FOSS Pyscript Vernier Labs & Simulations for Education Proposal

Angel Duarte

Problem

Proper physics education requires for students to be able to investigate the relationship between variables and create, evaluate, and apply usable theoretical models based on their research and investigations. The ability for computers to make computations and calculations at rates previously inaccessible to people makes them an indispensable tool in facilitating and accelerating the student reasoning with scientific topics, models, and principles; they grant students access to insights and visualizations historically locked away behind very laborious work.

Constructivist physics educators, such as those educated in modeling physics pedagogy at Arizona State University (ASU) and ASU's own Physics Instructional Resource Team (PIRT), often depend on computer-based data collection and analysis methods and computer simulations to form their instructional methods. Unfortunately, it is not uncommon for software to be proprietary and closed source, and therefore present the following accessibility problems:

- The software is locked behind a paywall and has expensive licensing fees.
- The software has limited or no ability for customization/adaptation.
- The software is buggy.
- The software is discontinued.

The company Vernier, in particular, has very useful hardware probes and sensors useful for all sorts of physics education demonstrations, labs, and experiments. However, their keystone piece of software for the usage of a great majority of their legacy hardware, Logger Pro, has been discontinued, and is being replaced by an updated version of Graphical Analysis. The latter option, however, does not count with the same features, is not immediately compatible with most of the legacy hardware, and not available for Linux desktops.

While many people involved in physics education are competent at using python for computational reasons,

It is therefore my intention to use our knowledge of python to create accessible software to be applied in physics education.

Approach

The company Vernier has python code under the GPL that is compatible with their probes and sensors. This code can be used locally on any machine to make use of their sensors. This code can be adapted to for the purposes of particular labs, activities, and demonstrations.

In 2022, PyScript was released to the public. PyScript is a library that allows for Python code to be embedded into HTML documents. This implies that it would be possible for a python-based simulation or lab activity to be hosted on an HTML file hosted locally or on a web page. Additionally, HTML formatting could allow for the convenient and easy creation of front-ends using widely understood web-design principles.

Matplotlib has libraries that create animations and graphs, that would be very useful for the purpose of allowing for the visualization of objects and graphs for students.

Our understanding of computational physics would allow for the data collected using sensors to be easily processed for the purpose of analysis using our learned ODE and integration methods. Alternatively, data points can be generated for simulations.

The combination of all of these things presents an avenue for the creation of Free and Open Source (FOSS) software that is convenient and accessible for educators and learners.

Objectives

- Create a lab activity that uses vernier probes for collecting data.

    - Using ODE methods and the relevant python libraries, create options for data analysis and integration. For example, create velocity and acceleration vs time graphs based on data sets for position and time.
    - Create the pertinent graphs required by the lab experiment.
    - Make parameters, such as data collection time and data collection rate, customizable.

- Alternatively, create a useful educational lab.
    - Use relevant computational methods to generate data points. For example, integrators can be used to find position vs time points based on initial velocity, acceleration, and time parameters.
    - Allow for parameter customization.
    - Create graphs and animations for each simulation.

- For the desired activity, create an HTML document or web-page that the user can interface with to achieve the intended demonstrations or educational outcomes.
    - Make a user interface that does not require for the user to understand python code to run it.
        - If possible, make the python code available for user in case they would like to customize it themselves or have a look ‘under the hood’.
    - Have any graphs created appear on the webpage.
    - Provide the option to view tables or data collected.
    - Make the design intuitive and easy to use.