Improving Teaching Techniques Using Visual Python: A Case Study In Physics Laboratories

Ergi Bufasi, Kawtar Lakrad

Abstract:— In order to describe physics phenomena, science develops sophisticated models that use mathematical and formal program languages. When a phenomenon is not accessible to our sense, its description is indirect and understanding can be difficult for students. When one succeed to obtain a direct visualization of a phenomenon inaccessible to sense, it is possible to get a deeper understanding since a very effective channel of learning is involved. Using a variety of visualization tools for teaching and learning science is necessary for students to better understand physics phenomena and formulate appropriate mental models. Here, a simple rendering tool for 3D objects and graphs is presented which is aimed at students with minimal programming background. The purpose of the presented study is to determine the role of simple computer codes to improve the teaching techniques for physics laboratories. Students learn the importance of a virtual laboratory as a visualization element when addressing physics contents within science classes. From the basic atomic physics to nuclear physics, from the magnetic field to the electric field, many phenomena can be suitable for creating 3d visualization that allow capturing a greater perception of the physical world.

Index Terms:— Computational modeling, 3D visualization, physics laboratories, PASCO Capstone, teaching techniques, VPython.

1 INTRODUCTION

Science educators have believed that the laboratory is an important means of instruction in science since late in the 19th century. Laboratory teaching assumes that first-hand experience in observation and manipulation of the materials of science is superior to other methods of developing understanding and appreciation. Smith (1991), contended that most scientific theories are based on a large number of very sophisticated experiments. He suggested that, if the lecture topics are to be illustrated, this should be done through the use of audio-visual aids or demonstrations [1]. Further, many of the skills students learn in laboratories are obsolete in science careers. Positive research findings on the role of the laboratory in science teaching do exist. Laboratory activities appear to be helpful for students rated as medium to low in achievement on pretest measures [2]. However, practical teaching and learning process in the labs can be improved by introducing more effective technique of delivery through the use of software in the computer labs. A central aspect of science is the modeling of complex real-world phenomena .A physical model is based on what we believe to be fundamental principles; its intent is to predict or explain the most important aspects of an actual situation. Modeling necessarily involves making approximations and simplifying assumptions that make it possible to analyze a system in detail [3]. Computational modeling is now as important as theory and experiment in contemporary science and engineering [4]. Modern developments in computer languages and computer hardware have enhance the use of programs, codes and libraries with minimal programming skills. Nowadays it is possible to create, diffuse and improve source codes by educators and students for solving or illustrating textbook exercise of physics [5].

- *Ergi Bufasi : College of Engineering and Technology, American University of the Middle East, Kuwait ; E-mail: ergibufasi@hotmail.com*
- *Kawtar Lakrad : Subatomic and medical Physics Group, Hassan II University, E-mail: kawtarlakrad@gmail.com*

2 DESCRIPTION OF THE PROGRAM

2.1 What is Python?

Python is the easiest to learn and nicest to use programming language in widespread use. One of Python's great strengths is that it comes with a very complete standard library. It can be used to program in procedural, object-oriented, and to a lesser extend [6]. The Python language plus its family of packages comprise a variable of ecosystems for computing. A package or module is a collection of related methods that are assembled together into a subroutine library. In order to do 3D visualizations and matrices with Python an additional module need to be included, VPython (Python plus the Visual package).

2.2 What is Visual Python?

Visual Python (VPython) is the Python programming language plus a 3D graphics module called "Visual" originated by David Scherer in 2000. Creating animations with visual is essentially just making the same 2-D plot over and over again, with each one at a slightly differing time, and then placing the plots on top of each other. When performed properly this gives the impression of motion. VPython was initially written by David Scherer under the supervision of Bruce Sherwood and Ruth Chabay and is released under the GNU Public License [7]. When using VPython the display window shows object in 3D the center of the display window is (0, 0, 0). The +x axis runs to the right, the +y axis runs up and the +z axis points out of the screen, toward you. You could create different shapes like sphere, arrow, box, cylinder, text, label, etc. [8]

2.3 Input

The program requires the user to enter the values for the parameters describing mass, gravitational force, velocity and to define the type of object to be animated. The units are relative and can be outlined as a comment using hash character, #, and extend to the end of the physical line.

2.4 Output

The output of any–print-statements you execute in your program goes to the Output window, which is a scrolling text window. When you ran the program, two new windows will appear; one displaying the object created and one displaying text. The text-output window, also called “Shell” window, displays any text you tell your program to print as well as displays any error statements that help you to find mistakes in your code. The Shell window is also used to display any text that you instruct the program to print; often for printing the final results. [9]

3 MATERIALS AND METHODS

In this study we will construct simple computational models based on fundamental physic principles. Using VPython, a computational environment based on the Python programming language, we will create simple computational model that produces a navigable 3D animation as a side effect of the physics code. Computational modeling allows us to analyze complex systems that would require very sophisticated mathematics or that could not be analyzed at all without a computer. The focus of this paper will be on a single program developed to create 3D visualization especially for the experiment conducted in the labs before performing them, giving students the opportunity to watch the dynamical evolution of the behavior of a system, understand and modify these expressions without any difficulty. Such animations provide powerfully motivating and instructive visualizations of fields and motions. The PASCO Model SE-9638 e/m apparatus will be used in this study to match up the experimental work performed in physics laboratories with the computational model produced in 3D animation based on the same experiment. [10]. The PASCO Model SE-9638 e/m apparatus provides a simple method for measuring e/m, the charge to mass ratio of the electron. The method is similar to that used by J.J. Thomson in 1897.

***Fig.1**: PASCO Model SE-9638 e/m apparatus*

A beam of electrons is accelerated through a known potential, so the velocity of the electrons is known. A pair of Helmholtz coils produces a uniform and measurable magnetic field at right angles to the electron beam. This magnetic field deflects the electron beam in a circular path [11].

4 RESULTS

With the use of the computer language VPython we will be able to model physical phenomena to help us visualize and make physics laboratories more interesting. We will simulate“Electron-Charge-to-Mass-Ratio” experiment in order to understand how electric and magnetic fields impact an electron beam. For successfully implementing the simulation we will need to know the magnetic field B and the velocity of the electric charge. In every computer we need to specify some basic variables without which a computer program fails to run.

```
from visual import *
qe=-1.6e-19
m=9.11e-31
vi=vector(0,-5e5,-1e5)
B=vector(-10e-5,0,0)
```

Here qe is the charge of an electron and m is the mass of an electron. In VPython we can display various windows, but initially there is one Visual display window named scene. Any objects we begin creating will by default go into the scene. In this program we embellish the display window by modifying the attributes with the help of the display function. scene=display(title='ElectronChargetoMassRatio',width=1600, height=1600,center=(0,0,0),background=(1,1,1)) In order to make the particle move across the screen and to calculate the force and the position as it moves perpendicular to a uniform magnetic field (B) we will use while loop. A loop is an instruction that tells VPython to perform a series of tasks repeatedly until the program meets some criterion.

```
while t<3.5e-7:
        rate(10)
        F=qe*cross(vi,B)
        vf=vi+F*dt/m
        electron.pos=electron.pos+vf*dt/2
        vi=vf
        t=t+dt
```

In this example, the variable t is given as initial value before the loop begins. The while statement instructs VPython to execute the indented statements over and over, until the value of t becomes equal to or greater than 3.5e-7. At that point, the indented lines will no longer be executed. Now running the code, the 3D simulation will appear in the visual window, but in the output window, a long list of position will appear (it prints in each iteration).

```
Print 'Electron position =', electron.pos,'m'
```

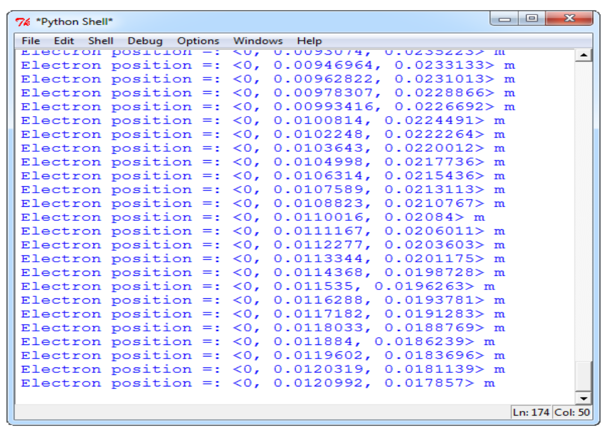

***Fig.2**: Output window - “Shell window” showing the electron position in each iteration*

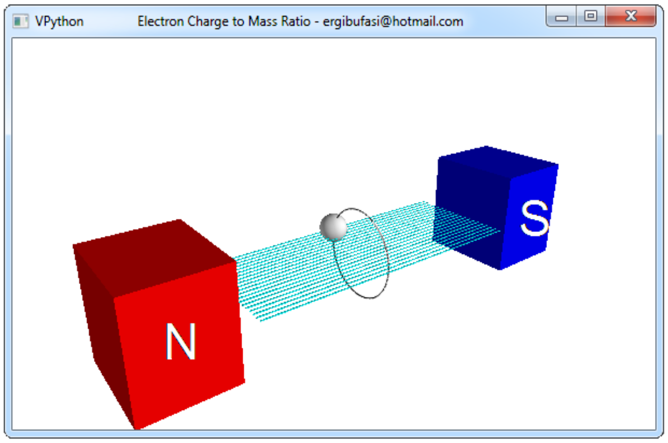

***Fig.3**: Electron charge to mass ratio simulation.*

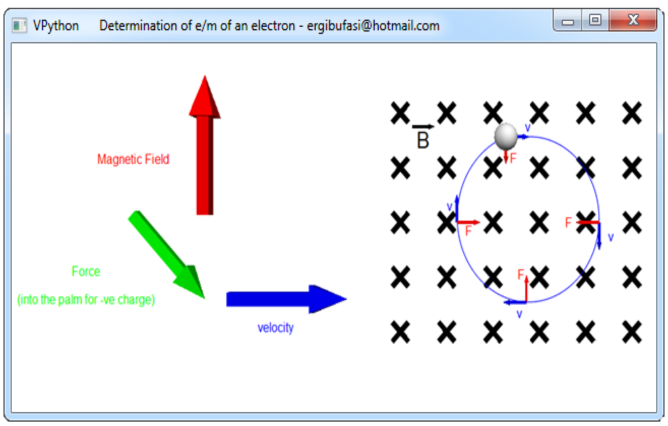

***Fig.4:** Determination of e/m of an electron simulation.*

5 Conclusion

Presenting physics with more than one method leads to an easier understanding. The philosophy behind this study is to attract students towards physics by offering new perspective over the theoretical concepts and experiments. Combining theoretical knowledge with simple programming skills, students will be able to simulate and model physics experiments before performing them on the laboratories. We believe that it is easy to solve exercises and problems with the presented program and the students benefit from it in different ways: checking the laboratories results with the ones obtained on the computer, the code is easy changeable to suit the needs and substantially they get a clear perspective on how to solve physics related calculations.

References

[1] Smith, M. E., C. C. Hinckley, and G. L. Volk (1991). Cooperative learning in the undergraduate laboratory. Journal of Chemistry Education. 68:413-315.

[2] Orzechowski, R. F. (1995). Factors to consider before introducing active learning into a large, lecture-based course. J. Coll. Sci. Teaching 24(5):347-349.

[3] Chabay, R. W. & Sherwood, B. (2008). Computational physics in the introductory calculus-based course. American Journal of Physics 76 (4&5), 307-313.

[4] Beichner, R, Chabay, R. W. & Sherwood, B. (2010). Labs for the Matter & Interactions curriculum. American Journal of Physics 78 (5), 456-460.

[5] Fangohr, H. "Lecture notes on Computational Science" pp. 136-146,2006.

[6] Summerfield, M (2009). "Programming in Python 3: A complete introduction to the Python language", Crawfordsville, Indiana: Addison –Wesley.

[7] Scherer, D & Dubois, P & Sherwood, B (2000)." VPython: 3D interactive scientific graphics for students. Computing in Science & Engineering. 2.56-62.10.1109/5992.877397.

[8] Olenick R. P. (2007). "Visual Python in a computational physics course" PyCon, Addison, TX.

[9] Chabay, R. & Sherwood, B. (1999). "Bringing atoms into first-year physics" American Journal of Physics 67, 1045-1050.

[10] Chabay, R. W. & Sherwood, B. (2006). "Restructuring Introductory E&M. American Journal of Physics 74, 329-336.

[11] PASCO Capstone, "Instruction manual and experiment guide for PASCO scientific Model SE-9638", 012-03471E.