

Simple Physics with Python: a workbook on introductory Physics with open source software

Andrea Mandanici¹, Giuseppe Mandaglio¹, Giovanni Pirrotta¹, Valeria Conti Nibali¹, and Giacomo Fiumara¹

¹University of Messina

March 11, 2022

Abstract

Skills in computer science can have great value in studying, doing and communicating physics. As educators, we asked ourselves how to make students aware of that, and how to offer them a new and appealing approach to physics. We also wondered how to increase students' engagement, participation and understanding, particularly when lessons are delivered online. Thus, we began a project to develop study materials for an introductory course in physics for computer science and we chose to use open source software. The materials are organized as a set of Jupyter notebooks hosted on an open GitHub repository. The notebooks deal with fundamental concepts of physics related to everyday life, offering examples of what can be done with a few lines of Python code. In the notebooks we propose activities to observe phenomena, describe problems, experiment, acquire and analyze data, and model the behavior of systems. The contents are suitable for undergraduates, high-school students, and evergreen students. We have used the materials for lectures, guided laboratory activities and presentations to freshmen and younger students, and we plan to continue with this project.

Introduction

Digital technology provides many innovative tools for teaching and learning, including powerful open-source software and virtual spaces to collaborate and share documents. We see these as real opportunities to develop and provide learning materials with which students can get more deeply involved in science. In particular, Jupyter Notebook (<https://jupyter.org/>) offers a versatile framework for producing scientific worksheets, course modules [1], or open reproducible papers [2, 3]. A rich gallery of examples is available at the Jupyter wiki (<https://github.com/jupyter/jupyter/wiki>). Many collections of Jupyter notebooks are hosted in GitHub (<https://github.com/>). Open notebooks available publicly concerning topics in physics are mostly on advanced topics, whereas we wanted to exploit the features of the Jupyter Project at a beginner level, with emphasis on practical aspects, experiments, and a hands-on approach.

Simple Physics with Python is the name of a collection, hosted in a GitHub repository, in which we present a set of Jupyter notebooks on elementary mechanics [4]. They cover topics in kinematics and are grouped in chapters concerning one-dimensional motion, two-dimensional motion, and support topics, respectively. A further chapter contains notebooks on slightly more advanced examples.

Each notebook begins with the statement of a physics problem, connected to a real situation. Then we discuss the problem in a wider context and propose a model, along with equations involving relevant variables. Finally, we use simple Python code to calculate and visualize the results.

We proposed this material as workbooks for courses in general physics for majors in Computer Science, Information Technology, and Data Science. It also would fit well in an introductory physics course for undergraduate students in Science and Engineering.

Concepts

Through our digital workbooks we want to convey some fundamental messages to users, and focus on some key aspects.

Physics is not just equations

For each problem, we propose a model of a physical phenomenon, so that we can describe what happens with suitable detail, and possibly make reliable predictions. For example, we can compute the position of an object as a function of time, and the instantaneous values of velocity and acceleration. We discuss the case of [projectile motion](#), in which we can draw the provisional trajectory, given the initial velocity and the initial angle with which the object is launched. We can easily observe what happens on varying the initial angles, i.e., drawing many possible trajectories, so we discover that an entire region could never be reached by the object under the given initial conditions.

Results are not just numbers

Everybody knows, more or less, how tall they are, what is their weight, or how much they should pay for a sandwich. It would be great if participating in a physics course one could learn to know what is roughly the expected value of some physical variables, so as to judge whether the results of an analysis are reasonable. In almost every notebook we include a section entitled *Get a feel of*, in which we try to make the reader enter in closer contact with the problem discussed and think about the values of the relevant variables. To this aim, we also make comparisons with well-known examples: we compare the Earth-to-Moon distance with the mean Earth radius, or the acceleration of a car with the acceleration of gravity g . Moreover, we take the opportunity to use the example on acceleration for suggesting a comparison between the performance of electric vehicles and diesel vehicles.

Simple models are useful

Even simple models such as that of a particle with constant speed can be applied to many real cases. For example, we consider [ultrasonic distance sensor](#) for assisted parking, the measurement of the [Earth-to-Moon distance](#), the time-of-flight of electromagnetic signals for [communications from Earth to the International Space Station](#). We can also evaluate how long one travels in a car while reacting to some external stimulus ([reaction time](#), like a traffic light turning from green to yellow). We also discuss the case of freely falling objects and that of a car going from 0 to 100 km/h, as examples of one-dimensional motion with constant acceleration.

In our experience, these models, usually presented at the very beginning of physics courses, are dealt with too quickly by the students (and sometimes by the instructors), with the risk of remaining unclear.

Methods

Jupyter Notebook as a powerful and accessible workspace to study, research, report

Thanks to the Python code cells embedded in Jupyter notebooks, the reader can change the values of some variables, and observe the effects on the other physical quantities that characterize the problem. Actually, in the examples proposed, Python scripts are used for several tasks: * using Python as a calculator * computing multiple variables in a automated way * evaluating data, reading data from a text file, writing data to a text file * visualizing the results with plots and graphs * saving the graphs as high-quality images in widely used formats like png, pdf, jpeg. * analysing data, carrying out linear regression * displaying the results as a nicely formatted table, in Markdown language or in HTML;

Other specific advanced examples could be developed concerning: - how to plot a graph using a selected image as background; - how to select frames from a video - how to get data from a video frame in order to detect the position of an object during its motion

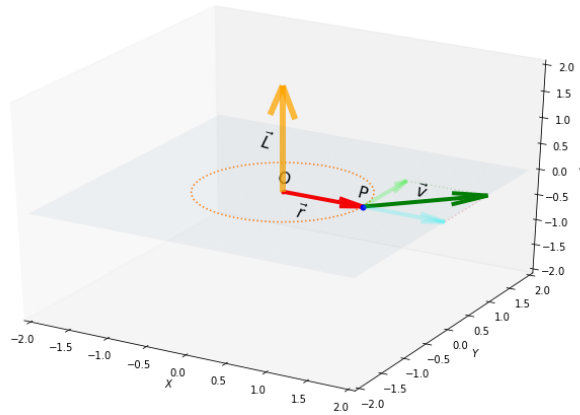


Figure 1: The angular momentum, computed as vector product using NumPy, and plotted with Matplotlib

Going through the material, readers can get an idea of what can be done with Python and how to use Jupyter Notebook as a tool for their study and professional needs. They could opt for Python with Jupyter to actively perform research, and to report on scientific tasks.

Moreover, the skills acquired can be used outside the world of physics: readers can step into the field of data analysis or enjoy interdisciplinary examples, such as evaluating data about physical training and fitness.

Experiments, model, simulation

We introduce some of the topics starting from experiments that can be easily done by students (*[Trip on a lift](#)*, *[Free fall](#)*) and experiments reproduced in a laboratory or at home with inexpensive hardware and **open-source software** (*[Ultrasonic distance sensor](#)*, *[Motion on an inclined plane](#)*).

A Jupyter notebook such as *Trip on a lift* or *Inclined Plane* can offer opportunities for an extended interactive session including hands-on experiments, data acquisition, data evaluation, graphical representation, analysis, modelling and discussion.



Figure 2: ‘Trip on a lift: experiment and data analysis’ is one of the activities proposed in the workbook.

In particular, the *Trip on a lift* notebook allows learning step-by-step how to evaluate velocity and acceleration, and highlights also the concept of negative velocity. The experimental investigation of *Projectile motion* allows to detect and compare directly a case of constant velocity with a case of constant (negative) acceleration, while looking specifically at the motion along the two axes.

The power of images

Images and videos can be used to better describe a problem or for gaining the attention of the readers (*Earth to Moon distance*, *Long distance communications*), but they also become tools for running experiments, for instance by means of video capturing.

Python AND open-source software

The notebooks refer not only to Python cod but also to free and open-source applications such as for instance [Phyphox](#), an App turning the smartphone into a mobile lab. We use the [Tracker](#) package for recording videos of the experiments and for extracting the coordinates of the object as a function of time. Finally, we use the [Audacity](#) software for recording sounds, for Fourier analysis of audio traces, and for generating tones at a desired frequency. Along with simple and inexpensive experiments, suitable to be performed also in distance-learning settings, and open-source Apps, the Jupyter Notebook can be used to wrap up the information in an optimized way.

The importance of definitions

Angles, trigonometric functions, and vectors remain topics not well understood by many freshmen. A thorough understanding of these, however, is fundamental for taking full advantage of physics courses. For this reason, the workbooks contain a section with supplementary materials in which readers can play with the measurements of angles, or with the values of trigonometric functions, and can visualize vectors as well as the results of operations on them.

Mark the route

“How did we do this?” Several times you need to do something similar to a project that you developed previously, but you do not remember the details of how you arrived to that beautiful result. Taking notes that are going to be helpful for future work should become a habit. For this reason, every notebook of our collection also contains references, notes, and URLs.

Use reliable sources

Nowadays we often find many suggestions by searching the Internet, blogs, discussion groups, etc.; it is very useful to have good references to start with: books, online courses, original documentation of software packages. Using verified sources should be recommended.

How to better exploit this material

The collection Simple Physics with Python can be viewed via a web browser, in its public [GitHub](#) repository. The Jupyter notebooks can also be rendered under the [nbviewer](#) application. Each notebook also has a button to launch the [binder](#) environment, which allows to use the notebook in an interactive mode even if Python is not installed on the user’s computer. Some tens of seconds will be needed to load the environment. The notebooks can be read also on smartphones, but some of the interactive objects, like the slider widgets, become then more difficult to use. A suggested way to exploit this GitHub resource is for the user to install the [Anaconda](#) package, download a clone of the repository, and run the notebooks on their local machine. Other services like [Google Colab](#) could be used to develop and run the notebooks on a cloud platform, with no need to install applications on the local machine. On the other hand, Jupyter allows preparing the notebooks while working offline.

This work was inspired by

- Lorena Barba, [Engineering Computations](#)[5].
- David J. Pine, [Introduction to Python for Science](#)[6].
- Lorena Barba and Olivier Mesnard Olivier, [Aero Python: classical aerodynamics of potential flow using Python](#)[7].

Outcomes and further work

The work for developing these thematic notebooks on Simple Physics with Python has led us to experience a different way of teaching, and to provide students with new tools towards quality education. We have used the notebooks in several contexts. Other than in lectures, we exploited them for laboratory activities and meetings or events for young students. We would like to improve and extend the materials to cover other topics on general physics.

Concluding remarks

The collection Simple Physics with Python can be used for remote learning projects, courses and presentations for prospective students. We hope this helps students to approach physics but also interest readers to glimpse the advantages of Python and of the Jupyter environment for scientific purposes. Specifically, in the notebooks we like to propose *a* method to solve a problem, which is not *the* only one, neither *the best* one necessarily. This concerns both physics and coding in Python. We hope to raise in users their curiosity for physics with open-source software and inexpensive hardware, stimulating at the same time a hands-on approach in learning.

Acknowledgements

We thank the Editor-in-Chief Lorena Barba and the Department Editors Dirk Colbry and Sharon Broude Geva for insightful comments and suggestions.

References

1. Lorena A. Barba, Lecia J. Barker, Douglas S. Blank, Jed Brown, Allen B. Downey, Timothy George, Lindsey J. Heagy, Kyle T. Mandli, Jason K. Moore, David Lippert, Kyle E. Niemeyer, Ryan R. Watkins, Richard H. West, Elizabeth Wickes, Carol Willing, and Michael Zingale, [Teaching and learning with Jupyter](#), Open book (2019)
2. Achintha Rao, [Standing on the shoulders of programmers: the power of free and open-source software](#), Physics World, September 2021
3. Jana Lasser, [Creating an executable paper is a journey through Open Science](#), Communication Physics, 3 (2020), 143.
4. Andrea Mandanici, Giuseppe Mandaglio, Giovanni Pirrotta, Valeria Conti Nibali, Giacomo Fiumara, [Simple Physics with Python](#), GitHub.
5. Lorena Barba, [Engineering Computations](#), Computing in Science & Engineering, 22(4), pp.26-35, “Engineers Code: reusable open learning modules for engineering computations” 2020, doi:10.1109/MCSE.2020.2976002, preprint on arXiv::2001.00228.
6. David J. Pine, [Introduction to Python for Science](#) and [Introduction to Python for Science and Engineering \(1st ed.\)](#) CRC press (2019). <https://doi.org/10.1201/9780429506413>
7. Barba, Lorena A., and Mesnard, Olivier (2019). [Aero Python: classical aerodynamics of potential flow using Python](#). Journal of Open Source Education, 2(15), 45, (<https://doi.org/10.21105/jose.00045>)

The authors

- **Andrea Mandanici** is associate professor of experimental physics at the University of Messina. He teaches physics courses for the Degree in Computer Science and for the Degree in Mathematics. He’s coworking on activities concerning Physics with Open Source Software. He is corresponding author of this article.
- **Giuseppe Mandaglio** is associate professor of particle physics at the University of Messina. He teaches courses on Data Analysis and Particle Physics Laboratory for the Degree in Physics and for the Master in Physics. He’s Coordinator of the Degree Course in Physics, Coordinator of the master’s degree course in Physics, and member of the doctoral school of Physics of the University of Messina. He’s author of course materials on C++ and on data analysis with the use of the ROOT framework developed by CERN. He is corresponding author of this article.
- **Giovanni Pirrotta**, Master in Computer Science and PhD in Mathematics, is IT expert staff member of the University of Messina. He’s author of projects and publications on Open Data.
- **Valeria Conti Nibali**, is tenure-track professor at the University of Messina and teaches Physics for the Bachelor’s Degree in Computer Science - Data Analysis. Her research interests are in Computational and Experimental Condensed Matter Physics and Biophysics.
- **Giacomo Fiumara** is associate professor at the University of Messina, senior member of the board of professors of the bachelor’s degree in computer science, and of the master’s degree in engineering and computer science with the University of Messina. He is a member of the doctoral board of the School of Mathematics and Computational Sciences, University of Palermo. He also teaches bioinformatics for the international degree course in medicine and surgery. His research interests include social network analysis, data analysis, information system analysis, parallel computing, parallel and distributed algorithms for the analysis of large-scale networks, statistical physics, and condensed-matter physics.