

USING PHYSLETS AND OPEN SOURCE PHYSICS (OSP) TO TEACH PHYSICS & ASTRONOMY

OSP COMPADRE COLLECTION: WWW.COMPADRE.ORG/OSP/
PHYSLETS ORG: [HTTPS://PHYSLETS.ORG/PP/](https://PHYSLETS.ORG/PP/)

Mario Belloni and Wolfgang Christian

Davidson College

**Doug Brown, Anne Cox, Francisco Esquembre, Harvey Gould,
Robert Hanson, Bruce Mason, Todd Timberlake, Jan Tobochnik,
and Kristen Thompson**

**This work is supported by the National Science Foundation
(DUE-0442581, DUE-0126439, and DUE-9752365)**



Today

- Brief description/demo of tools.
- Explore *Physlet Physics 3E* demo site/materials.
- Explore OSP site/materials.
- Discussion of how to use.

What We Do

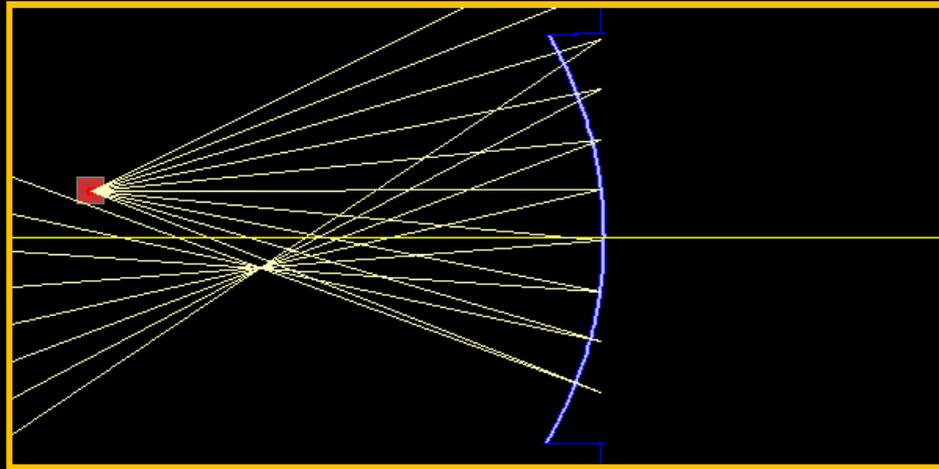
The **Physlet** and **Open Source Physics** Projects under the umbrella of “OSP” provide **curriculum**, **curricular resources**, and **tools** designed to actively engage students in:

- ▣ learning astronomy & physics
- ▣ computer modeling and video analysis
 - ▣ computational physics

with the goal of giving students new ways to understand, describe, explain, and predict physical phenomena.

All of our materials are freely available on the **OSP Collection in ComPADRE**.

But first, a demo...



A point source is located to the left of a mirror. You can *click-drag* the point source to any position (**position is given in centimeters**).

1. Find the focal length of the mirror.
2. **Describe the technique(s) you used to determine the focal length.**

Problem 33.1:

<https://physlets.org/pp/optics/mirrors/default.html>

Simulation Terminology 101

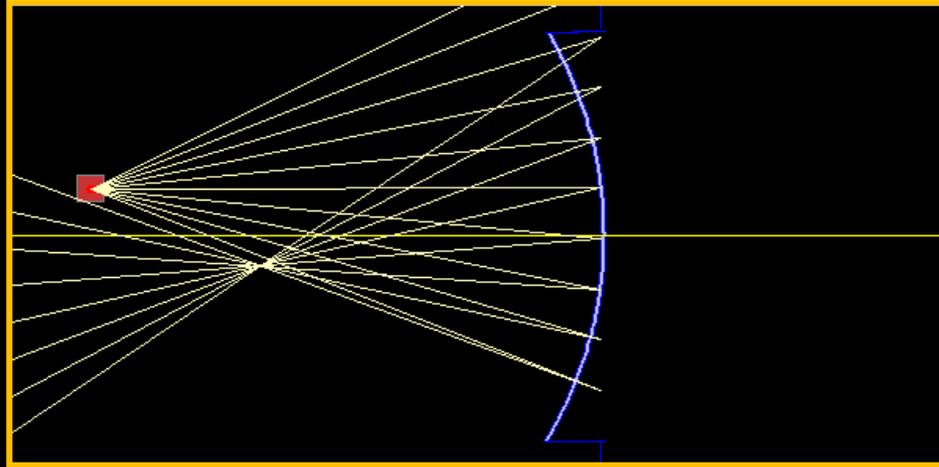
- ▣ **Java:** Programming language for simulations. Simulations can be run as stand-alone archives (jars) or as applets. **Caution: Java applets are deprecated and only work in small number of browsers. All desktop Java applications (jar files) continue to function if Java the JRE is installed.**
- ▣ **JavaScript:** Scripting language created in 1995 and is part of the HTML5 standard. Recently updated to European Computer Manufacturers Association (ECMA) Script 6.
- ▣ **HTML5:** Markup language standard for the Web going forward. Supports JavaScript, CSS, etc. Does not support Java Applets or Flash plug-ins.

Our Approach

The **Physlet** and **Open Source Physics** Projects are under the umbrella of “**OSP**” provide **curriculum, curricular resources, and tools** designed to actively engage students with the goal of providing students with new ways to understand, describe, explain, and predict physical phenomena.

- One Java application....one Java simulation.....one idea/one concept.
- One HTML pageone JavaScript simulation.....one idea/one concept.
- One eBook pageone JavaScript simulation.....one idea/one concept.

5 minutes to explore Optics...



...thinking about the material available, how it behaves, how you and your students might use, etc.

Problem 33.1:

<https://physlets.org/pp/optics/mirrors/default.html>

How can we use simulations in teaching physics?

- ▣ **User:** Students access pre-made simulations that (hopefully) they must interact with.
- ▣ **Modeler:** Students are given access to a software package with a simple user interface. Students must then simulate the physics of a problem by modeling at a high level of abstraction. For example, adding the physics in the form of differential equations (rates of change) and initial conditions.
- ▣ **Programmer:** Students are given tools to program a physics example using traditional computational physics techniques.

The 3 OSP Resource Types

Open Source Physics: www.compadre.org/OSP/

U

- *Physlet Physics 2E*: www.compadre.org/Physlets
- *Physlet Quantum Physics 2E*: www.compadre.org/PQP

U

- *Physlet Physics 3E*:
<https://physlets.org/pp/optics/>

M

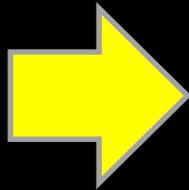
- Tracker Video Analysis Tool:
www.cabrillo.edu/~dbrown/tracker/

M

- Easy Java/JavaScript Simulations (EJS/EjsS):
fem.um.es/Ejs/

P

How can we use simulations in teaching physics? Materials?



User: Students access pre-made simulations that (hopefully) they must interact with.

- ▣ **Modeler:** Students are given access to a software package with a simple user interface. Students must then simulate the physics of a problem by modeling at a high level of abstraction. For example, adding the physics in the form of differential equations (rates of change) and initial conditions.
- ▣ **Programmer:** Students are given tools to program a physics example using traditional computational physics techniques.

THE PAST 20 YEARS: 1,000 PHYSLET-BASED EXERCISES ON COMPADRE FOR INTRO (800) & MODERN AND QM (200)

Physlets, Physics applets, *were* small,
flexible Java applets that *were* ...

Visual and Interactive
Uniform in User Interface
Pedagogically Adaptable
Web based with **Java-enabled browser**
Free for noncommercial use.

Physlet Physics 2E

The screenshot shows the homepage for Physlet Physics 2E. At the top, there is a navigation menu with links for I.Mechanics, II.Fields, III.Waves, IV.Thermodynamics, V.Electromagnetism, VI.Consults, and VII.Optics. The main content area is divided into several sections: a welcome message stating that Physlets run in a Java-enabled browser; a list of topics including Mechanics, Waves, Thermodynamics, Electromagnetism, Circuits, and Optics; a preface section with links for Preface, System Requirements, Java Security and Browser Settings, Credits, and Conditions of Use; and a section for downloading eBooks, with buttons for Google Play and iBooks. On the right side, there is a large graphic for the 'PHYSLET PHYSICS 2E JAVA EDITION' by Wolfgang Christian and Mario Belloni, featuring a colorful, abstract circular pattern. At the bottom, there is a footer with the OSP Network logo and copyright information for 2013.

Physlet Quantum Physics 2E

The screenshot shows the homepage for Physlet Quantum Physics 2E. At the top, there is a navigation menu with links for 0.Preface, I.Introduction, II.Special Relativity, III.Need for a Quantum Theory, IV.Quantum Theory, V.Applications, and Bibliography. The main content area is divided into several sections: a welcome message stating that Physlets run in a Java-enabled browser; a list of topics including Special Relativity, The Need for a Quantum Theory, Quantum Theory, and Applications; a preface section with links for Preface, System Requirements, Java Security and Browser Settings, Credits, and Conditions of Use; and a section for downloading the eBook, with a button for iBooks. On the right side, there is a large graphic for 'PHYSLET QUANTUM PHYSICS 2E AN INTERACTIVE INTRODUCTION' by Mario Belloni, Wolfgang Christian, and Anne J. Cox, featuring a colorful, abstract circular pattern. At the bottom, there is a footer with the OSP Network logo and copyright information for 2014.

www.compadre.org/Physlets

www.compadre.org/PQP

SOON*: 1,000

JAVASCRIPT PHYSLET-BASED EXERCISES ON COMPADRE FOR INTRO (800) & MODERN AND QM (200)

JS Physlets, Physics applets, *are* small,
flexible JavaScript simulations that *are* ...

Visual and Interactive

Uniform in User Interface

Pedagogically Adaptable

Web based w/ JavaScript-enabled browser

Free for noncommercial use.

PRESENT: Converting these 1,000 pieces of curricular
material (Illustrations, Explorations, and Problems)...
to JavaScript to run everywhere:

<https://physlets.org/pp/>

Currently 127 Optics + 130 Fluids and E&M simulations
and growing (fast).

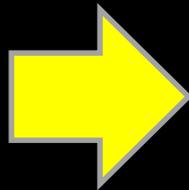
When completed, the 3E of Physlet books will be on
ComPADRE.

Coupling Simulation with Pedagogy

- ▣ **User:** Students access pre-made simulations that they must interact with.
- ▣ **Pedagogies:**
 - Peer Instruction / Think-Pair-Share / Clickers
 - Just-in-Time Teaching
 - Guided Inquiry/Tutorial
 - Group Problem Solving
 - Lecture Demonstration
 - TIPERs (Ranking Tasks, etc)
 - In-class Exercise
 - Homework
 - Laboratory Exercises (pre-lab, in-lab, post-lab)
 - Etc.

How can we use simulations in teaching physics?

- ▣ **User:** Students access pre-made simulations that (hopefully) they must interact with.



Modeler: Students are given access to a software package with a simple user interface. Students must then simulate the physics of a problem by modeling at a high level of abstraction. For example, adding the physics in the form of differential equations (rates of change) and initial conditions.

- ▣ **Programmer:** Students are given tools to program a physics example using traditional computational physics techniques.

Tracker Video Analysis and Modeling Tool

The screenshot displays the Tracker software interface with several key components highlighted by yellow callouts:

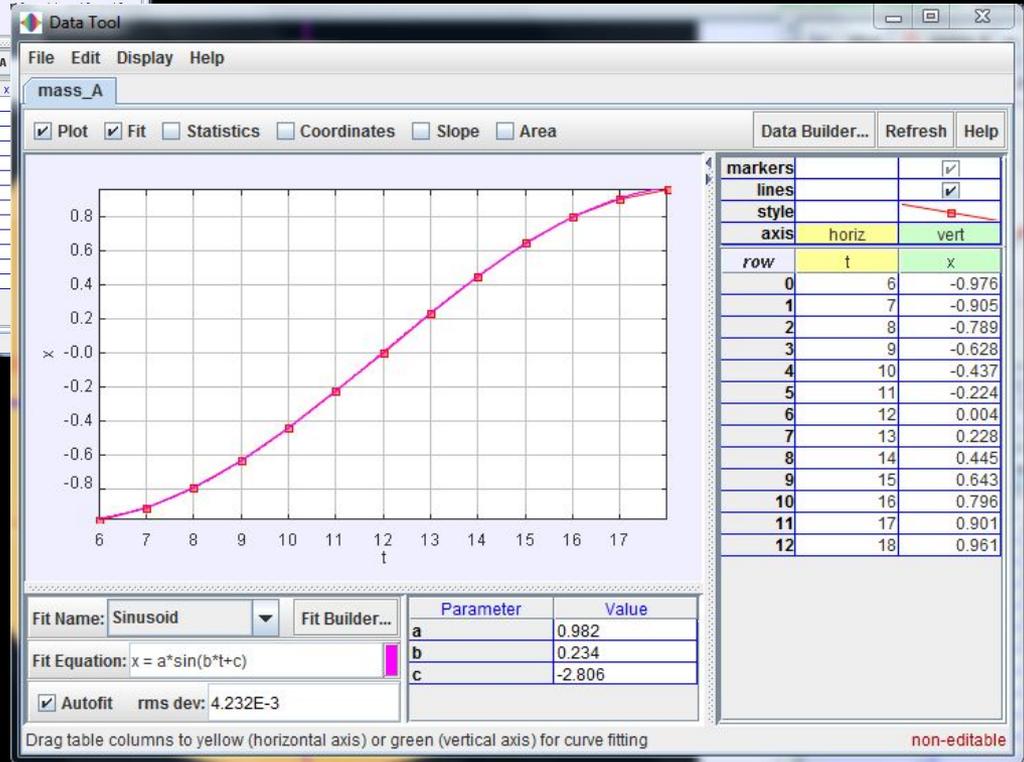
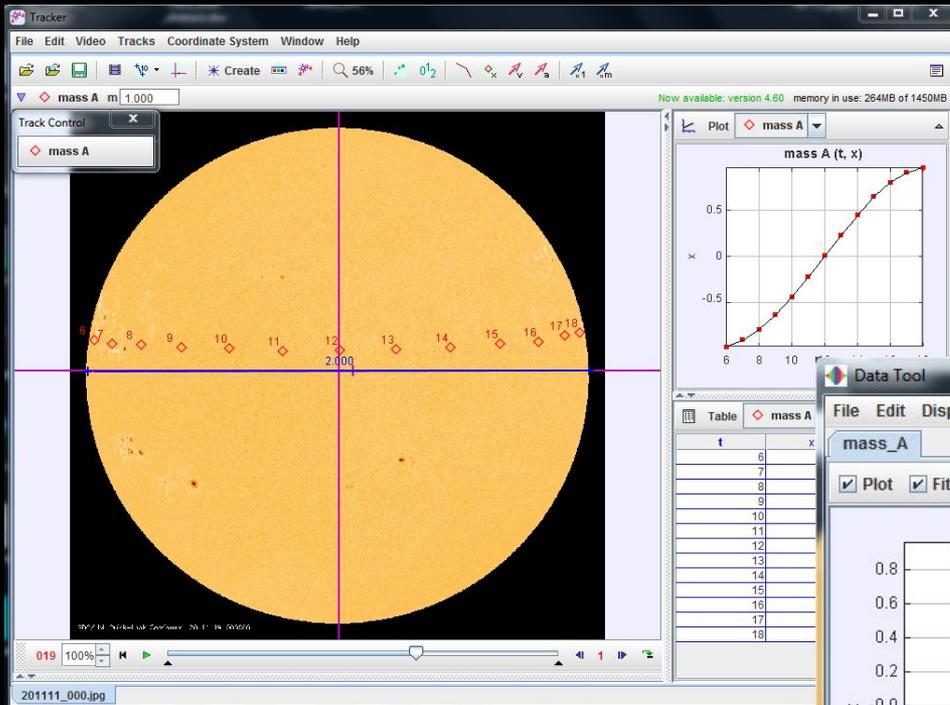
- Set Tracks and Parameters:** A callout pointing to the top menu bar and the 'star B' track control panel.
- Image or Video:** A callout pointing to the central video frame showing a star's path.
- Play and set time:** A callout pointing to the playback controls at the bottom of the video frame.
- Graph Data:** A callout pointing to the 'Plot' window showing a graph of star A's position over time.
- Data Table:** A callout pointing to the 'Table' window showing a table of data for star A.

The 'Table' window displays the following data for star A:

t	ω
0	0.0043064
601	0.0042736
1,202	0.0041936
1,803	0.0041336
2,404	0.0041596
3,005	0.0041936
3,606	0.0041986
4,207	0.0041336
4,808	0.0041986
5,409	0.0041336
6,010	0.0041596
6,611	0.0041936

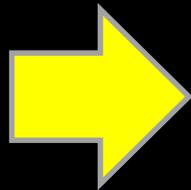
OSP-based free and open source (hmm..) video analysis program. On the OSP Collection there are numerous examples from Angry Birds to solar rotation rates, plus students can take their own videos. Rhett Allain often uses Tracker as part of his WIRED blog.

Tracker: Solar Rotation

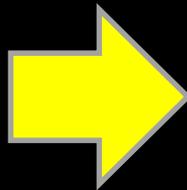


How can we use simulations in teaching physics?

- ▣ **User:** Students access pre-made simulations that (hopefully) they must interact with.



Modeler: Students are given access to a software package with a simple user interface. Students must then simulate the physics of a problem by modeling at a high level of abstraction. For example, adding the physics in the form of differential equations (rates of change) and initial conditions.



Programmer: Students are given tools to program a physics example using traditional computational physics techniques.

EJS/EjsS

Creating Java and JavaScript Simulations

To create Java and JavaScript (HTML5 compliant) simulations we use the EJS/EjsS Tool

- EjsS creates both Java and JS simulations
- To see examples (>500) of stand-alone Java simulations: www.compadre.org/osp search “...”
- To see examples (>200) of JS simulations: www.compadre.org/osp search “JS” for JavaScript

Creating an EJS Simulation

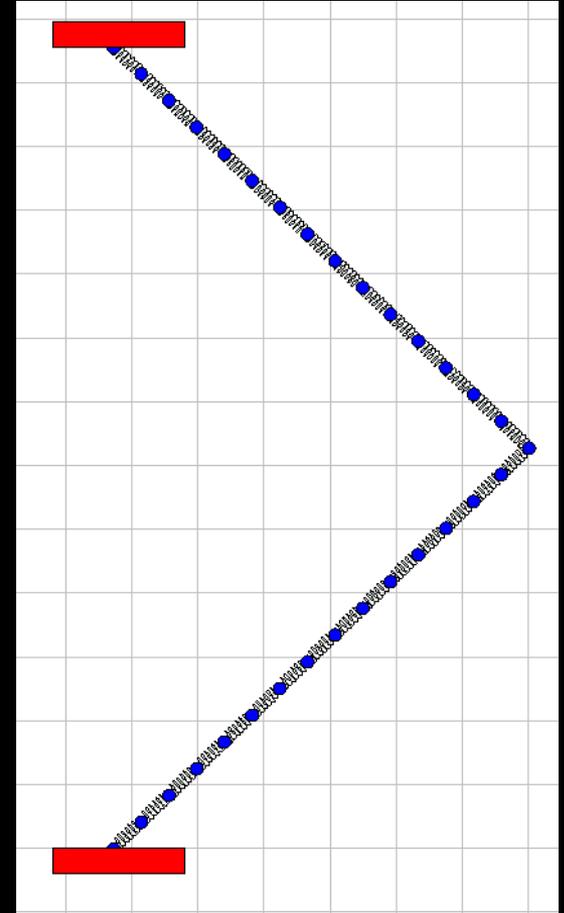
Description			
Model			
Variables			
Dynamic Vars			
Constants			
Name	Initial value	Type	
x	0.0	double	[n]
dx	$2 * \text{Math.PI} / (n-1) ;$	double	
k	1	double	

Oscillator ODE	
Indep. Var.	Increment
t	dt
State	Rate
$\frac{d y[]}{d t} =$	v
$\frac{d v[]}{d t} =$	a

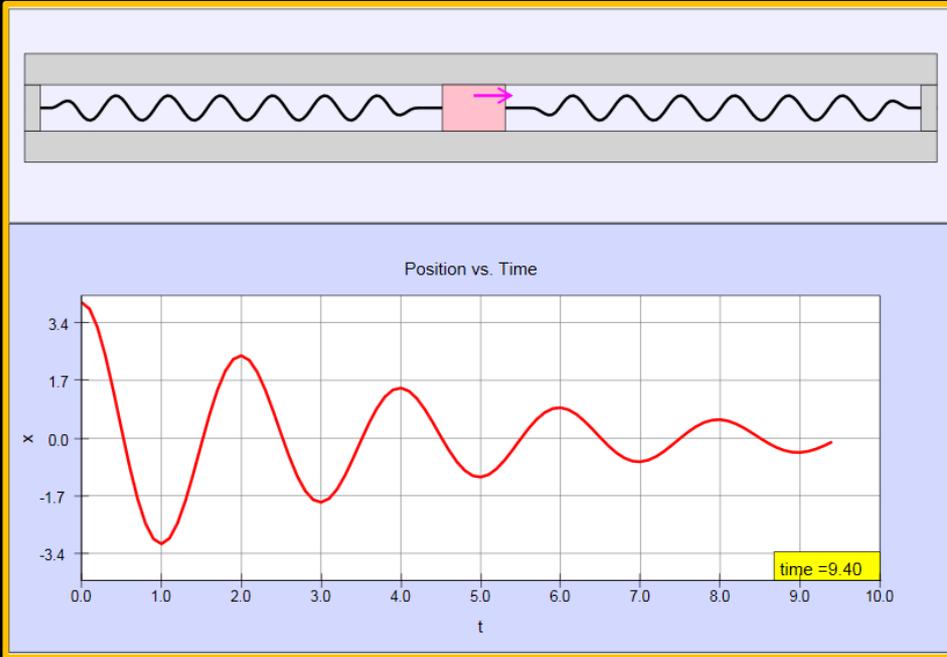
Solver: Cash-Karp 5(4) Tol: $1.0e-6$ Events: 0

Easy Java Simulations: (EJS) is free open source software that is designed to create interactive simulations in Java (applications and applets) without the necessity of prior programming knowledge.

The advantage of EJS for physics teaching is it separates the model into logical parts (variables and evolution) and it separates the model from the view (the visualization of the simulated model).



JiTT Example: A JS Simulation



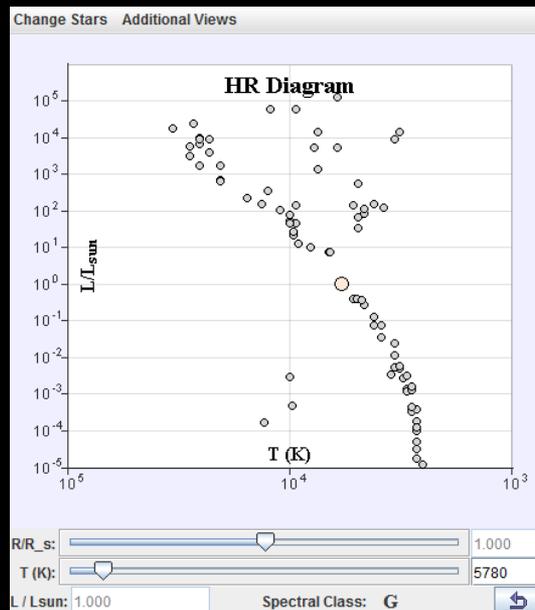
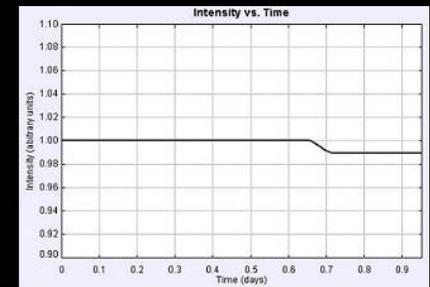
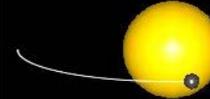
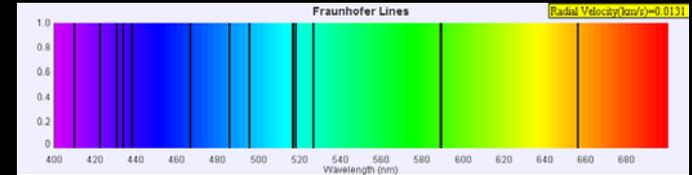
The Horizontal Mass and Spring Harmonic Oscillator model illustrates the forces and dynamics of a simple oscillator.

This simulation uses **the accelerometer on your mobile device** to read the direction of the gravitation field g .

You may need to lock the orientation of your screen to maintain a fixed view as you tilt your device. For use on computers without an accelerometer or for classroom use, the "No sensor" mode simulates a tilted device in the view.

Mass and Spring Simple Harmonic Oscillator JS Model

OSP/EJSs Astronomy Simulations



Over 50 stand-alone Java created with EJSs programs for intro astronomy on **OSP Collection on ComPADRE**

Also organized in M.B. Astro Filing Cabinet

Shared Folders



Astronomy 105 (2 resources, [14 subfolders](#))

Materials in Support of a College-Level Introductory Astronomy Course at Davidson College.

A survey of the current scientific view of the Universe. Emphasis on the physical and mathematical principles necessary to understand how astronomers observe and interpret phenomena. Topics include the historical development of major astronomical theories, the interaction of light and matter, the life cycle of stars, and the structure and evolution of the Universe. No laboratory.

Astronomy 105 Course Home Page

This website is the course homepage for the Davidson College Astronomy (PHY 105) course from the Spring of 2012 taught by Mario Belloni. Many of the following materials were used in the teaching of this course during the spring of 2012.

[website](#)



Davidson College Astrophotography Project

For the past year, as part of teaching the astronomy class (PHY 105), we have been taking astrophotographs. Follow the link to see both our personal and student photos taken in Davidson, NC either on campus or at the Pine Road Observatory.

[website](#)

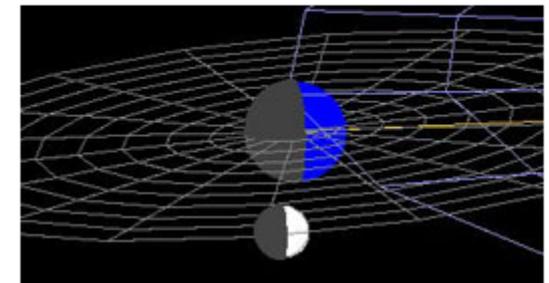


Copy selected into:

Astronomy 105 Subfolders

- ↳ [Naked Eye Astronomy](#) (12)
- ↳ [Optical \(Classical\) Astronomy](#) (0)
 - ↳ [Solar System Models](#) (10)
 - ↳ [Orbits](#) (7)
 - ↳ [Optics](#) (11)
- ↳ [Modern Astronomy](#) (0)
 - ↳ [Stars and Stellar Properties](#) (6)
 - ↳ [Stellar Aberration and Parallax](#) (5)
 - ↳ [Exoplanets](#) (2)
 - ↳ [Galaxies](#) (9)
 - ↳ [General Relativity](#) (5)
 - ↳ [Classical Simulations](#) (3)
 - ↳ [Schwarzschild Metric Simulations](#) (9)
 - ↳ [Kerr Metric Simulations](#) (4)

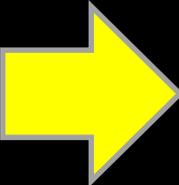
Featured Curriculum Package



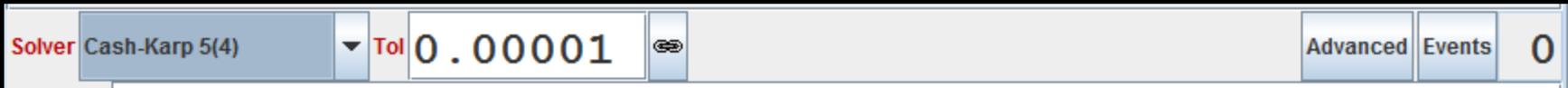
[Introductory Astronomy Models](#)

A shared file folder of Astronomy models designed for a college-level introductory astronomy course. This shared folder contains over 50 EJS models and is broken up into three parts: naked-eye astronomy, classical astronomy (optics and orbits), and modern astronomy.

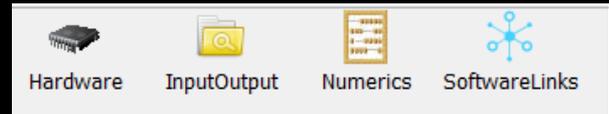
How can we use simulations in teaching physics?

- ▣ **User:** Students access pre-made simulations that (hopefully) they must interact with.
- ▣ **Modeler:** Students are given access to a software package with a simple user interface. Students must then simulate the physics of a problem by modeling at a high level of abstraction. For example, adding the physics in the form of differential equations (rates of change) and initial conditions.
-  **Programmer:** Students are given tools to program a physics example using traditional computational physics techniques.

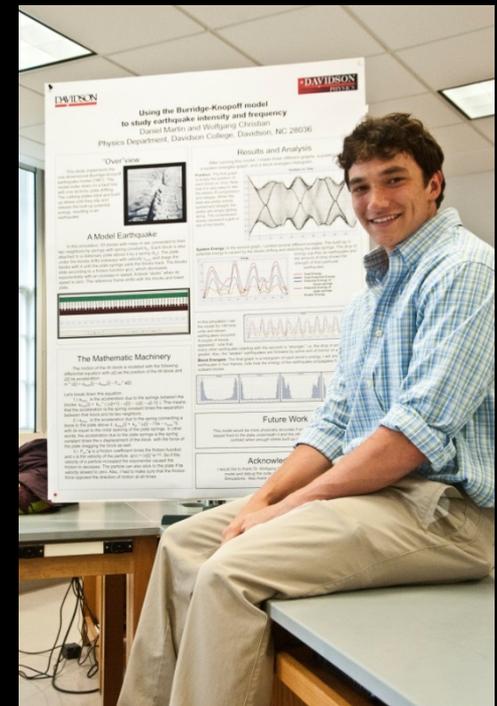
Computational Physics



```
public void poincarePanelAction() {  
    double x = _view.poincarePlottingPanel.getMouseX();  
    double y = _view.poincarePlottingPanel.getMouseY();  
    if (!checkValues(x, y, E0)) return;  
    if (t != 0) {
```

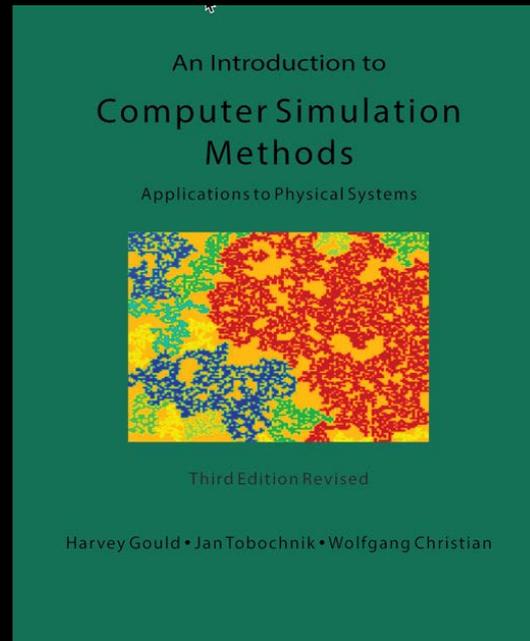


- Differential equations and ODE numerical algorithms: oscillators, Newtonian orbits, and few-body problems.
- PDEs and boundary value problems: Laplace and Poisson equations.
- Stochastic models and Monte Carlo algorithms: Random walks and the Ising model.
- Chaos theory: Logistic map and driven pendulum.
- **Final project of the student's choice.**



Computational Physics Book

PDF available at no cost on ComPADRE: [ID 7375](#)



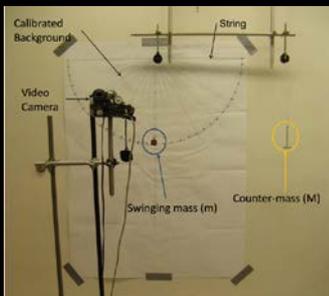
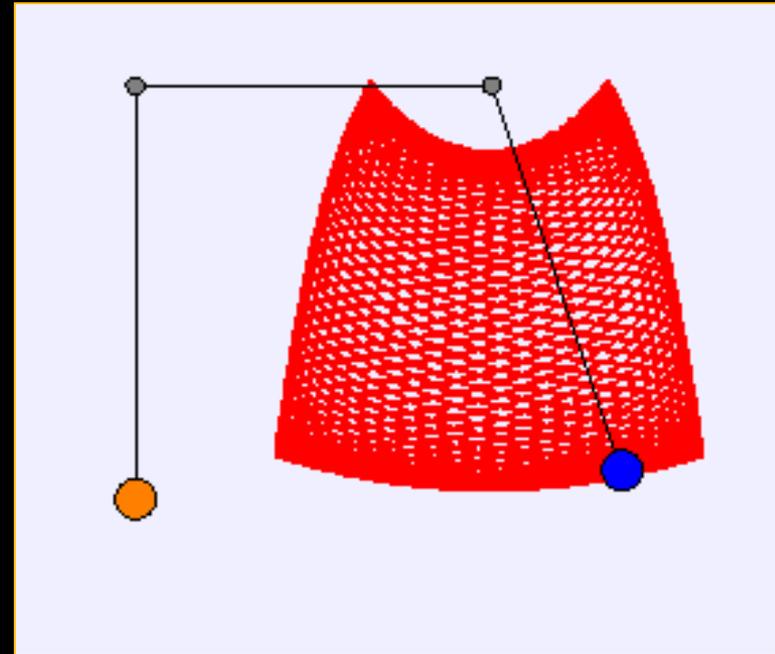
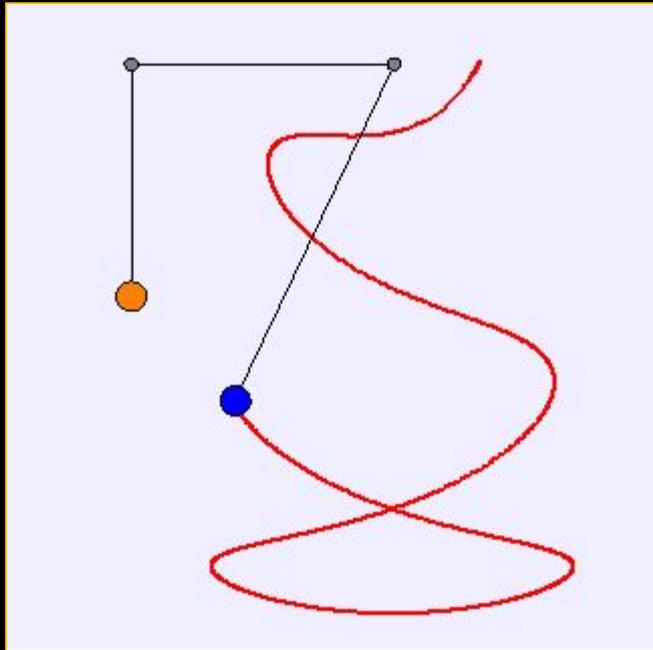
Available at low cost on Amazon using print on demand.

Final Example: Coupling Simulation with Pedagogy

Modeler: Students (intro, **classical mechanics**, computational physics) are given access to a software package with a simple user interface. Students simulate the physics of a problem by modeling at a high level of abstraction. For example, adding the physics in the form of differential equations (rates of change) and initial conditions.

- ▣ **Tracker:** is a free and open source video analysis software program.
- ▣ **Easy Java Simulations:** (EJS) is free open source software that is designed to create interactive simulations in Java (applications and applets) without the necessity of prior programming knowledge to quickly and easily prototype, test, and distribute packages of Java simulations. EJS allows students, teachers, and curriculum authors to easily write and/or change simulations. **Can also be used to teach computational physics.**

Example of Computational Physics and Experimental Physics: The Swinging Atwood's Machine



OSP ComPADRE [Item 11247](#)



What Level of Course?

- ▣ **Non-Science Major/Astronomy**
- ▣ **Introductory Physics and Astronomy**
- ▣ **Major-Level Courses**
- ▣ **Introducing Current Research into Courses**

But expectations, outcomes, and scaffolding are different

What version you use will be related to the course you are teaching, your student body, and your expertise.

In general...

...the less sophisticated the student,
the more sophisticated the user interface
...and the more interactive, the better....

...keeping in mind that
technology without **pedagogy**...
...is just **technology**.



The OSP Collection

Open Source Physics: www.compadre.org/OSP/

U

- *Physlet Physics 2E*: www.compadre.org/Physlets
- *Physlet Quantum Physics 2E*: www.compadre.org/PQP

U

- *Physlet Physics 3E*:
<https://physlets.org/pp/optics/>

M

- Tracker Video Analysis Tool:
www.cabrillo.edu/~dbrown/tracker/

M

- Easy Java/JavaScript Simulations (EJS/EjsS):
fem.um.es/Ejs/

P



The OSP Network:



Open Source Physics - Tracker - EJS Modeling



Physlet Physics



Physlet Quantum Physics

Our Group

- **OSP Java Code**: Wolfgang Christian
- **EJS**: Paco Esquembre and Felix Garcia
- **Tracker**: Doug Brown
- **ComPADRE**: Bruce Mason, Matthew Riggsbee, and Lyle Barbado
- **Java and JavaScript Simulations**: MB, WC, Todd Timberlake, and Michael Gallis
- *Physlet Physics 3E* : Robert Hanson, Anne Cox, MB, and WC
- **Intro Astronomy eBook**: Kristen Thompson, MB