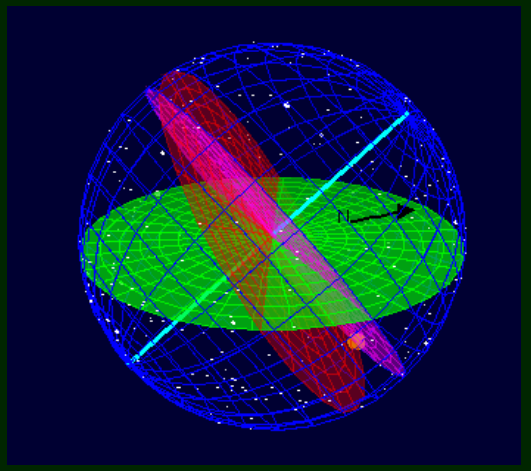
Using Open Source Physics to Teach Physics and Astronomy



New Faculty Workshop
November 2-4, 2017
American Center for Physics, College Park, MD

Wolfgang Christian
Mario Belloni
Davidson College, USA

ComPADRE ID: 14326

Partial funding for OSP was obtained through NSF grants DUE-0442581 and DUE-0937836 and Davidson College. Opinions expressed here are not those of the AAPT, NSF or Davidson College.

OSP Breakout Session

- Brief description/demo of tools
- Explore OSP site/materials
- Discussion of how to use

Open Source Physics Resources/Tools

Open Source Physics (OSP) provides curriculum resources and tools that engage students in physics, computation, and computer modeling. Computational physics and computer modeling provide students with new ways to understand, describe, explain, and predict physical phenomena. This workshop explores the **AAPT-ComPADRE OSP Collection**.

- Physlets are small interactive simulations that are designed for the teaching physics in a web-based environment. **Physlet Physics** is a collection of Java and JavaScript items for the introductory physics sequence.
- **Easy Java Simulations** encourages modeling and authoring with basic programming. EJS removes many of the complicated tasks involved in integrating computation into the classroom allowing students and teachers to focus on the science. The EJS environment allows learners to explore new physics and to test the limitations of the models being used.
- **Tracker** video analysis and modeling tool that analyzes video clips. Students can both analyze the motion of objects and overlay simple dynamical models on the video and see how well the model matches the real-world.

The **OSP Collection** is an AAPT-ComPADRE repository where OSP-based curricular materials can be organized and shared.

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.

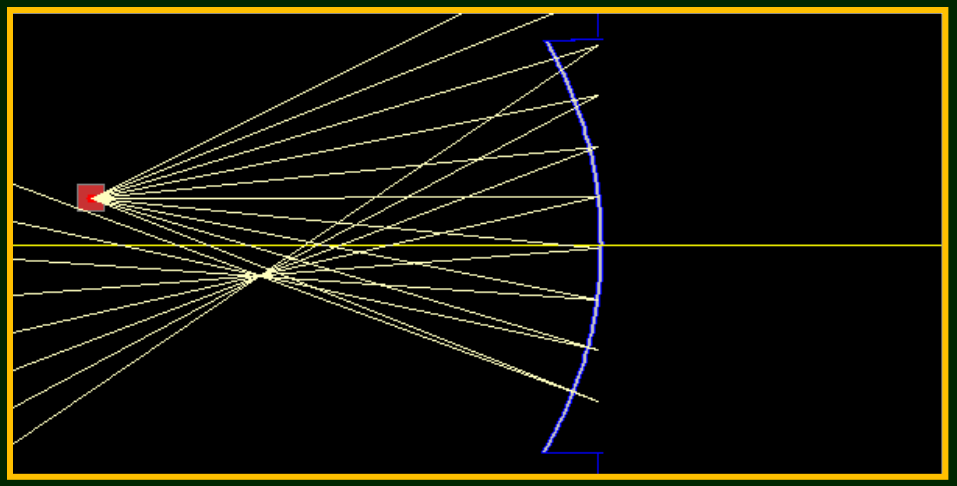
- JavaScript: Scripting language created in 1995 and is part of the HTML5 standard. Recently updated to European Computer Manufacturers Association (ECMA) Script 6.

- HTML5: Markup language standard for the Web going forward. Supports JavaScript, CSS, etc. Does not support Java Applet or Flash plug-ins.

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.

JiTT or ILD Examples



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)**.

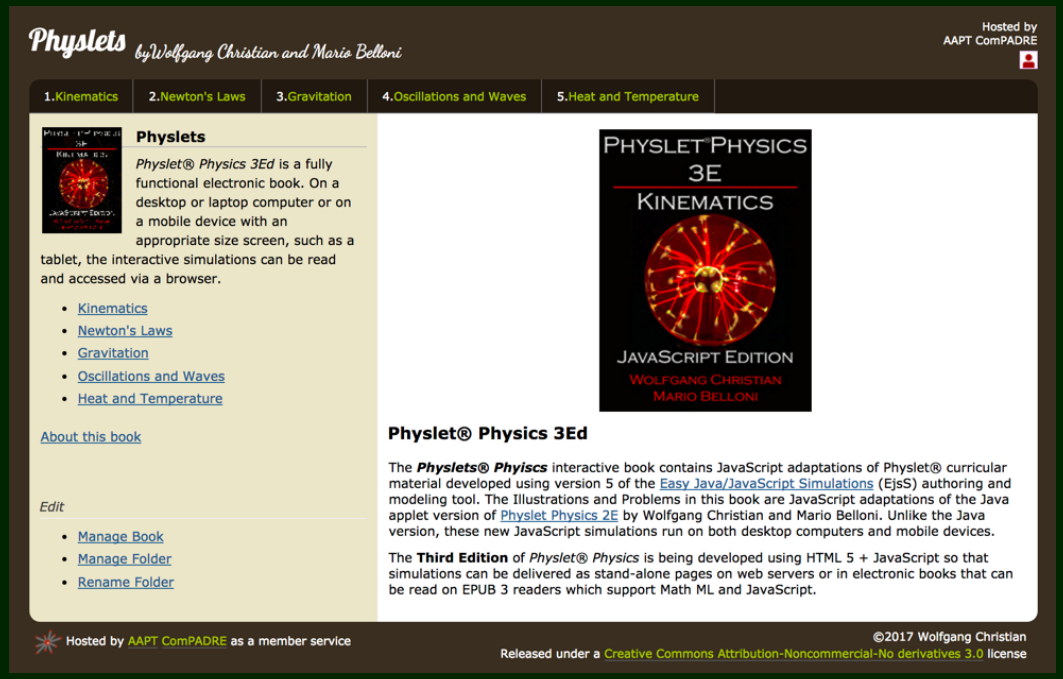
1. Find the focal length of the mirror.
2. Describe the technique(s) you used to determine the focal length.

OSP ComPADRE seach:

- **Focal Length JS (Answer: ID 14037)**
- **Incline Plane JS (Answer: ID 14054)**

EJS Sims & Physlet Physics

Java applets – but not Java programs! – have been deprecated by Oracle and will no longer run in most browsers. The 3rd Edition of Physlet Physics is being developed using HTML 5 + JavaScript so it can be run on any platform including mobile devices.



http://www.compadre.org/books/Physlets-3E

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.

Interactive Engagement

One problem with our system of education is….

…that we reward students for knowing the answers….

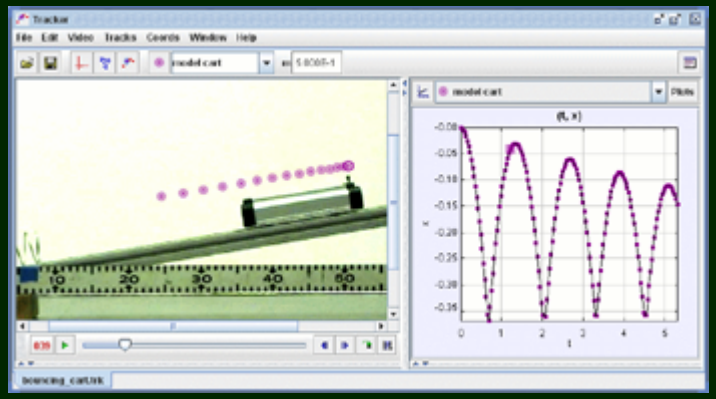
….to questions they have never asked.

Video Modeling with Tracker

Download and install Tracker: https://physlets.org/tracker/

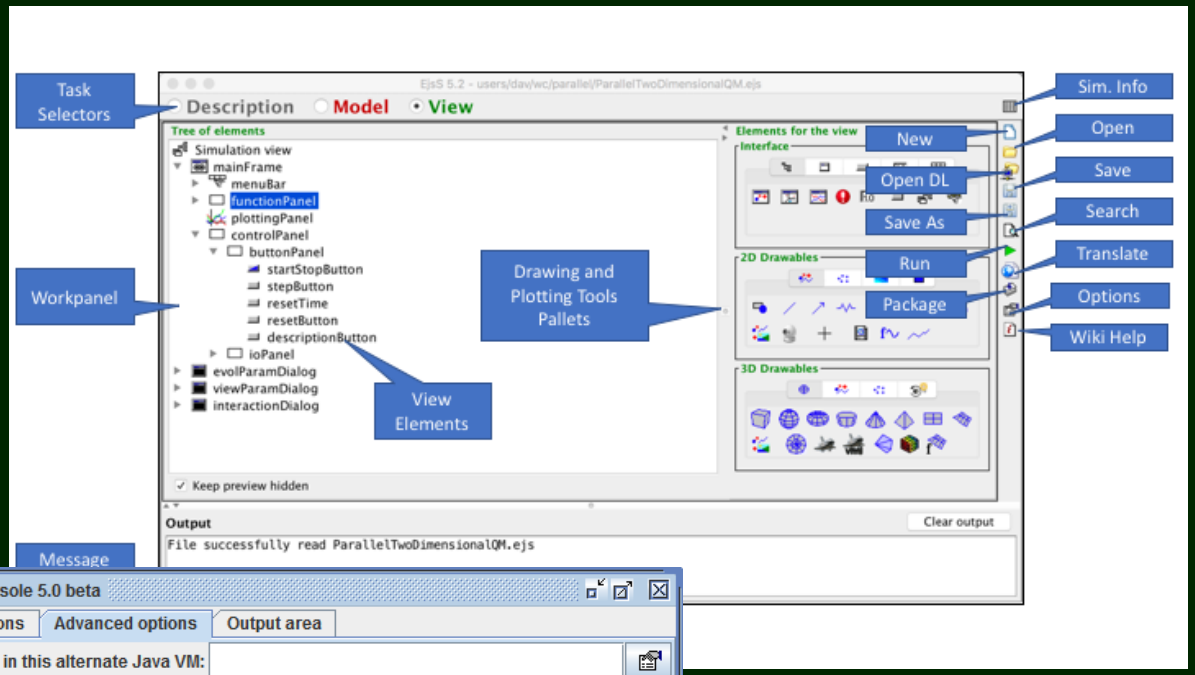
Three easy steps:

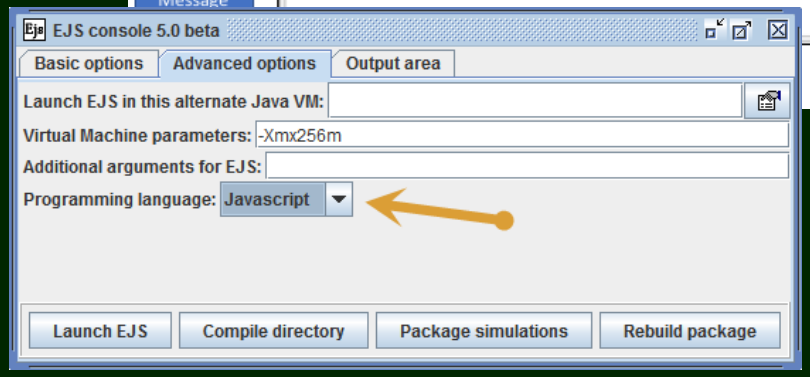
- Load Video
- Set scale and origin
- Shift-click to take motion data



Time for a demo!

Easy Java/JavaScript Simulations

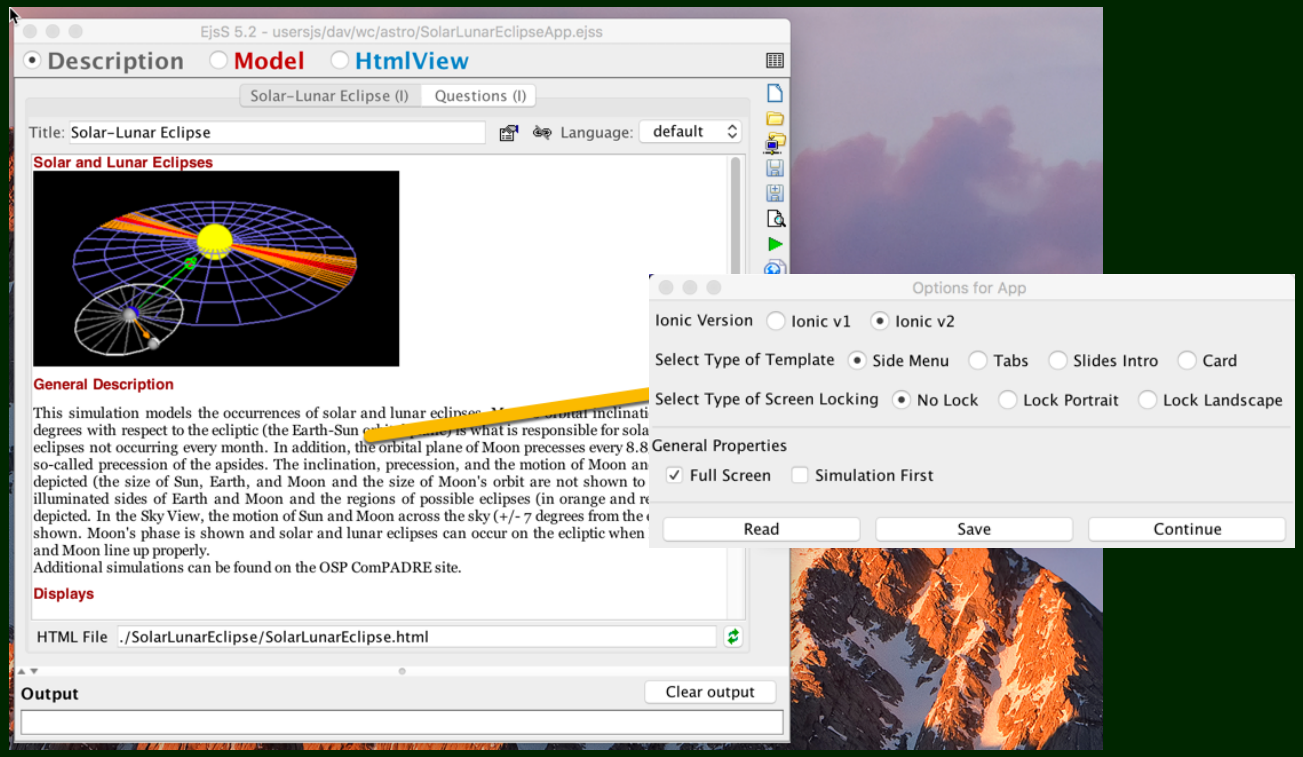




Supports Java and JavaScript models.
Exports JavaScript models as xhtml and as ePub.

Demonstrate how to load, modify, and save a model.

EJS + Ionic creates iOS and Android Apps



Examples: Compadre ID 14328 ID 14326 and ID 13337

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.

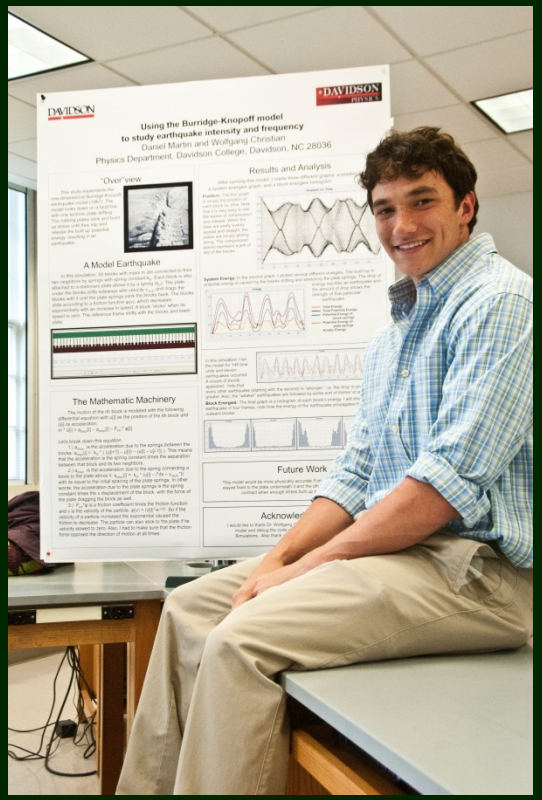
Why might we want to integrate computation into teaching physics?

- Models allow students to think about things in terms of simpler artificial things.
- Computer-based modeling
  - Exploratory simulations engage the student in ideas presented by an expert. Students are led to confront another's view of a problem.
  - Simulation-based and programming activities are expressive exercises that require students to externalize their own ideas and assumptions and to create concrete representations that they can reflect on.

Computational-Physics-Education

Teaching should reflect current research and professional practice. Every undergraduate physics major should know about computational physics, including essential algorithms, some level of programming experience, and computational ways of thinking.

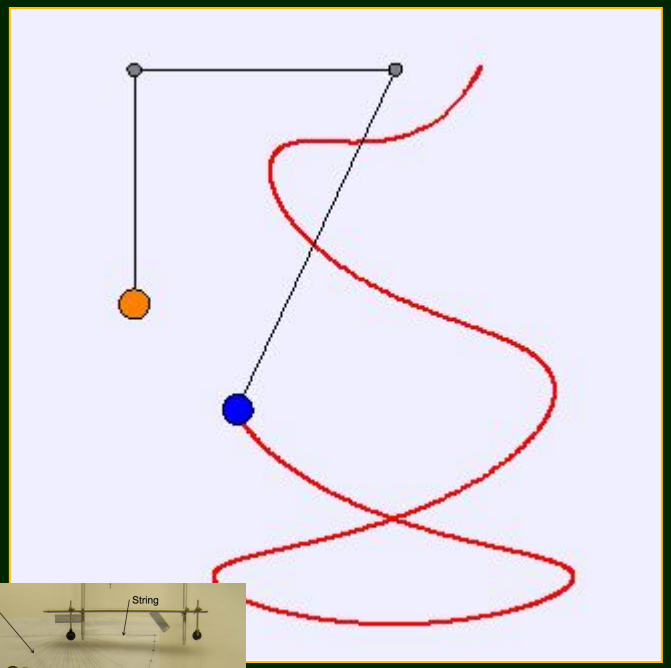
- Differential equations and ODE numerical algorithms: oscillators, Newtonian orbits, and few-body problems.
- PDEs and boundary value problems: Laplace and Poisson equations.
- Stochastic models and Monte Carlo algorithms: Random walks and the Ising model.
- Chaos theory: Logistic map and driven pendulum.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

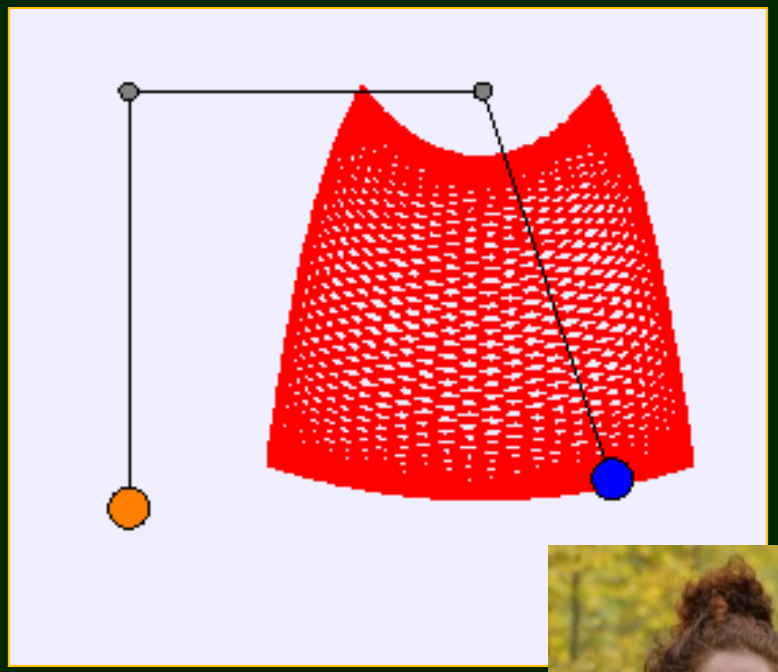


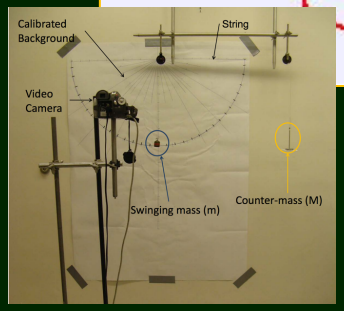
See shared filing cabinet.

Example of Computational Physics and Experimental Physics:

The Swinging Atwood's Machine







OSP ComPADRE Item 11247

Modeling Cycle

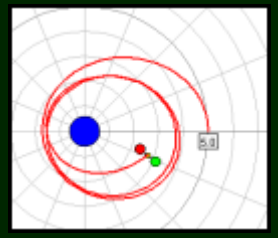
- The goal of modeling is to teach in a student-centered environment where students do not solve problems in a formula-centered way.
- Modeling Instruction attempts to enhance student achievement through a process called the **Modeling Cycle**, (following Robert Karplus' Learning Cycle).
- Throughout the Modeling Cycle we rely on student engagement and explanation as the dynamic of learning.
- The start of the modeling cycle is the development phase:
  - Qualitative description
  - Identification of variables
  - Planning an experiment
  - Performing the experiment
  - Analysis of experiment
  - Presentation of results
  - Generalization

Although the Modeling Cycle can be used without computers, it is well suited for computer modeling if we replace the word "experiment" with "simulation" in the development phase.

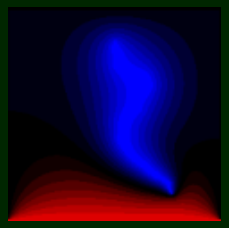
After the development phase, the model is deployed in a variety of new physical situations in a variety of different ways.

Projects in ComPADRE (shared filing cabinet)

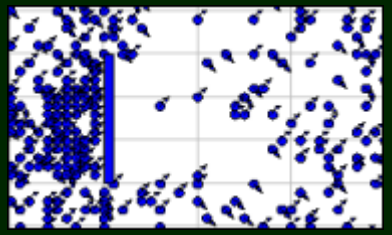
- Hyperion Orbit
  (J. Barrick)



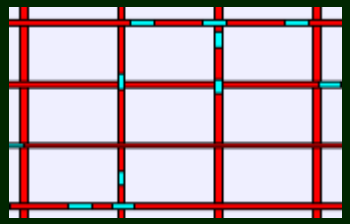
- Lightning
  (S. Castle)



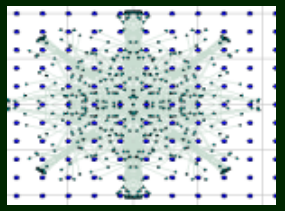
- Lattice gas
  (B. Gautier)



- 2D Traffic Flow
  (F. Healy)

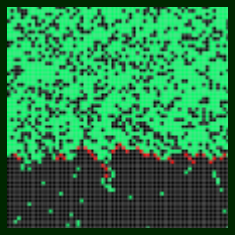


- Lorentz Gas
  (S. Keller)

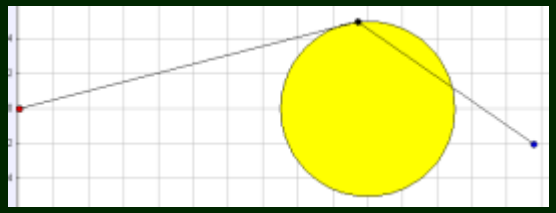


- Fractals
  (S. Mohammed)

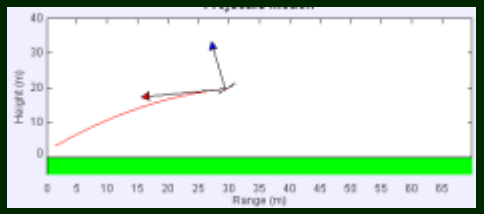
- Forest Fires
  (M. Mohorn)



- Catastrophe Theory
  (D. Glassman)

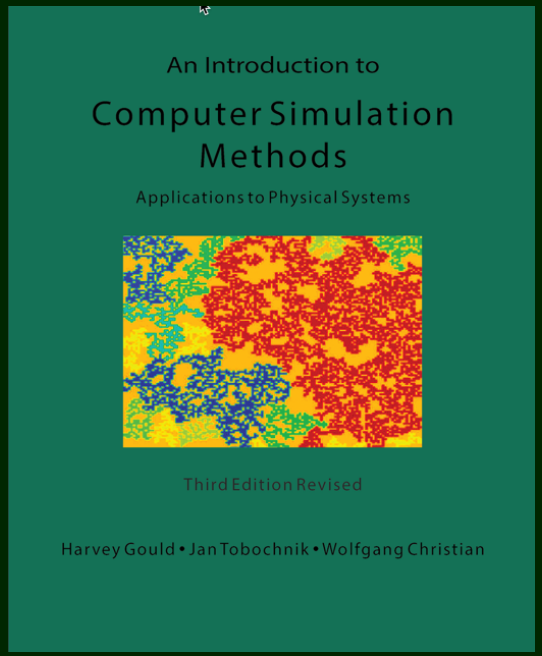


- Javelin Throw
  (P. Wall)



Computational Physics Book

PDF available at no cost on ComPADRE: ID 7375



Available at low cost on Amazon using print on demand.

Need for Digital Libraries

A Google search for "***pendulum***" returns 11,600,000 pages; while "***pendulum simulation***" returns 2,490 pages (The search for ***pendulum simulation*** without the quotes returns 449,000 pages).

- Most of the simulations (or animations that "fake" the physics) are inappropriate for teaching.
- There is usually no instructional material, no support materials for teachers, and no information about how these materials are correlated to state or national science standards.
- Most of these simulations also support a passive (viewing) pedagogy versus an active (interacting) pedagogy.

In order to be effective for instruction, simulations need to be easy to find, simple, adoptable, adaptable, and coupled with support content for students and teachers.

ComPADRE

We are out of the business of web hosting and let the experts do it.

Standard and Custom Library and Web Services

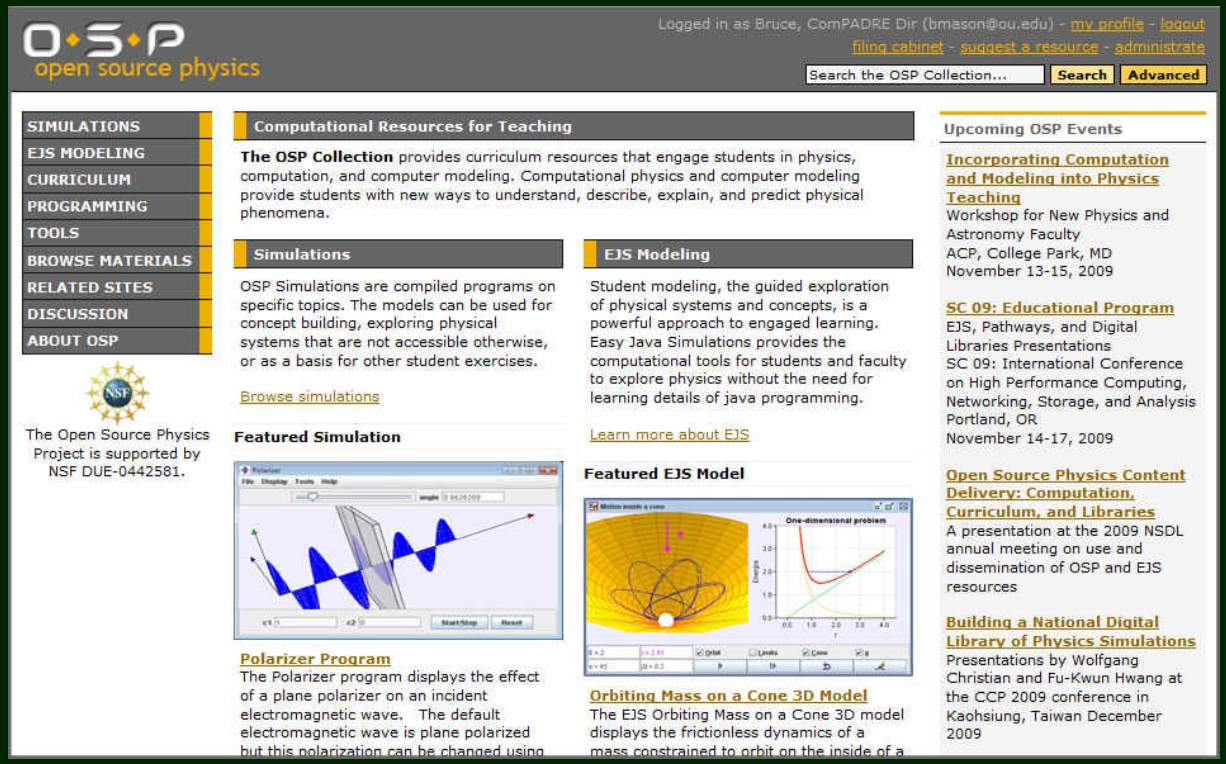
Connections to NSDL Users

500 OSP Resources

800 Physlet Resources

12,000+ visitors/month

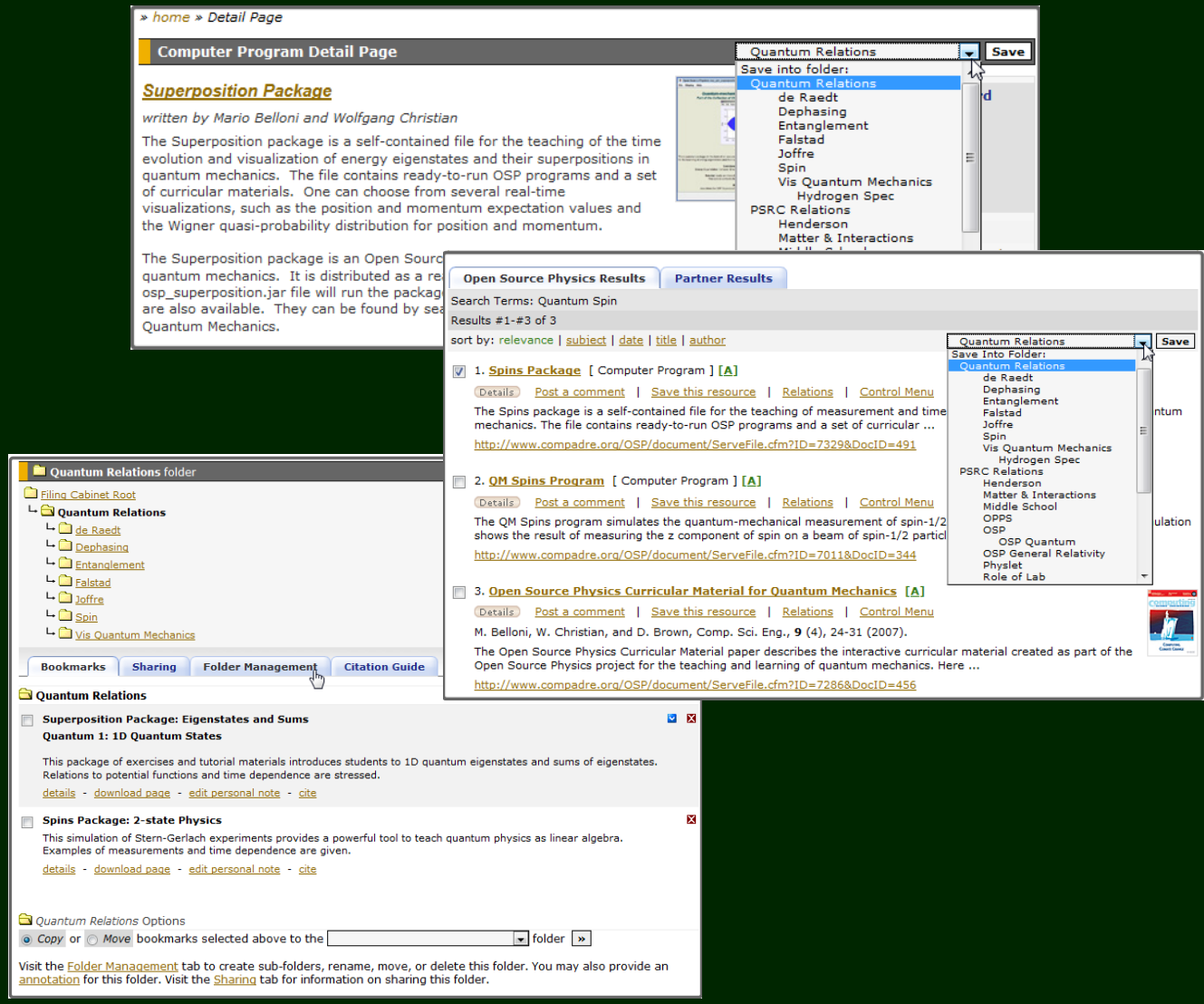
5,000 simulation downloads/month



Personalization

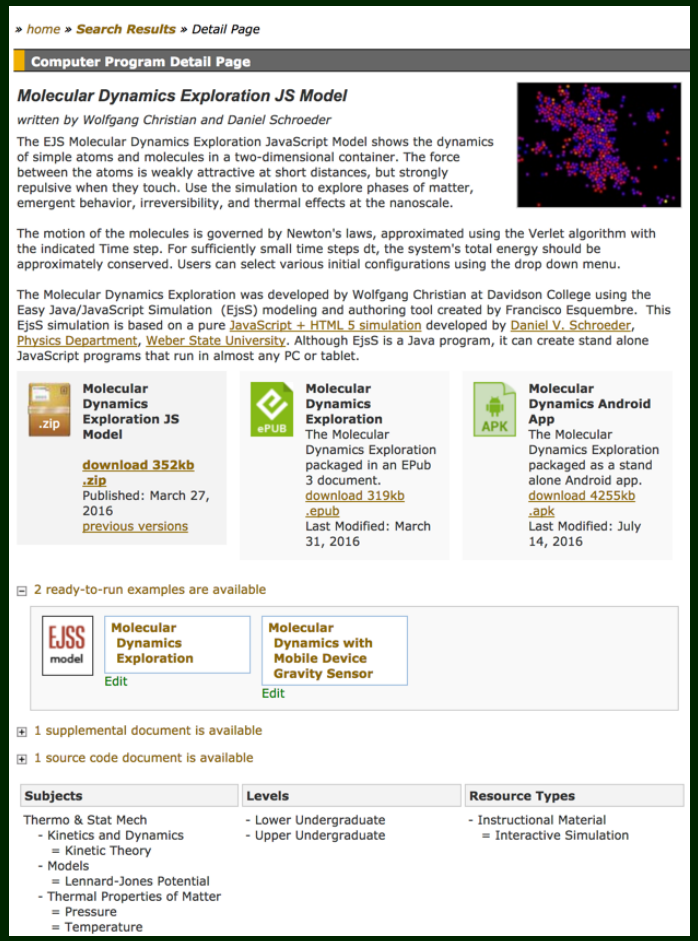
Content …

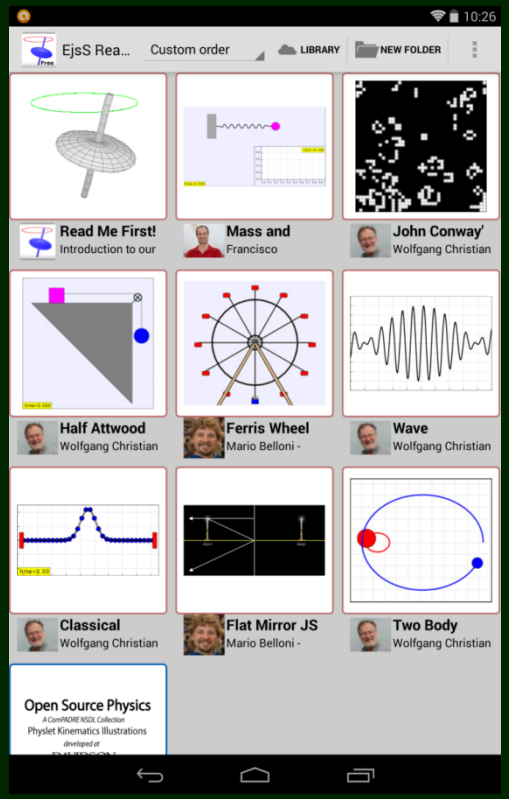
- Find
- Collect
- Sort
- Relate
- Annotate
- Share



OSP Support for new technologies

EJS creates ePubs and Apps that run on mobile devices.





Reader App (Android) (iTunes)

Why open source curricular material?

- Shift from low-value work to high-value work. We like to say in open source that all the easy problems have already been solved.
- Lower total cost. Using open source curriculum shifts the cost from licensing and purchasing to customization and implementation.
- Given enough eyeballs, many problems are shallow (Linus's Law). Empirically, open source tends to produce better quality material because more people can contribute.
- Open source provides many advantages. First, you have the opportunity to tap the knowledge of the world's best educators, not just those in your organization. Second, the number of potentially contributors and thus the potential knowledge pool is orders of magnitude larger. Finally, open source curricular material gets adapted to a variety of use cases, not just the one the creator originally intended.
- 0pen source promotes the sharing of ideas. Open source isn't a fad, or a bunch of hippies experimenting with illicit substances.

What 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.

Summary

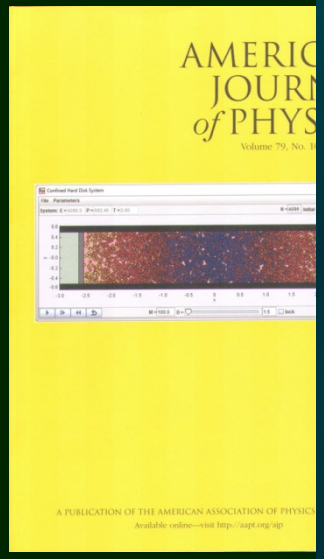
The **OSP Collection** removes many of the complicated tasks involved in integrating computation into the classroom allowing teachers to focus on the science.

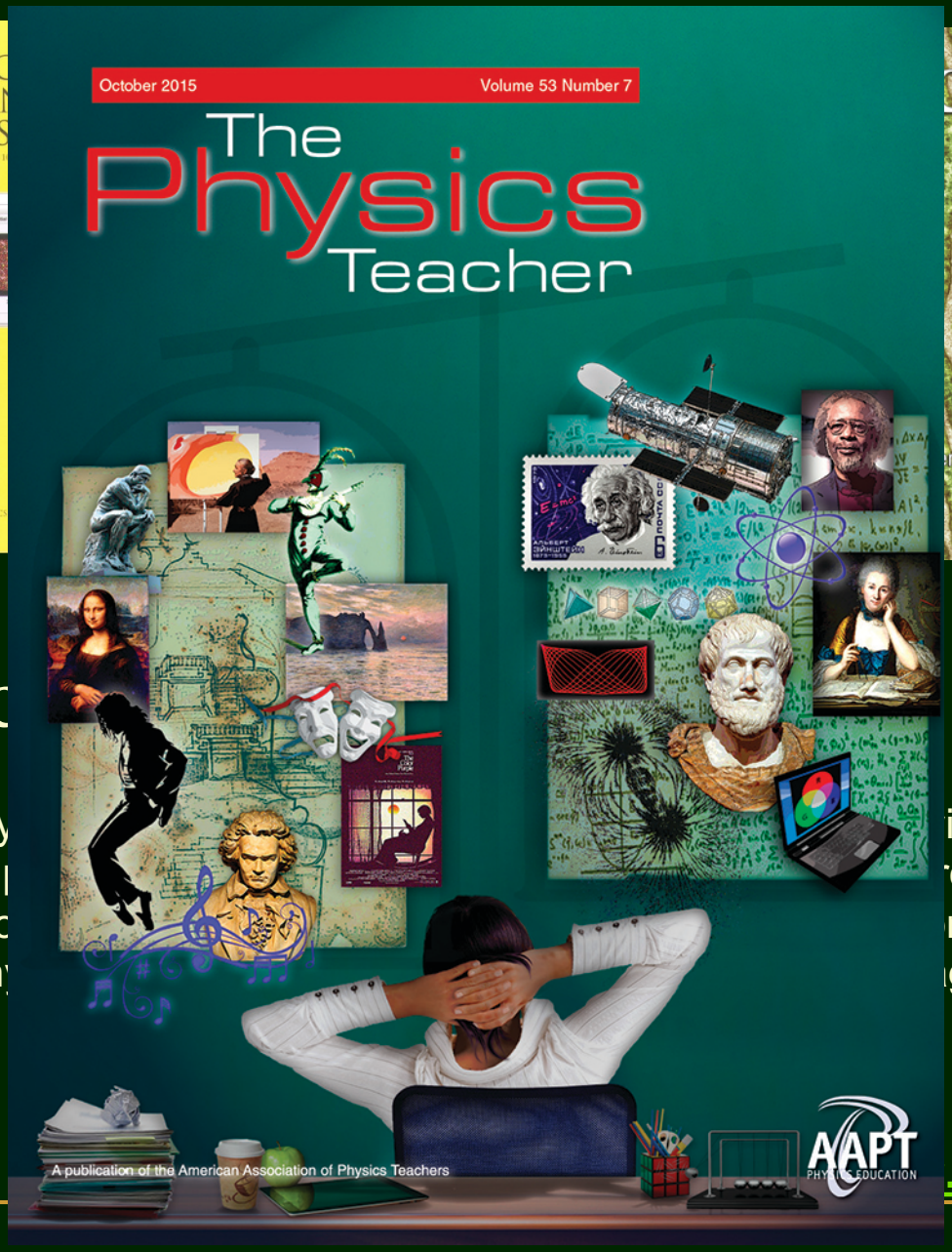
- OSP provides computational tools, including a computational physics textbook, for our project.
- OSP allows learners to engage in computational physics modeling.
- OSP encourages the sharing of curricular materials by allowing instructors to adapt existing EJS models to their particular needs.
- ComPADRE supports distribution and collaboration by providing an internet portal and a web service of models that are directly downloadable into the EJS and Tracker modeling tools.

The OSP Collection in ComPADRE is a repository where programs, models, and curricular materials can be organized and shared by developers and instructors around the world.

AAPT - ComPADRE Team

- **OSP Managing Editor:**
  - Wolfgang Christian - Davidson College
- **OSP and EJS Curriculum Authors:**
  - Mario Belloni - Davidson College
  - Anne Cox - Eckerd College
  - Fu-Kwun Hwang – National Taiwan Normal University
  - Harvey Gould - Clark University
  - Jan Tobochnik - Kalamazoo College
- **Tracker and OSP Tools Developer:**
  - Doug Brown - Cabrillo College
- **Easy Java Simulations Developer:**
  - Francisco Esquembre - Universidad de Murcia
- **OSP Java Library Editor:**
  - Wolfgang Christian - Davidson College
- **AAPT-ComPADRE :**
  - Bruce Mason – University of Oklahoma
  - Lyle Barbato - AAPT
  - Caroline Hall - AAPT
  - Erwin Campbell- AAPT



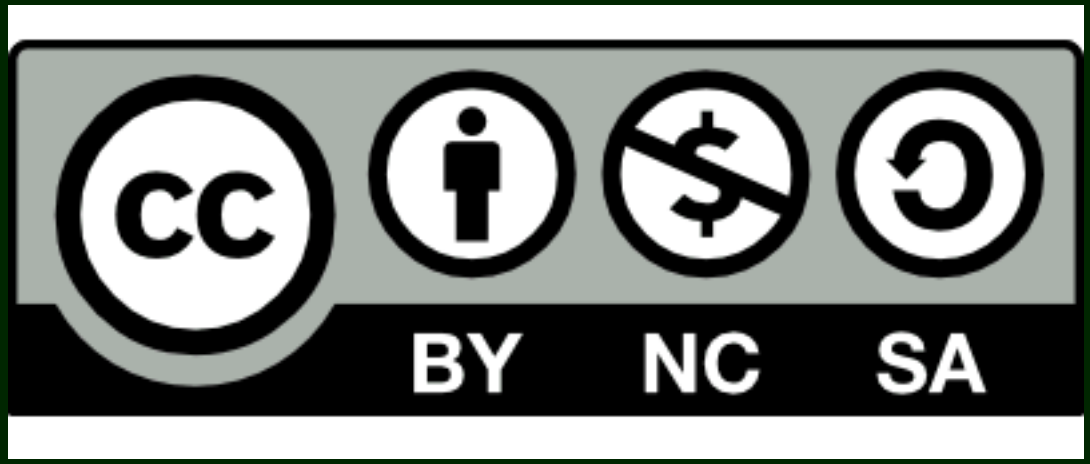


Open Sourc

During the past y iews and 100,000 visitors. More imp om the Collection and many additio rkspaces. During the past year, Phy ges viewed.

.compadre.org/osp

Thank your for your attention.



Download our work from the OSP Collection on AAPT-ComPADRE