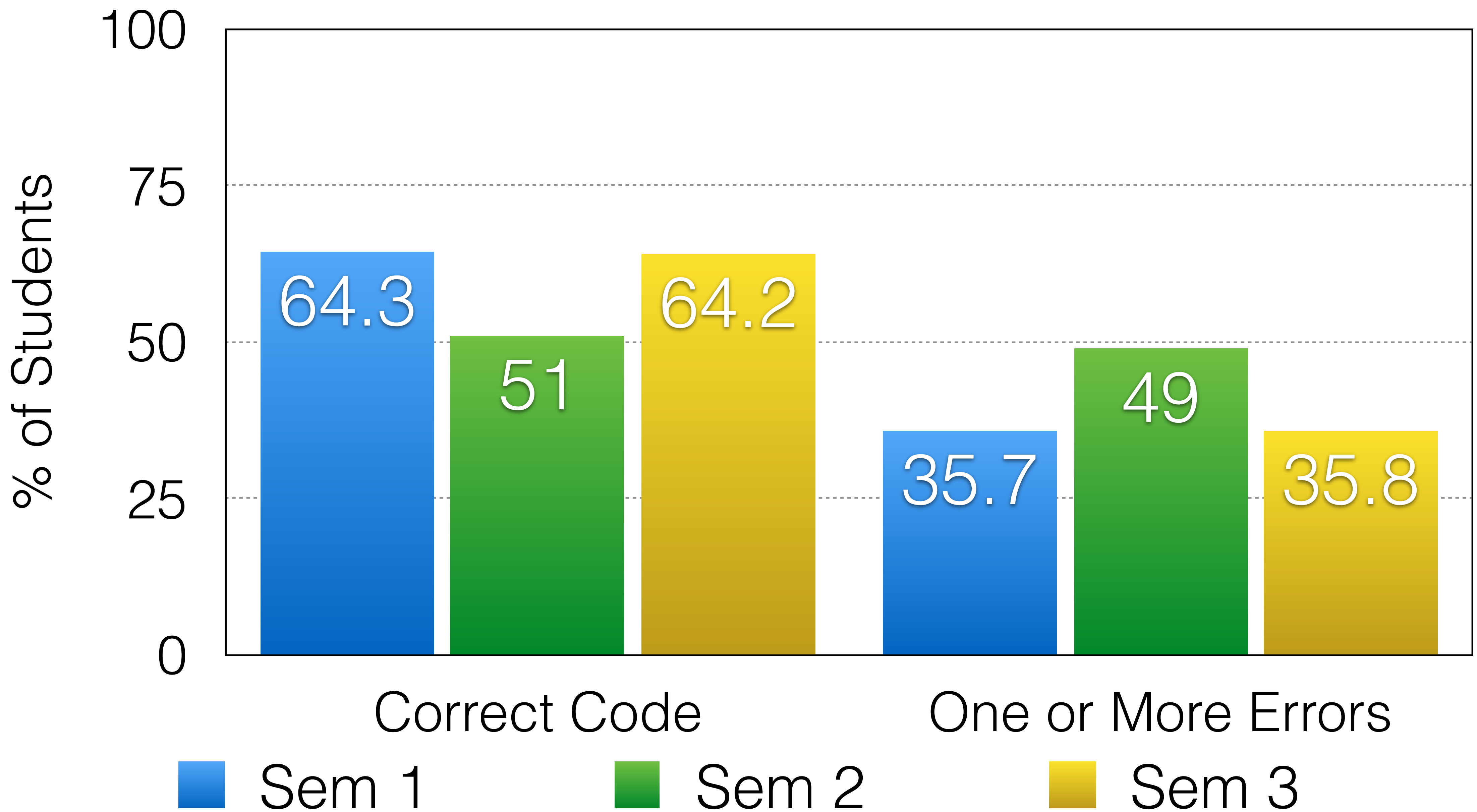
Initial Conditions (identify and assign)

Force Calculation (contextualize)

Newton's Second Law (implement)

Caballero, Kohlmyer, Schatz, PERC Proc. 1413, 15 (2012)
Caballero, Kohlmyer, Schatz, PRST-PER 8, 020106 (2012)

How'd they do?



Caballero, Kohlmyer, Schatz, PERC Proc. 1413, 15 (2012)
Caballero, Kohlmyer, Schatz, PRST-PER 8, 020106 (2012)

Context:

9th Grade Conceptual Physics

Context:

9th Grade Conceptual Physics

Research-based curriculum

Conceptual focus

Model driven

http://modeling.asu.edu

Caballero, Aiken, et al., Phys. Teach., 52, 38 (2014)

Context:

9th Grade Conceptual Physics

AMTA Modeling Instruction™

Research-based curriculum

Conceptual focus

Model driven

http://modeling.asu.edu

Novices easily create graphics

http://www.vpython.org

python™ Visual

Caballero, Aiken, et al., Phys. Teach., 52, 38 (2014)

Context:

9th Grade Conceptual Physics

AMTA Modeling Instruction™

Research-based curriculum

Conceptual focus

Model driven

http://modeling.asu.edu

Novices easily create graphics

http://www.vpython.org

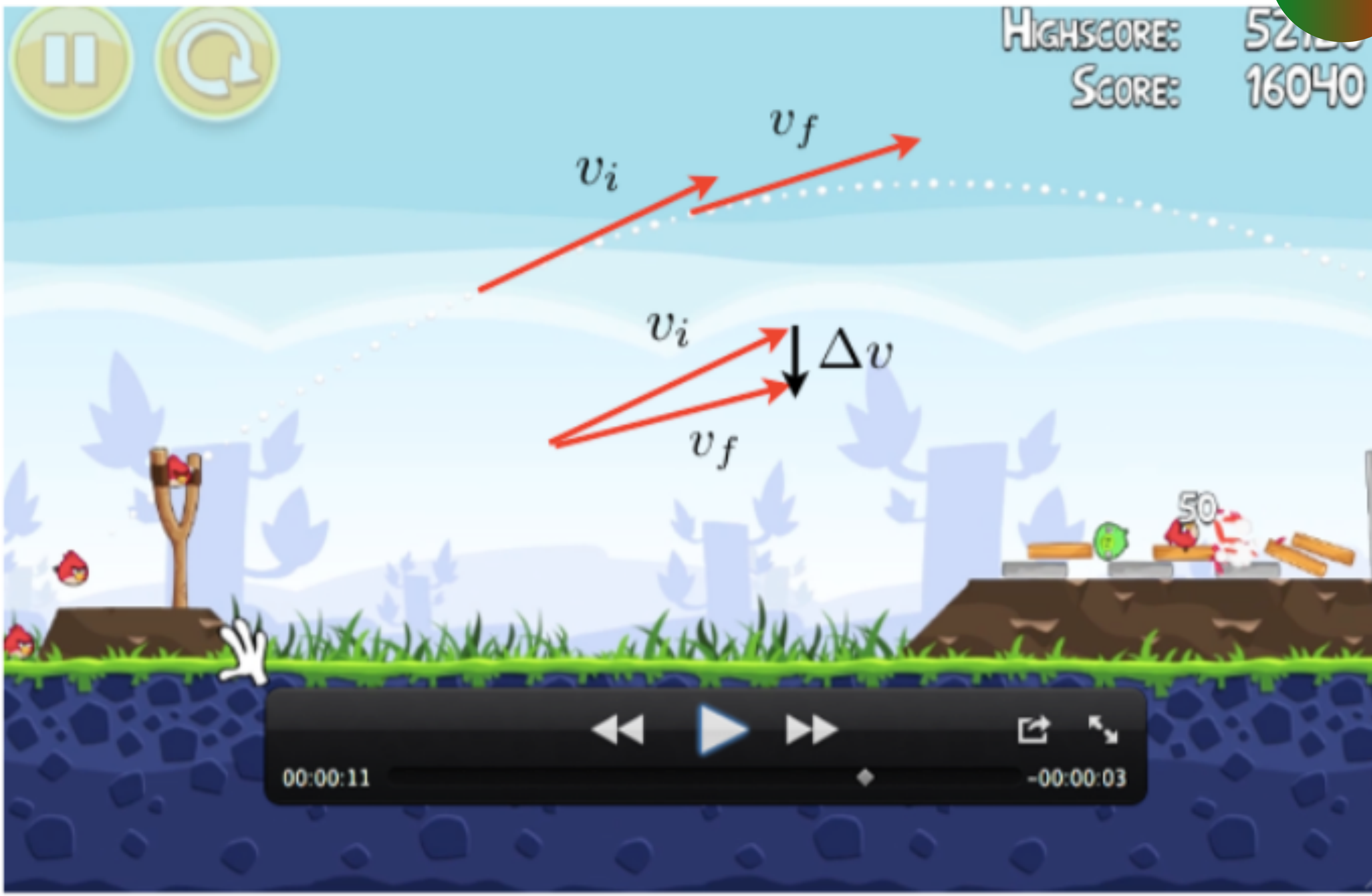
python™ Visual

PhysUtil

Can create graphs, motion maps, axes, timers

https://github.com/perlatmsu/physutil

Caballero, Aiken, et al., Phys. Teach., 52, 38 (2014)



HIGHSCORE:
SCORE: 16040
v_f
v_i
v_i
Δv
v_f
50
00:00:11
-00:00:03

Goals: Identifying important physics in code; Modeling observations

```
trail = curve(color = color.yellow, radius = 1)

# Define axis that marks the field (divide into 15 equal intervals)
#IMPORTANT Question: Why do you need 16 Markers to divide the field into 15 intervals?
axis = PhysAxis(field, 16)


# Define physics parameters
ball.mass=0.6 #mass of ball
ball.velocity = vector(50,0,0) #initial velocity of ball in (vx,vy,vz) form
ball.netForce = vector(0,0,0)
...
time.start = 0 #start time
time.deltaT = 0.001  #time step

# Set timer in top right of screen
time.display = PhysTimer(150,150)

#calculate time for object to cross the field (think carfully about assumptions here)
time.final=field.size.x/ball.velocity.x

# Set up MotionMap to display breadcrumbs
motionMap = MotionMap(ball, time.final, 10, markerType="breadcrumbs",    #drop 10 breadcrumbs between t=0 and time.final
                      labelMarkerOffset=vector(0,-20,0),        #put lables below the marker
                      dropTime=True, timeOffset=vector(0,35,0)) # put times above the marker

# Set up MotionMap to display velocity vectors
velocityMap = MotionMap(ball, time.final,10, markerScale = 0.5,labelMarkerOrder = false)

###END OF SETUP

# MAIN UPDATE LOOP; perform physics updates and drawing
while ball.pos.x < 150:  #while the ball's x-position is less than 150
    # Required to make animation visible / refresh smoothly (keeps program from running faster than 1000 frames/s)
    rate(1000)

    # Ball physics update
    ball.velocity = ball.velocity + ball.netForce/ball.mass*time.deltaT
    ball.pos = ball.pos + ball.velocity*time.deltaT

    # Update motio
    motionMap.updat
```

Caballero, Aiken, et al., Phys. Teach., 52, 38 (2014)

Goals: Identifying important physics in code; Modeling observations

```
trail = curve(color = color.yellow, radius = 1)
```

```
# Define physics parameters
ball.mass=0.6 #mass of ball
ball.velocity = vector(50,0,0)
ball.netForce = vector(25,0,0)
# Define time parameters
...
time.start = 0 #start time
time.deltaT = 0.001  #time step
```

m_{ball} $\vec{v}_{ball}$ $\vec{F}_{net}$ t_0 dt

Initial conditions & model parameters

```
# Set timer in top right of screen
time.display = PhysTimer(150,150)

#calculate time for object to cross the field (think carfully about assumptions here)
time.final=field.size.x/ball.velocity.x

# Set up MotionMap to display breadcrumbs
motionMap = MotionMap(ball, time.final, 10, markerType="breadcrumbs",     #drop 10 breadcrumbs between t=0 and time.final
                      labelMarkerOffset=vector(0,-20,0),           #put lables below the marker
                      dropTime=True, timeOffset=vector(0,35,0)) # put times above the marker

# Set up MotionMap to display velocity vectors
velocityMap = MotionMap(ball, time.final,10, markerScale = 0.5,labelMarkerOrder = false)

###END OF SETUP

# MAIN UPDATE LOOP; perform physics updates and drawing
while ball.pos.x < 150:  #while the ball's x-position is less than 150
    # Required to make animation visible / refresh smoothly (keeps program from running faster than 1000 frames/s)
    rate(1000)

    # Ball physics update
    ball.velocity = ball.velocity + ball.netForce/ball.mass*time.deltaT
    ball.pos = ball.pos + ball.velocity*time.deltaT

    # Update motior
    motionMap.updat
```

Caballero, Aiken, et al., Phys. Teach., 52, 38 (2014)

Goals: Identifying important physics in code; Modeling observations

```
trail = curve(color = color.yellow, radius = 1)
```

```
# Define physics parameters
ball.mass=0.6 #mass of ball
ball.velocity = vector(50,0,0)
ball.netForce = vector(25,0,0)
# Define time parameters
...
time.start = 0 #start time
time.deltaT = 0.001  #time step
```

m_{ball} $\vec{v}_{ball}$ $\vec{F}_{net}$ t_0 dt

Initial conditions & model parameters

```
# Set timer in top right of screen
time.display = PhysTimer(150,150)

#calculate time for object to cross the field (think carfully about assumptions here)
time.final=field.size.x/ball.velocity.x

# Set up MotionMap to display breadcrumbs
motionMap = MotionMap(ball, time.final, 10, markerType="breadcrumbs",    #drop 10 breadcrumbs between t=0 and time.final
```

```
# Ball physics update
ball.velocity = ball.velocity + ball.netForce/ball.mass*time.deltaT
ball.pos = ball.pos + ball.velocity*time.deltaT
```

$$\vec{v}_f = \vec{v}_i + \vec{F}_{net}/m_{ball}\, dt$$

$$\vec{r}_f = \vec{r}_i + \vec{v}_{avg}\, dt$$

Integration Loop (Euler-Cromer Step)

```
    ball.pos = ball.pos + ball.velocity*time.deltaT

    # Update motion
    motionMap.updat
```

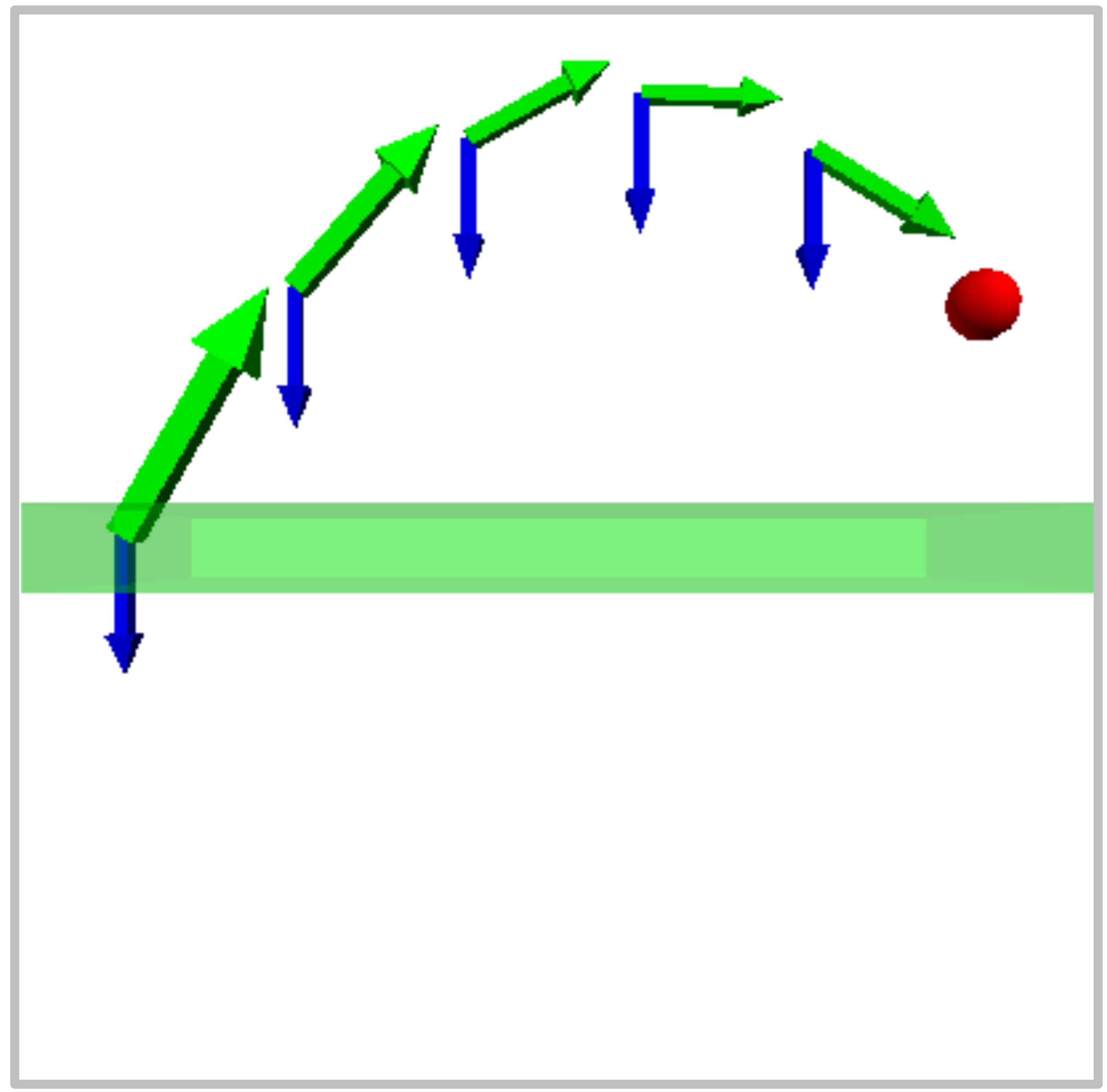
Caballero, Aiken, et al., Phys. Teach., 52, 38 (2014)

Proctored Assignment

Students solved a new problem

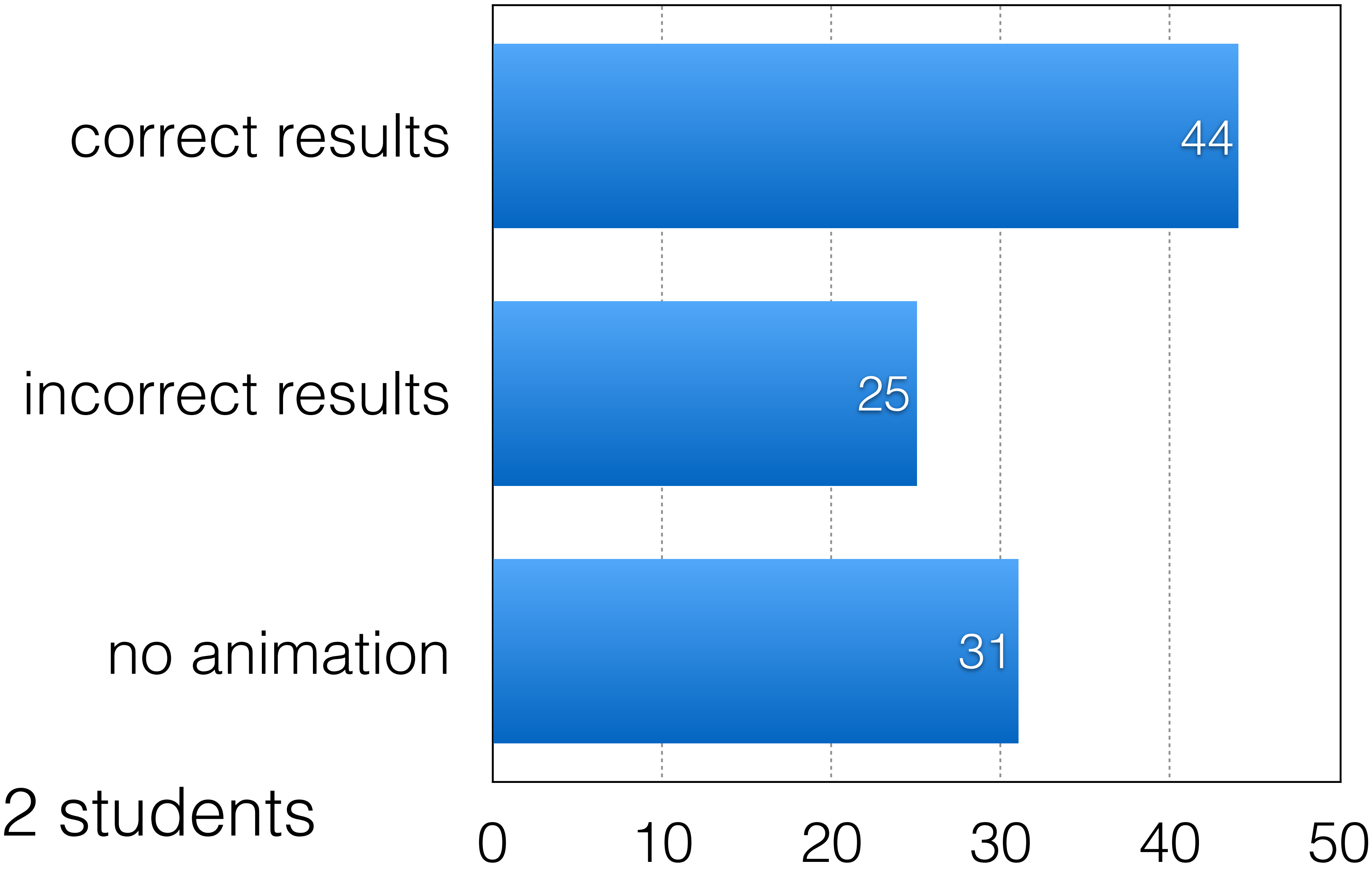
Model a home run

Emphasis
- initial conditions
- numerical integration



Aiken, Caballero, et al., PERC Proc. 1531, 46 (2012)

How'd they do?

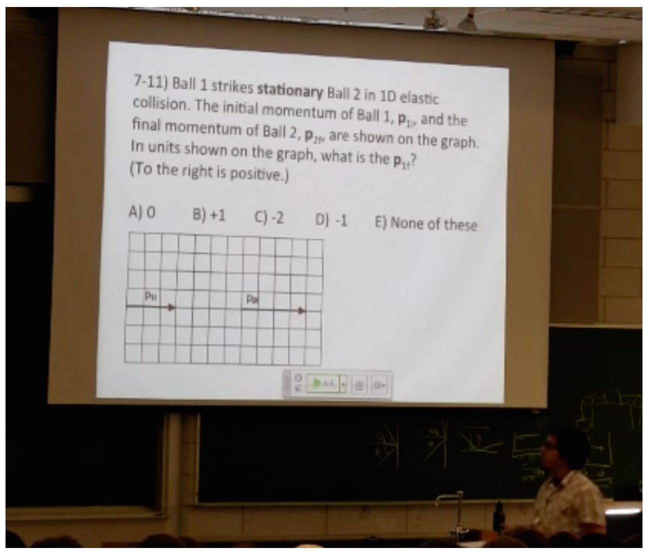


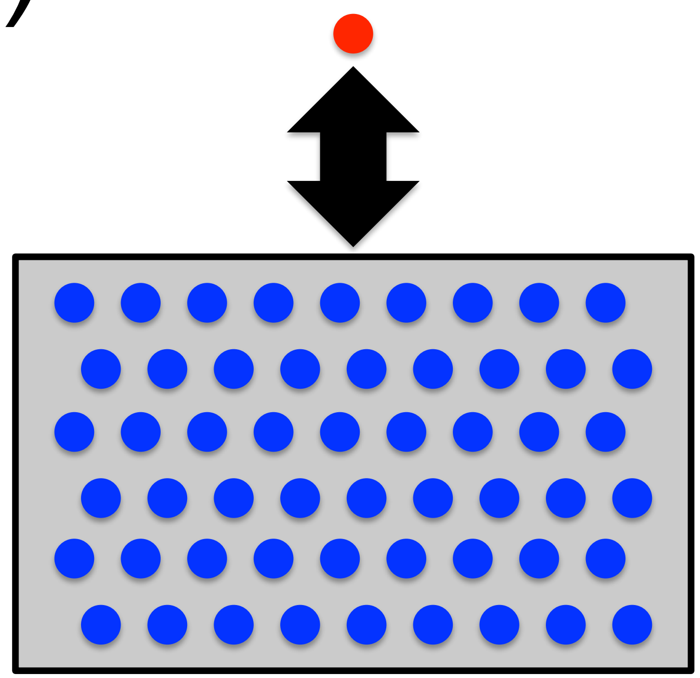
N = 32 students

Aiken, Caballero, et al., PERC Proc. 1531, 46 (2012)

Context:

Projects and Practices in Physics (P^3)

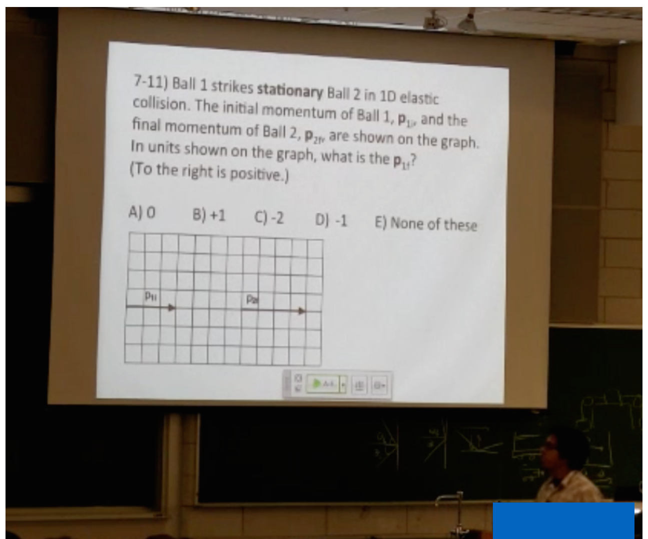


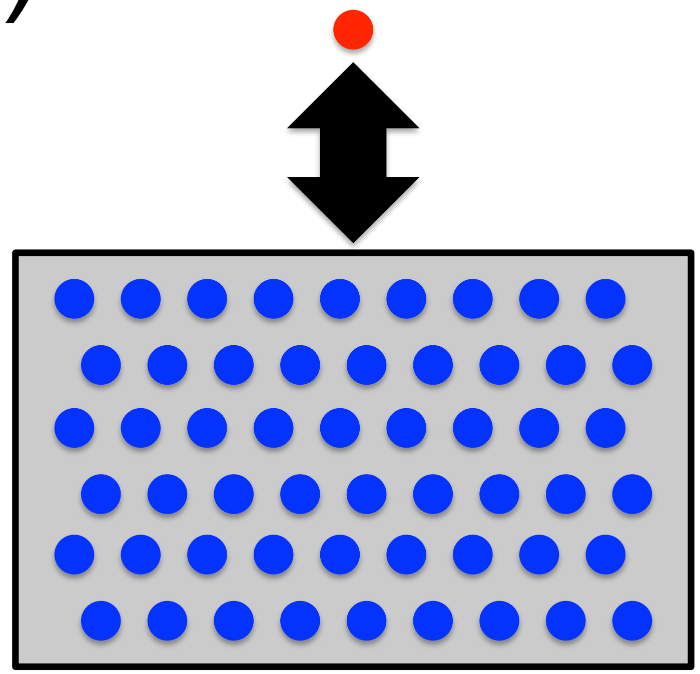


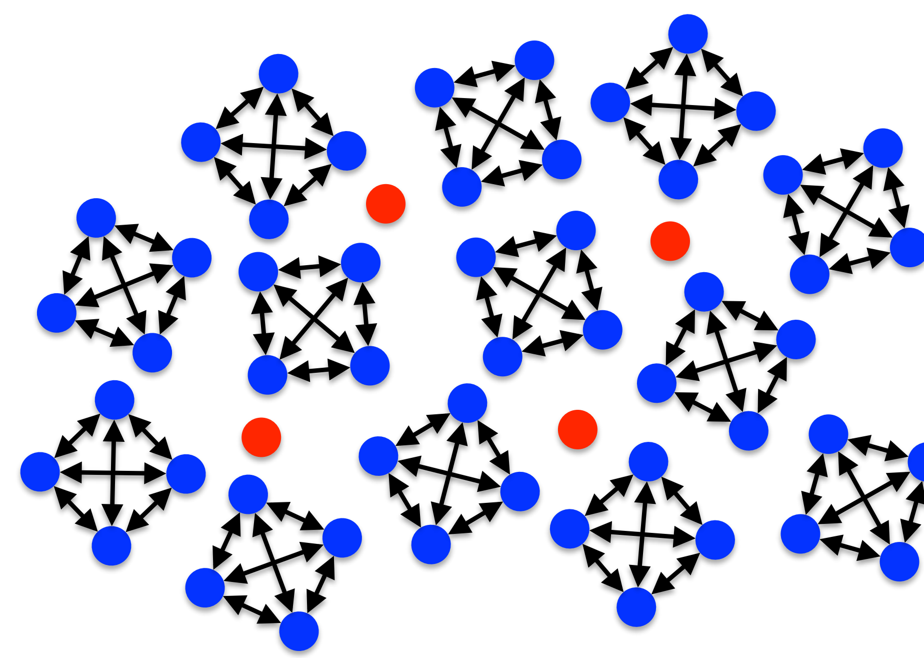
Funded by:

Context:

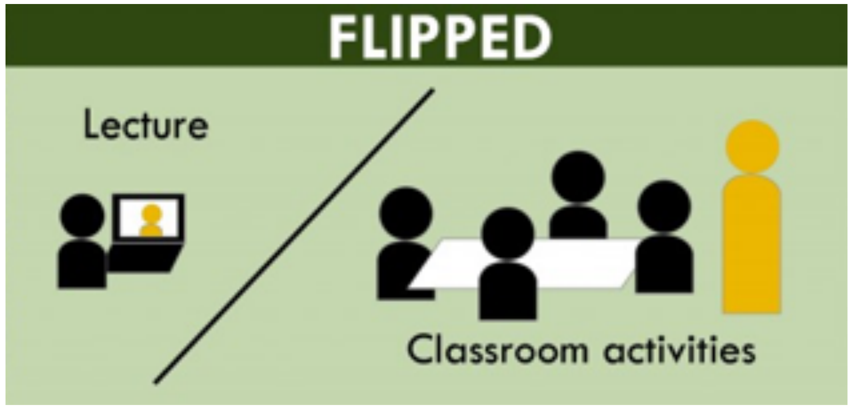
Projects and Practices in Physics (P^3)







Funded by:


FLIPPED
Lecture
Classroom activities

+


Problem Based Learning (PBL)
Problem assigned
Identify what we need to know
Learn and apply it to solve the problem

=

3D Learning Goal:

National Research Council, 2012
Cooper, Caballero, et al., Science, 2015

3D Learning Goal:

Apply the momentum principle iteratively/computationally to predict the motion or determine the properties of motion/ net force acting on a single-particle system where the net force is not constant (e.g., due to spring-like restoring forces or dissipative drag forces).

National Research Council, 2012
Cooper, Caballero, et al., Science, 2015

Marcos Caballero ▾ (Course Coordinator) **PHY183, Fall 2015 - Projects and Practices in Physics** Messages Roles Help Logout

Main Menu | Contents | Course Editor | What's New | Grades ▾ | People ▾ | Settings ▾ | Public ▾ | Switch role ▾

Course Contents » ... » Pre-class HW 2 (Due Sept. 14) » **Modeling Motion of a Fan Cart** Notes Stored Links Evaluate Feedback Print Info

Functions Edit Content Grades Content Settings Edit Folder

Notes and a video lecture on essential Python programming is available on this course notes page.

Consider the following VPython script that was written to model the motion of a fan cart.
You may assume the script uses SI units.

```
from visual import *

fancart = sphere(pos=vector(16.8,16.7,0), radius=0.5, color=color.red)

mcar = 0.5
vcart = vector(7.05,0,0)
pcart = mcart*vcart

Ffan = vector(3,0,0)

t = 0
dt = 0.1
tf = 0.4

while t<tf:
        rate(150)

        pcart = pcart + Ffan*dt
        fancart.pos = fancart.pos + (pfancart/mcart)*dt

        t = t + dt
```

Randomized VPython Code

(a) At what location is the fan cart when the script starts?

$\vec{r}_i = \left\langle \mathbf{16.8000}, \mathbf{16.7000}, \mathbf{0} \right\rangle$ m

Computer's answer now shown above. Tries 0/5

(b) What is the velocity of the fan cart when the script starts?

$\vec{v}_i = \left\langle \mathbf{7.0500}, \mathbf{0}, \mathbf{0} \right\rangle$ m/s

Computer's answer now shown above. Tries 0/5

(c) At what time does the script stop running?

$t_f = \mathbf{0.4000}$ s

Computer's answer now shown above. Tries 0/5

(d) What is the momentum of the fan cart when the script stops running?

$\vec{p}_f = \left\langle \mathbf{4.7250}, \mathbf{0}, \mathbf{0} \right\rangle$ kg*m/s

Computer's answer now shown above. Tries 0/5

Identify and Interpret VPython Scripts

Project 3: Geosynchronus Orbit: Part A

The Carver Media Group is planning the launch of a new communications satellite. Elliot Carver (head of Carver Media Group) is concerned about the launch. This is a $200,000,000 endeavor. In particular, he is worried about the orbital speed necessary to maintain the satellite's geosynchronus orbit (and if that depends on the launch mass). You were hired as an engineer on the launch team. Carver has asked that you allay his concerns.

Project 3: Geosynchronus Orbit: Part B

Carver is impressed with your work, but remains unconvinced by your predictions. He has asked you to write a simulation that models the orbi 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.

Code for Project 3:
geosync.py
PhysUtil Module

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
```

Weatherford, PhD Thesis (2011)

AFTER

```
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
```

Weekly Feedback

<u>"Liam's" Feedback:</u>

Your group did have a tendency to jump quickly into a plan of action without evaluating whether it is the right plan or not and although it worked out for your group this time going forward it might not be as successful with the other problems. Going forward we would like to see you take on a more guiding role in your group…[w]e know this might sound like it might be a slower process…[w]e expect you to encourage another group member to take the lead when working with Python.

Irving, Sawtelle, Caballero 2015 PERC

Weekly Feedback

<u>"Liam's" Feedback:</u>

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.

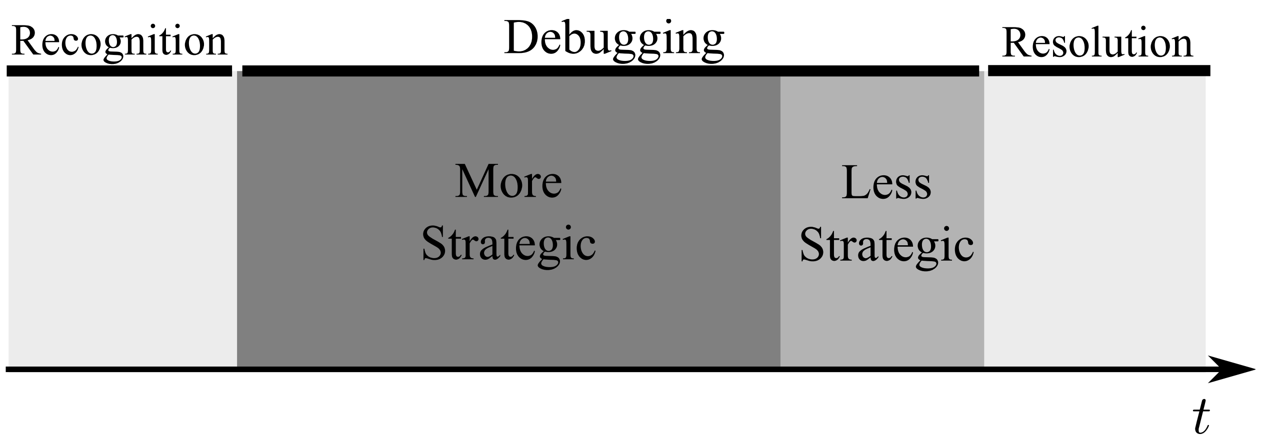
Irving, Sawtelle, Caballero 2015 PERC

Liam given "just-in-time" feedback

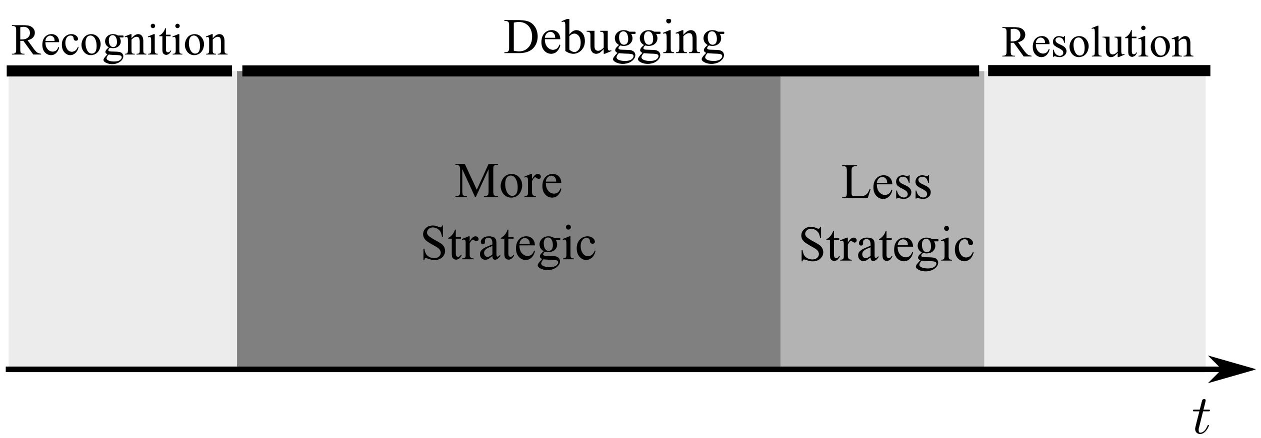
Liam given “just-in-time” feedback

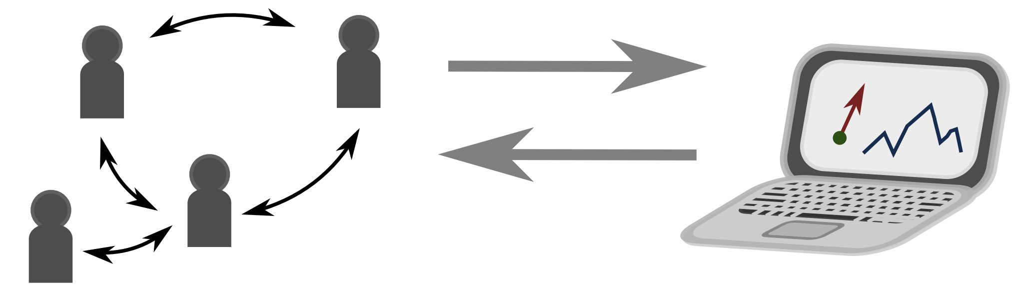
Computational Physics Education Research

Computational Physics Education Research

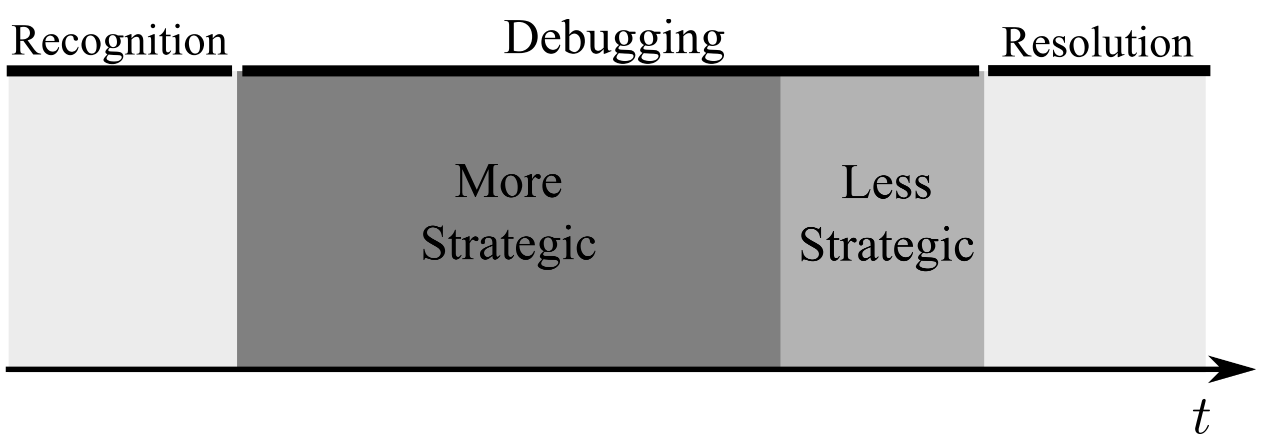


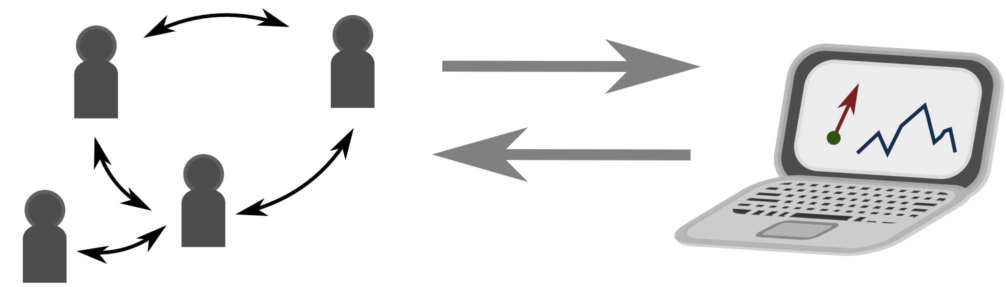
Computational Physics Education Research

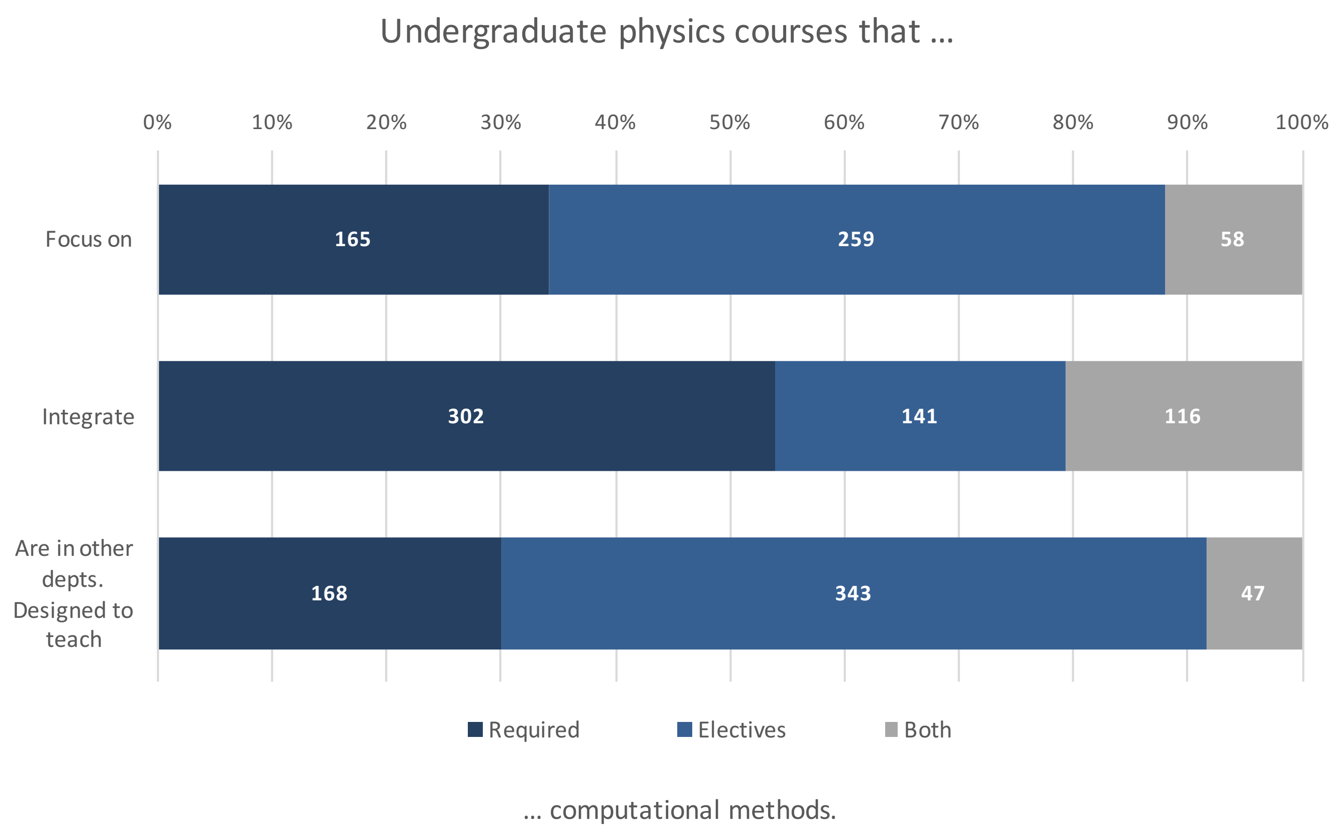




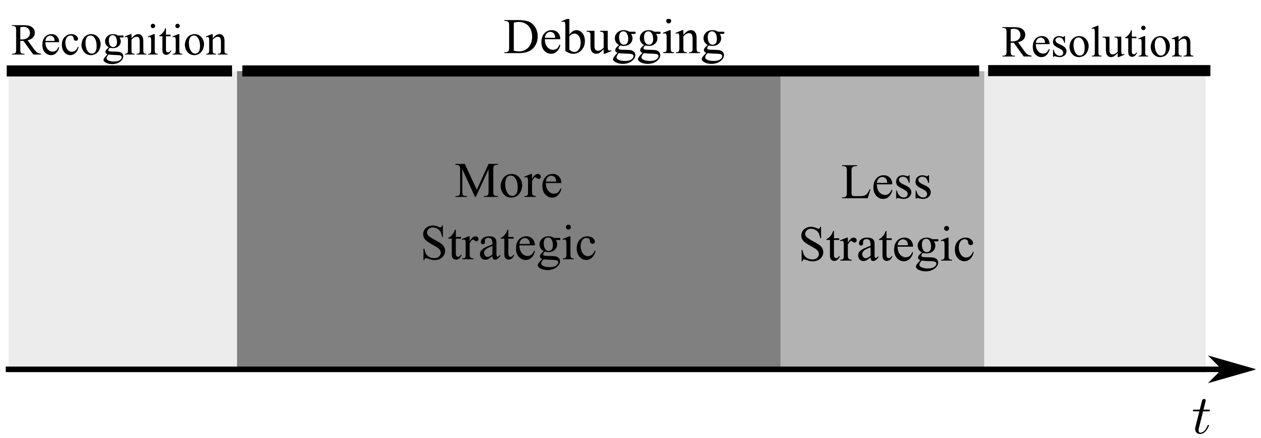
Computational Physics Education Research

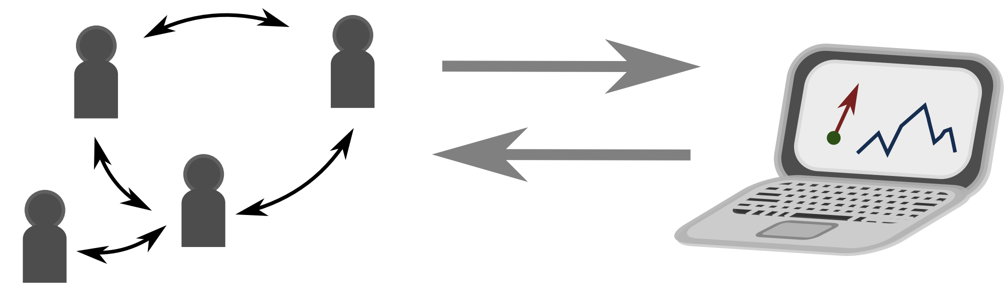


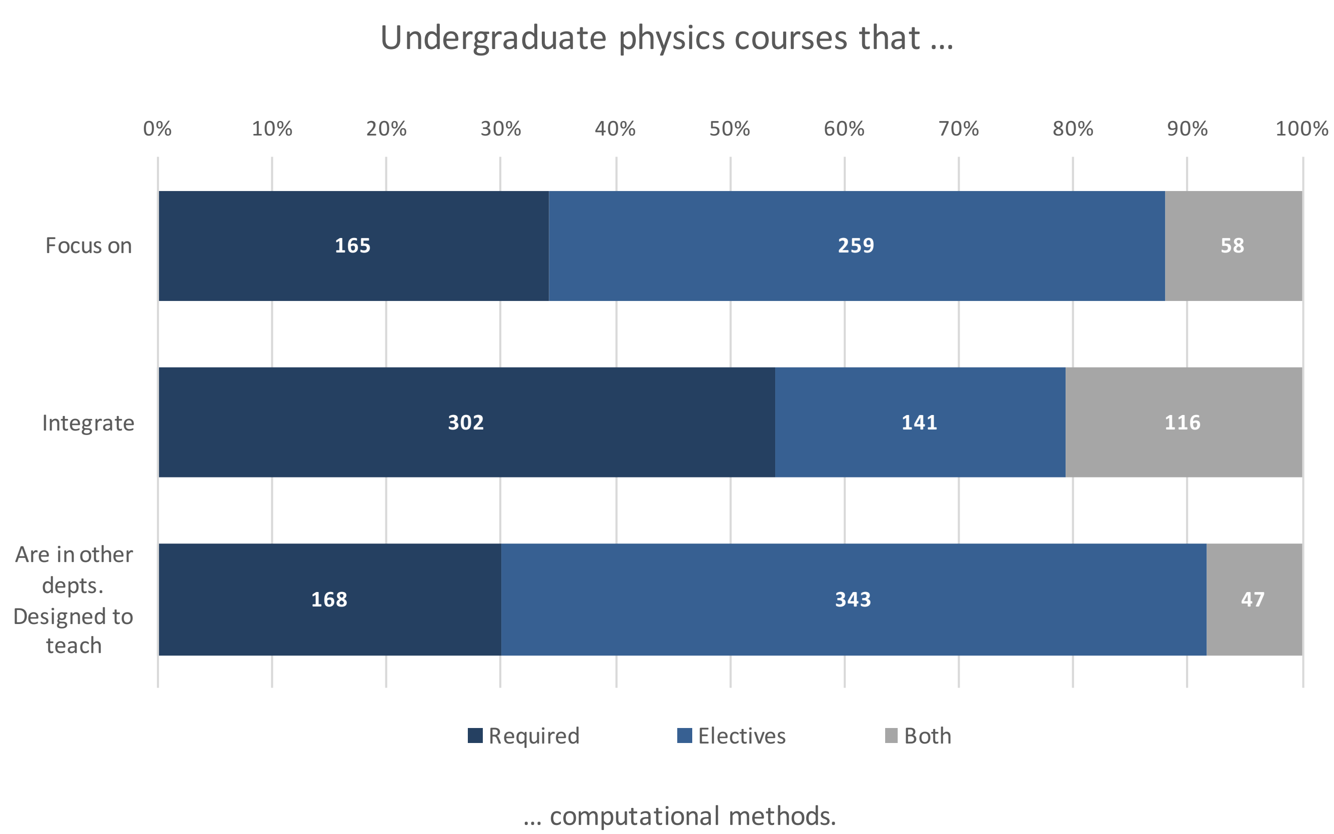


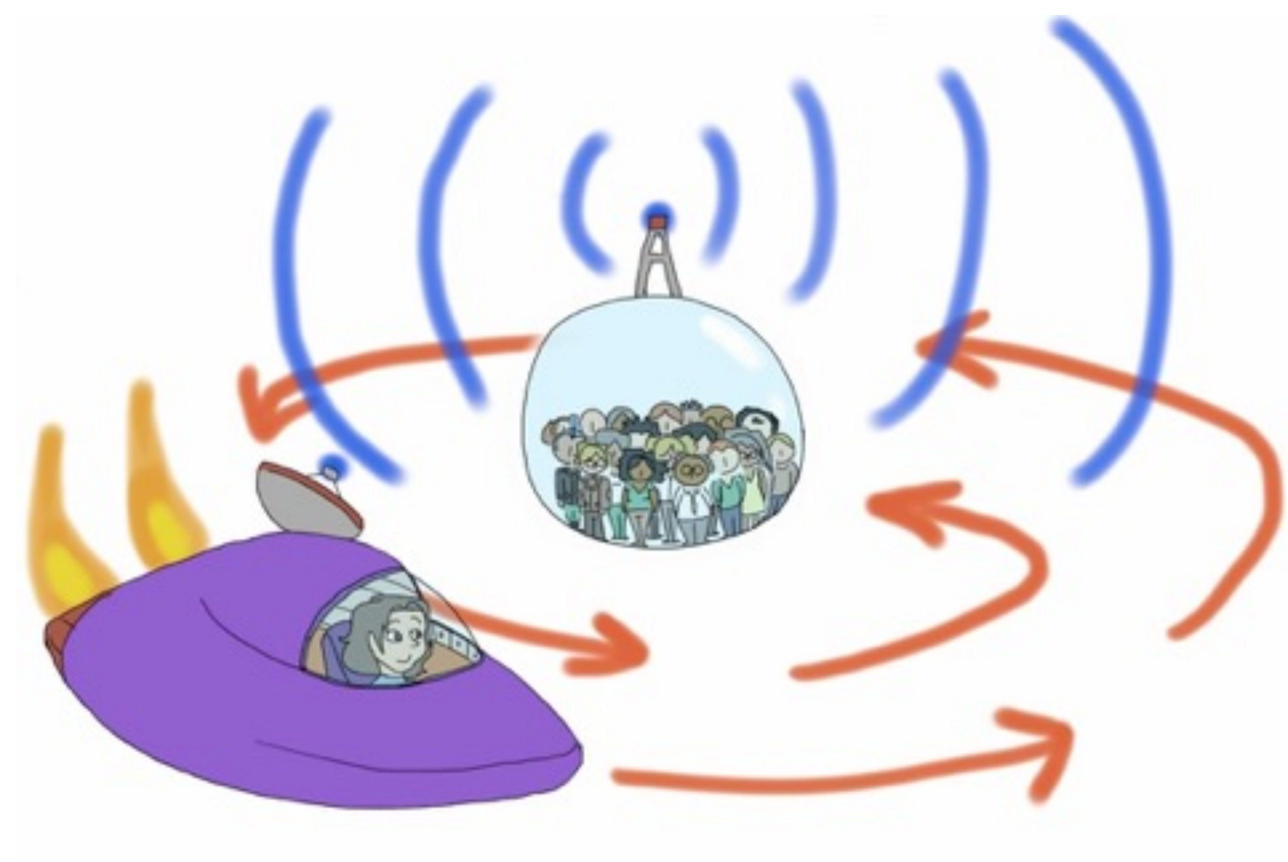


Computational Physics Education Research









How can I get started?

Supporting your students

Supporting your students

- Consider your goals
    - One size does not fit all. What do you want your students to be able to do?

Supporting your students

- Consider your goals
  - One size does not fit all. What do you want your students to be able to do?
- Consider your students' background
  - Students will benefit from scaffolding. What kind of prior experiences have your students had?

Supporting your students

- Consider your goals
  - One size does not fit all. What do you want your students to be able to do?
- Consider your students' background
  - Students will benefit from scaffolding. What kind of prior experiences have your students had?
- Consider your students' interests
  - Motivating students to do something new can be a challenge. What are your students interested in doing?

Supporting your students

- Consider your goals
    - One size does not fit all. What do you want your students to be able to do?
- Consider your students' background
    - Students will benefit from scaffolding. What kind of prior experiences have your students had?
- Consider your students' interests
    - Motivating students to do something new can be a challenge. What are your students interested in doing?
- Start small and iterate
    - Change is a marathon not a sprint. Collect data!

Interested in learning even more?

PICUP PARTNERSHIP FOR INTEGRATION OF COMPUTATION INTO UNDERGRADUATE PHYSICS

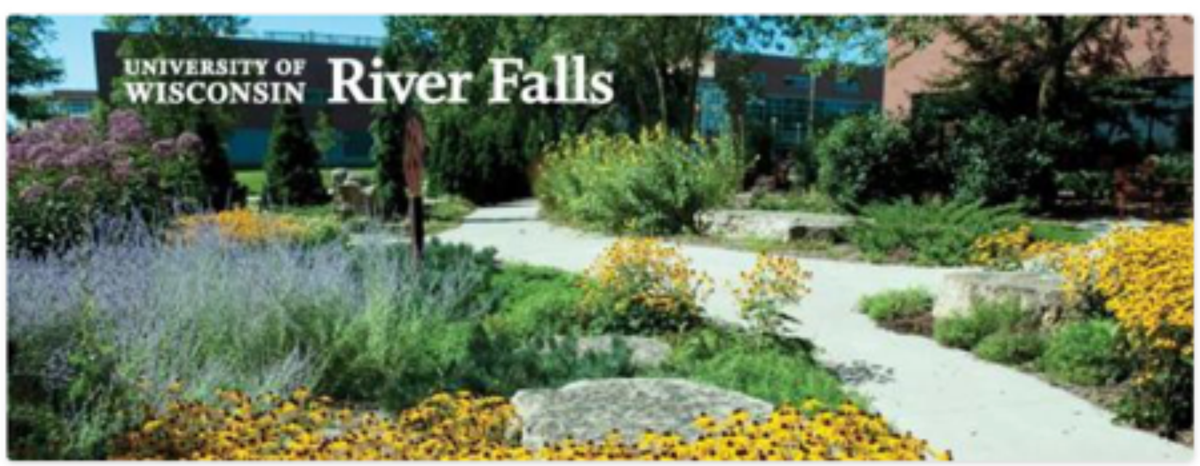
NSF

2016 PICUP Summer Faculty Development Workshop at UW River Falls

Energize Undergraduate Physics Courses with Computation

Fill out and submit a 2016 PICUP Summer Faculty Development Workshop Application

Dates: August 1-5, 2016; Participants should plan to arrive Sunday, July 31
Location: University of Wisconsin at River Falls campus, River Falls, Wisconsin



gopicup.org

caballero@pa.msu.edu

Who should apply: All physics faculty, especially those who have not yet attempted to integrate computation into their undergraduate physics courses.

In this workshop participants, with the assistance of the workshop coordinators, will develop a viable, personalized plan for integrating computation into their undergraduate physics course(s) to be implemented in the upcoming academic term.