

VISUALIZATION AND COMPUTATIONAL MODELLING USING VPYTHON TO TEACH PHYSICS AT THE FACULTY OF SCIENCE, UNIVERSITY OF DERNA

Aya Muhammad IDRIS¹

University of Derna – Libya

Aml Muhammad IDRIS

University of Derna – Libya


Abstract

Students face many difficulties in understanding some imaginary physical phenomena, mathematical or logical concepts and formations, which makes it difficult for the university professor to reach the required scientific perception, so the professors are forced to move forward using all digital innovations in order to provide the highest possible teaching quality., This paper describes computational modelling. And simulates and visualizes some physical phenomena in college using the VPython programming language (Visual Python), which is a great tool for demonstrating, calculating, and simulating physics displays and experiments, as VPython is a powerful and easy-to-use environment for creating 3D animations and publishing them on the web. It contains many built-in objects and shapes, as well as graphic capabilities. They can be used to create interactive programs using user interface elements. The process of making the program interactive starts first with creating the objects and setting variables for them, then creating a widget that the user can interact with, which is a useful exploration and a new way to improve students' skills and support the interactive teaching of computers and humans for Physics, as well as a response to the 'education' reform in Libya.

Keywords: *VPython, Animation, Visualization, Modeling, Education, Interactive.*

Introduction

Physics has been at the forefront of computational science since the invention of the computer [1-7]. Where it is consideredThe field of computational physics is a subset of mathematics and computer science[8]. It deals with the use of numerical techniques to address physical issues. It complements experimental physics and theoretical physics, the three common problem-solving approaches in physics. Because most physics problems can be nonlinear or difficult to solve precisely, computational physics involves computer

 <http://dx.doi.org/10.47832/2717-8234.17.12>

¹  amemohmd123@gmail.com



simulations and is often referred to as "computer experiments". Every field of physics has a computational branch in which it can be found. For example, computational fluid dynamics, computational mechanics, computational electromagnetics, computational astrophysics, computational solid-state, computational statistical physics, and computational particle physics [2-3-26] Many computer learning techniques that can be used to teach physics can be applied so that students can Understand, imagining, simulating and accessing basic physics concepts: Python[9] is one of the most widely used and user-friendly programming languages in the entire world. Visual Python[10-11], sometimes known as VPython, is a variable used frequently in physics. Python requires an additional package to generate 3D visualizations and arrays. VPython is visually packaged as part of a large library. A collection of connected methods is packaged into a library as a package or module. [4] This article suggests a teaching reform strategy for the 3D dynamic presentation of physics by fully utilizing VPython, a free, open-source programming language, and outside libraries built in the Python programming language. This can significantly lessen the students' anxiety about learning college physics, boost their motivation to learn and create a solid foundation for developing creative talents in the new period. One of the most widely used languages right now is Python. It is straightforward and simple to understand. Many free, open-source programs and extension libraries for scientific computing have been created in recent years using the Python programming language, including NumPy, SciPy, SymPy, which give Python quick access to calculations, sketching, and symbolic and numerical computation tools. To process experimental data, create charts, and even create scientific computing applications, scientific researchers can greatly benefit from the development environment created by the Python language and its many extended libraries. This is especially true for interactive teaching platforms based on the Jupyter Lab or Notebook, which can seamlessly import a large number of rich third-party libraries. Visual Python originated by David Scherer in 2000. which combines the Python programming language with the Visual 3D graphics package. Basically, making visual animations is just creating the identical 2-D graphic repeatedly, each at a somewhat different time, and then stacking the plots on top of one another. When done correctly, this creates the illusion of motion. Under the guidance of Bruce Sherwood and Ruth Chabay, David Scherer initially created VPython, which is made available under the GNU Public License [5] When creating a science visualization scene with VPython, the viewport is centred at (0, 0, 0) and displays 3D objects. The +y axis runs up, the +z axis points out, and the +x axis extends to the right. Student programmer from the screen. Many shapes, including spheres, arrows, squares, cylinders, and labels can be designed with different colours, locations, dimensions, variables, and constants. [6]

2-Developing Computational Modeling

It has been determined that computational reasoning is a crucial strategy for helping pupils comprehend STEM concepts and scientific methods. Mathematical problems with arithmetic solutions are helpful in STEM concepts because they enable applying physics or STEM theories to issues with complex mathematical explanations[12.13.23. 27]. With the help of the VPython library, students can create 3D models, program physical equations, and investigate the impacts of altering parameters. Students are able to comprehend challenging mathematical ideas and understand their relation to real-life problems because the environment produced uses basic equations (force, momentum, velocity, or gravity dependant) to compute solutions. Computational approaches can be incorporated in a variety of ways into the conventional modules that make up an established undergraduate or graduate curriculum. Typically, this is accomplished through computational homework, computational lectures, computational labs, and computational research projects[14]. The Partnership for Integration of Computation into Undergraduate Physics (PICUP) organization has built an outstanding platform for appropriate course materials, teaching methodologies, and workshops, in the field of physics [15]. In a survey conducted by Zhang et al., a majority of materials science and engineering students have indicated that learning computational skills in the second undergraduate year is optimal [16]. The American Association of Physics Teachers (AAPT) [17] standards for computational physics were created specifically to encourage a greater emphasis on computing in undergraduate physics curricula. The primary driving force behind this work is the growing recognition of computing as a post-baccalaureate degree requirement for physics majors: "Computational physics skills, or the ability to use a computer to calculate solutions to physics problems, are highly valued by research, industry, and many other employment sectors" [17].VPython can be introduced simultaneously with an introduction to programming if students have no prior programming experience]18]

3 -The concept of 3D visualization

Numerous fundamental physical quantities, including momentum, angular momentum, electric, and magnetic fields, are abstract three-dimensional vectors. These variables are dynamic in both time and space. Few first-year students have any prior experience visualizing a momentum vector changing in three dimensions as an object moves or the shifting magnetic field of a moving charge at various positions throughout space. Although pre-made graphics or movies have their uses, creating dynamic 3D visualizations utilizing the fundamentals of physics can give concepts like observation sites and relative position vectors more substance[19]. Even when a situation appears to an expert to be in 3D, it is in 2D to a rookie pupil and difficult to interpret. For instance, despite the fact that a particle's track in a cyclotron is planar, we have seen even well-prepared students struggle with addressing issues on paper or using computer programs to describe the process in 2D. The addition of a 3D wireframe model of the cyclotron dees and 3D arrows depicting the

magnetic and electric forces in a 3D programming environment, however, helped many students understand the issue. Another useful aspect of the 3D display was its interactivity, which allowed users to zoom and rotate the image.[29-20] VPython automatically allows the user to rotate and zoom using the mouse, which enhances 3D perception[21-24]. VPython has a 3D animation capability that makes it simple to represent physical phenomena in a 3D environment. It gives students the chance to select their viewing angle during a human-computer interaction learning session, increasing their interest in learning.[20]

4 - Examples of visual computational modeling activities using Vpython taught in college

These activities focus on helping students visualize physical quantities, especially directionality such as velocity, momentum, and electric and magnetic fields, as quantities that change over time. Example activities can be found on the Trinket website, which is an open site where users can add or modify VPython programming code [10].

4 (A)-Simulation and modelling of electromagnetic fields

In the study of electric and magnetic fields, the concept of electromagnetic fields in three-dimensional space is very abstract and difficult to understand. In addition, the distribution of electric and magnetic fields in different situations is very complex. It is very difficult to determine the distribution of these fields through experiments. VPython's embedded object provides a ball representing a point charge, where people can set how much charge it holds. Through programming based on Coulomb's law, the electric field distribution of different point charge systems can be calculated. Figure 1(a) shows the 3D model of the electric field of a positive point charge[28-22-25-]. To create the simulation only two objects were used ball and arrows. The ball represents the charge, where the codes are entered in terms of location, coordinates in the directions of the three axes, diameter, and colour. The value of the charge in coulombs is also entered, and then it is located on the screen. Create another object, which is the arrow that represents the field lines, and specify its location and direction based on the location of the charge. The electric field is a vector quantity whose value and direction the student must theoretically determine. Usually, the student finds it difficult to imagine but by creating a 3D scene, the student enters the directional values of the charge, the observer's position, and the reference point. The electric field strength of the point charge at the point of observation is calculated using the law numerically with the creation of a three-dimensional visualization. The student enters the coordinates of the point of observation and is far from the point of origin with a vector (\vec{r}_o), and the coordinates of the direction of the distance of the charge from the point of origin (\vec{r}_q), then calculates a vector (\vec{r}) where $\vec{r} = \vec{r}_o - \vec{r}_q$. From it, the unit vector of the field direction \hat{r} is calculated which determines the direction of the arrow representing the field direction

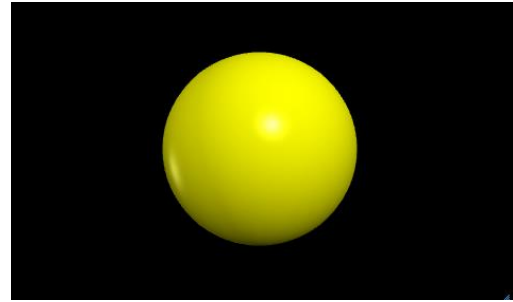
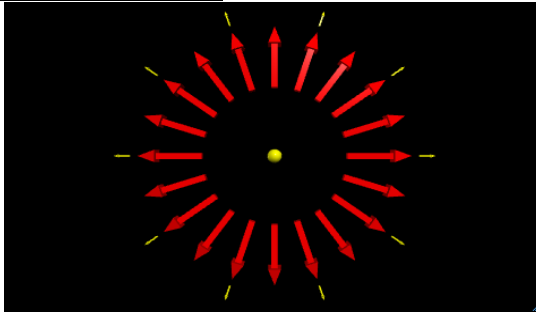
and equals $\vec{r} / |\vec{r}|$ where

$$\vec{r} = \text{norm}$$

$$|\vec{r}| = \text{mag}$$

They are built-in functions in Python. the same way simulation of an electric dipole Figure 1(b) [28-22-30] . This simulation needs three embedded objects. Two spheres represent positive and negative charges, and the arrows represent the field.

```
from vpython import
```



```
from visual import *  
sphere ()
```

Figure 1(a) shows the 3D model of the electric field of a positive point charge

```
from vpython import
```

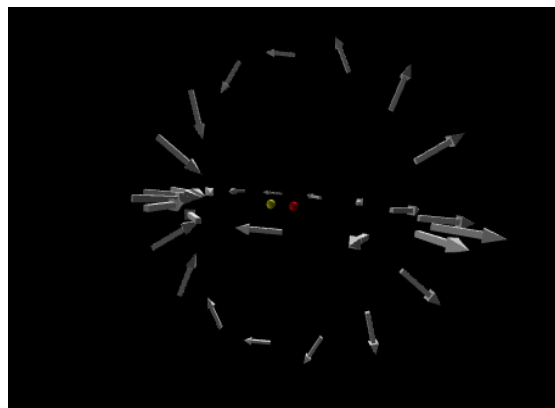


Figure 1(b). Screenshot of the 3D visualization of electric dipole by VPython

Simulation of the electric field of a charged rod, where the direction of the field depends on the fragmentation of the charge and the rod by using differential in theoretical calculations. By using VPython, students can create a realistic visualization and calculate the numerical value of the field. Where the rod consists of a group of balls that carry charges in the form. Figure 2(a) [28-22-25]

```
from vpython import
```

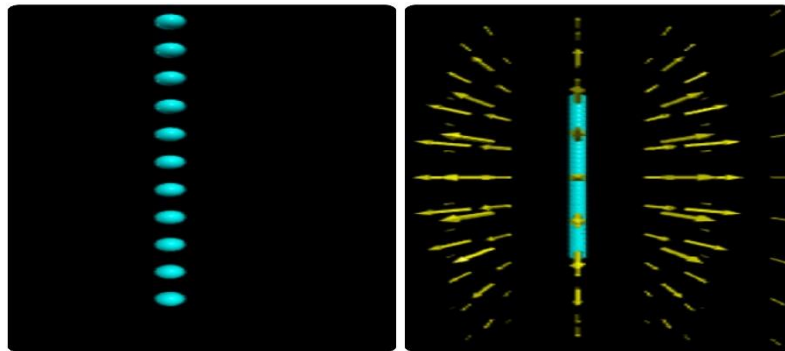


Figure 2(a) Simulation of the electric field of a charged rod. Screenshot of the 3D

Figure 2(b) shows

Simulation of the magnetic field generated by a charge moving rapidly through a wire. Two embedded objects are created, an arrow representing the direction of the charge current and the arrows representing the direction of the magnetic field. Where the simulator represents the base of the right hand to determine the direction of the magnetic field, where the arrows point to the fist of the hand and the arrow to the thumb. Using the left mouse button, the model moves in all directions. The students also calculate the value of the field strength by entering the values of the vectors, but by adding the velocity vector. Thus, the student compares the magazines of the point charge and the moving one. The full code is on the website.

```
from vpython import
```

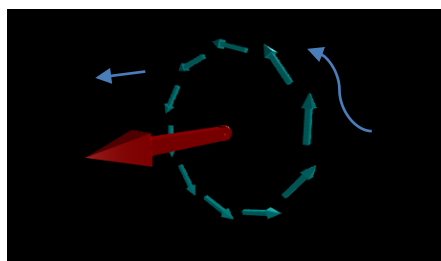


Figure 2b. Simulation of the magnetic field generated by a charge moving.

4 (B)-Simulation (Fig 3a) A simulation to study the movement of a particle falling to the ground and its bounce. Parameters such as initial position, final position, and gravitational acceleration have been introduced and they represent motion in its main parts, in addition to being basic parameters for describing the condition. Moreover, for a more dynamic visualization of the problem, the particle velocity graphs are constructed in y coordinates versus time (Fig. 3b), since python provides this type of construction. With these resources, the student can visualize the phenomenon in real time relatively completely, helping you to understand the relationship between motion and the equation responsible for characterizing it. To create the simulation, only two objects were used - a ball and a parallelepiped to represent the earth. It is indicated that the selection of the spherical geometry to represent the particle was due to its use in traditional physics approaches, and has negligible dimensions in simulation. Where the students introduced the gravitational acceleration of different surfaces such as the sun and the moon and compared the visible movement of the falling ball and the resulting graph of the rate of change of velocity with respect to time.

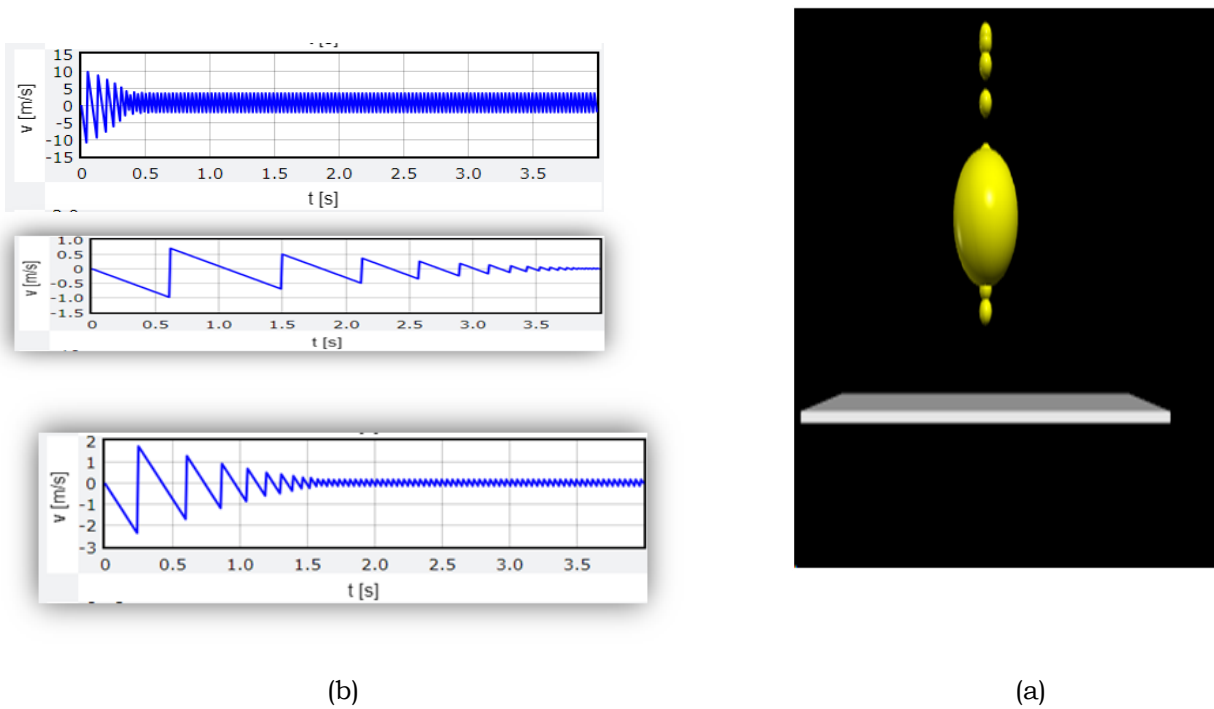


Figure 3(a). Screenshot of a 3D animation visualization based on VPython of the motion of a particle falling on the surface of the Sun Earth, Moon, . (b) the particle velocity graphs in y coordinates versus time

5-METHODOLOGY

In our research, we provided examples of developing programs to solve the problems of students who wanted to learn physics, and we used our knowledge of computer programming to solve the problem of a student who had learning problems in difficult subjects. [9] Our main goal was for students to approach difficult topics in their own way, and we made study simulations of several physical phenomena that could be completed. The goal of the simulation is to help students visualize electric and magnetic fields and their effects on particle trajectories. [10] All tasks had to be completed within two hours. Each activity includes a start page that describes the background of all the new VPython commands introduced in the activity. Students are not expected to use most of the commands, but a foundation is provided for more ambitious students. Each activity has two or three different icon areas on each page, and students gradually add additional features as the activity progresses. Each box contains code for at least one working program that requires modifications, or space to copy and paste previous code for later modifications. Tasks include calculating and representing electric field vectors in the vicinity of various charge distribution shapes. Drawing arrows with VPython is as easy as specifying where the arrow is and which direction it is pointing. The main goal of this process is to teach students how to correctly calculate the distance vectors required to find the electric field of a point charge and also learn how to find the electric field of charge distributions by summing the contributions. Small parts of the distribution. . The numerical integration of potential differences exercise allows students to calculate the electric field by factoring the charges and adding the contributions to the potential differences due to the different fields. Its purpose is to help students recognize connections between regions and possible differences. The other exercise is about particles moving in a magnetic field. Many students are surprised to see charged particles moving along spiral paths, even though they can easily point out that the magnetic force is perpendicular to the speed and magnetic field. Use overlapping electric and magnetic fields as a determinant of the speed of charged particles. The final activity demonstrates how students can use the program code to visually observe the rate of change of velocity as a function of time as the gravitational acceleration of a surface changes. This activity targets introductory modeling students and demonstrates the benefits of computational and conceptual modeling using VPython.

6-Discussion

Our development model is based on increasing students understanding of physics concepts through computational thinking activities and interactive visual modeling. In this study, we used the VPython programming environment because simulations can help promote CT by allowing users to adjust parameters and evaluate their predicted results. The findings contribute In a growing body of research, we find that arithmetic motivates students to search for new solutions. This study contributes significantly to the use of simulation in physics classrooms and, more precisely, to approaches to computational

thinking and includes creative experience with a wide range of principles of physics (with strong emphasis on the physical phenomena of vector quantities). It was taught by the college professor as part of the regular physics education. Computational physics is of great importance for understanding vector fields as major components of the electrodynamic cycle [31]. According to Charles Prosperiti and Tryggvason [3], learning the concepts and in-depth inquiries of fundamental physics depends on an understanding of computational physics. CT brings many benefits to professors and teachers—for example, as a large-scale simulation of transmission processes [32] and the driving forces of generators [33]. And physics experiment laboratories [34]. Regarding teachers' competency levels Zhang said et al. [16] A professor or teacher is considered not familiar with the causes of knowledge if he is ignorant of programming. Accordingly, teachers' limited knowledge of technology and its teaching methods must be expanded. In this study, we noticed that students were excited about visual thinking after the simulation and were able to imagine Solve and interact with the problem easily. The results support previous research [35–36], indicating that using simulations can help science students develop visual thinking skills such as observation and estimation. According to Vollmer [37], students' interactive visual abilities should be developed in order to help them visualize and delve into abstract events. Connect Yazo et al. [38] point out that in studies that focus on the use of computational modeling, more emphasis is placed on thinking skills. Many research studies [39–51] have identified Computer simulation benefits students by enhancing their' program interaction [52], improving their' performance when learning physics phenomena [53], and providing them with the opportunity to learn more aspects of physics using multiple representations [36, 54, 55].

7- Conclusions

In addition to being appropriate for college-level Physics instruction, the new 3D visual teaching simulation library VPython also satisfies the educational needs of other disciplines and has a wide range of applications. In overseas universities, interactive instruction is. The VPython program for interactive learning is being used in Derna and throughout Libya. It began a little later than in other nations, and there are no examples of its use. Simulators still need to be developed for a lot of material. There is still a lot of space for improvement in the actual process because it is not yet developed sufficiently. Particularly for academic instructors teaching labs in applied physics to study physical phenomena and build computer simulations, Visual Python is advised. It has been demonstrated that creating eye-catching interactive three-dimensional animations is simple in Visual Python. Making representations, simulations, or easy computer games of almost all physical processes is extremely helpful for supporting interactive education.

References

- [1]-Matthew N.O. Sadiku¹, Adebowale E. Shadare², Sarhan M. Musa Computational Physics: An Introduction International Journal of Engineering Research Volume No.6, Issue No.9, pp : 427-428
- [2]- R. H. Landau and M. J. Paez, Computational Physics: Problem Solving with Computers. New York: John Wiley & Sons, 1997.
- [3]- A. Prosperetti and G. Tryggvason, "Appendix 3: Report of study group on computational physics," International Journal of Multiphase Flow, vol. 29, 2003, pp. 1089–1099.
- [4]- Ergi Bufasi, Kawtar Lakrad, Improving Teaching Techniques Using Visual Python: A Case Study In Physics Laboratories, International journal of scientific & technology research: Volume 8, Issue 12, December 2019
- [5] -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
- [6]- Olenick R. P. –Visual Python in a computational physics course! PyCon, Addison, TX(2007).
- [7]-Todd A. Zimmerman. Computational Modeling in Introductory Physics Courses and Across the Curriculum American Society for Engineering Education, 2020
- [8]-M. Caballero, "Integrating computation into core physics courses: A perspective from a research-intensive university," Bulletin of the American Physical Society, vol. 64, 2019.
- [9]-Beazley D. M., Python Essential Reference, Sams Publishing, 2006
- [10] - VPython." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. Retrieved 13 June, 2011
- [11]- VPython, <http://www.vpython.org/>, Retrieved 13 June, 2011
- [12]-Gambrell, J., & Brewe, E. (2023). Developing and Validating a Computational Thinking Assessment Instrument for Introductory Physics. *Bulletin of the American Physical Society*
- [13]- Lonsky, M., Lang, M., Holt, S., Pathak, S. A., Klause, R., Lo, T. H., ... & Fangohr, H. (2023). Developing computational skills through simulation based problem-solving in science. *arXiv preprint arXiv:2303.01784*.
- [14] - A. Kononov, P. Bellon, T. Bretl, A. Ferguson, G. Herman, K. Kilian, J. Krogstad, C. Leal, R. Maass, A. Schleife, J. Shang, D. Trinkle, and M. West, Computational curriculum for MatSE undergraduates, 2017ASEE Annual Conference & Exposition Proceedings 10.18260/1-2{28060 (2017).
- [15] -M. D. Caballero, N. Chonacky, L. Engelhardt, R. C. Hilborn, M. L. del Puerto, and K. R. Roos, PICUP: A community of teachers integrating computation into undergraduate physics courses, The Physics Teacher 57, 397 (2019)

- [16] -X. Zhang, A. Schleife, A. Ferguson, P. Bellon, T. Bretl, G. Herman, J. Krogstad, R. Maass, C. Leal, D. Trinkle, J. Shang, and M. West, Computational curriculum for MatSE undergraduates and the influence on senior classes, 2018 ASEE Annual Conference & Exposition
Proceedings 10.18260/1-2{30213 (2018)
- [17] -R. Chabay and B. Sherwood, Computational physics in the introductory calculus-based course, *American Journal of Physics* 76, 307 (2008)
- [18] -H. Fangohr, Exploiting real-time 3d visualisation to enthuse students: A case study of using visual python in engineering, in *Computational Science { ICCS 2006* , edited by V. N. Alexandrov, G. D. van Albada, P. M. A. Sloot, and J. Dongarra (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 139{146
- [19]- B. Tversky, J. B. Morrison, and M. Betrancourt, "Animation: Can it facilitate?," *Int. J. Hum.-Comput. Stud.* 57, 247–262 2002
- [20]-Countryman, C., Mburu, T. K., & Rodelli, L. (2023). Teaching Electric Fields with a Simulation and Game. *The Physics Teacher*, 61(3), 170-171.
- [21] -Atherton, T. J. (2023). Resource Letter CP-3: Computational physics. *American Journal of Physics*, 91(1), 7-27.
- [22] -Available at https://trinket.io/todd_zimmerman_phd_gmail_com/courses/second-semester-introductory-physics-electricity-and-magnetism#/getting-started-with-vpython/why-vpython.
- [23] -Weller, D. P., Bott, T. E., Caballero, M. D., & Irving, P. W. (2021). Developing a learning goal framework for computational thinking in computationally integrated physics classrooms. *arXiv preprint arXiv:2105.07981*
- [24]- Nowotniak, R., Draus, C., Nowak, M., & Rybak, G. (2011). Modelling Reality in Visual Python
- [25] -Wang, J., & Wang, A. (2023). Interactive Computing and Visualization. In *Introduction to Computation in Physical Sciences: Interactive Computing and Visualization with Python™* (pp. 43-77). Cham: Springer International Publishing
- [26] -Sands, D. (2010). First year mechanics taught through modelling in VPython. *New Directions in the Teaching of Physical Sciences*, (6), 47-50.
- [27]- Morgan, W. A. and {English}, L. Q. "VPython in Introductory Mechanics and Astronomy", *American Astronomical Society Meeting Abstracts* \2352020 {367.09},{Provided by the SAO/NASA Astrophysics Data System
- [28] - <https://trinket.io/glowscript/742aa12b37>
- [29] Loo, G., & Jianhong, C. (2010). VPython Application To The Computer-aided Drug Design Problem. *Python Papers Monograph*,

- [30] - Wu, T., Wang, Q., & Wang, S. (2021). Research on the application of VPython 3D dynamic visualization in the teaching of college physics. *Journal of Contemporary Educational Research*, 5(9), 1015
- [31] A. Budi and D. Mulyati, "Discovering and understanding the vector field using simulation in android app," *Journal of Physics: Conference Series*, vol. 1013, no. 1, pp. 012062, 2018. <https://doi.org/10.1088/1742-6596/1013/1/012062>
- [32] M. Ye, X. Jiang, S.-S. Li, and L.-W. Wang, "Large-scale first-principles quantum transport simulations using plane wave basis set on high performance computing platforms," *Computer Physics Communications*, vol. 260, pp. 107737, 2021. <https://doi.org/10.1016/j.cpc.2020.107737>
- [33] F. Bakri, D. Sumardani, and D. Mulyati, "The augmented reality application for simulating electromotive force concept," *Journal of Physics: Conference Series*, vol. 1402, pp. 066039, 2019. <https://doi.org/10.1088/1742-6596/1402/6/066039>
- [34] E. Bufasi and K. Lakrad, "Improving teaching techniques using visual Python: a case study in physics laboratories," *International Journal of Scientific & Technology Research*, vol. 8, no. 12, 2019
- [35] M. Droui and A. El Hajjami, "Simulations informatiques en enseignement des sciences: apports et limites," *EpiNet: revue électronique de l'EPI*, vol. 164, 2014.
- [36] M. Ben Ouahi, M. Ait Hou, A. Bliya, T. Hassouni, A. Ibrahmi, and E. Mehdi, "The effect of using computer simulation on students' performance in teaching and learning physics: are there any gender and area gaps?" *Education Research International*, vol. 2021, 2021. <https://doi.org/10.1155/2021/6646017>
- [37] G. Fulmer, "Secondary Students' Visual-Spatial Ability Predicts Performance on the Visual-Spatial Electricity and Electromagnetism Test (VSEEMT)," *Edoc.hu-berlin.de*, 2022. Available: <http://dx.doi.org/10.18452/8211>
- [38] E. Taslibeyaz, E. Kursun, and S. Karaman, "How to develop computational thinking: a systematic review of empirical studies," *Informatics in Education*, vol. 19, no. 4, pp. 701–719, 2020. <https://doi.org/10.15388/infedu.2020.30>
- [39] G. Fessakis and S. Prantsoudi, "Computer science teachers' perceptions, beliefs and attitudes on computational thinking in Greece," *Informatics in Education*, vol. 18, no. 2, pp. 227–258, 2019. <https://doi.org/10.15388/infedu.2019.11>
- [40] B. Upadhyaya, M. M. McGill, and A. Decker, "A Longitudinal Analysis of K-12 Computing Education Research in the United States: Implications and Recommendations for Change," in *Proceedings of the 51st ACM Technical Symposium on Computer Science*

- Education, 2020, pp. 605–611. <https://doi.org/10.1145/3328778.3366809>
- [41] K. El Kharki, F. Bensamka, and K. Berrada, “Enhancing practical work in physics using virtual javascript simulation and LMS platform,” In *Radical solutions and eLearning*, Springer, pp. 131–146, 2020. https://doi.org/10.1007/978-981-15-4952-6_9
- [42] E. D. Agyei and D. D. Agyei, “Promoting interactive teaching with ICT: features of intervention for the realities in the Ghanaian physics senior high school classroom,” *International Journal of Interactive Mobile Technologies*, vol. 15, no. 19, 2021. <https://doi.org/10.3991/>
- [43] X. Fan, D. Geelan, and R. Gillies, “Evaluating a novel instructional sequence for conceptual change in physics using interactive simulations,” *Education Sciences*, vol. 8, no. 1, pp. 29, 2018. <https://doi.org/10.3390/educsci8010029>
- [44] L. Nadiradze, M. Kapanadze, and B. Kvirkvelia, “Use of Technologies, as the Effective Instrument for Enhancing of Motivation in the Process of Physics Teaching,” in *INTED2020 Proceedings, 14th International Technology, Education and Development Conference, 2020*, pp. 2768–2773. <https://doi.org/10.21125/inted.2020.0828>