Migration Letters

Volume: 20, No: S3(2023), pp. 324-331

ISSN: 1741-8984 (Print) ISSN: 1741-8992 (Online)
www.migrationletters.com

Computational Physics: Challenges of University Education

Franyelit Suárez-Carreño[1], Luis Rosales Romero[2], Jose De La Paz-Ramos[3]

Abstract

This paper proposes an analysis of physics teaching in introductory university-level courses using programming techniques and computational tools. Little needs to be studied, such as the difficulties in learning computing to solve physics problems in a science classroom or how to help students overcome these challenges. This paper explores these challenges from the perspective of students and faculty in a physics classroom with a computer-integrated curriculum. The experience was put into practice in two classrooms taking advantage of the experience of the two teachers as researchers and with extensive programming knowledge. The results are mainly based on surveys to focus on the perspectives of students and teachers. Among other results, it is concluded that learning physics is more realistic for some students familiar with programming and software management. In some cases, it can generate frustration and aversion to physics in students who have yet to gain experience in software management.

Keywords: *Physics teaching, Computing, Programming, Learning.*

Introduction

There are growing and widespread efforts to introduce computing into teaching strategies. There are diverse discussions and consensuses (Hamerski, McPadden, Caballero, & Irving, 2022)(Fidan & Tuncel, 2019)(Bodin & Winberg, 2012) that integrating programming and computing into teaching gives students a more realistic view of what it means to do science, and better prepares students to pursue careers in a world where computing represents an essential element for a wide variety of applications. The interest in integrating computing into classrooms becomes more frequent with the implementation of virtual platforms during the pandemic. This work contributes to the effort to deepen the perspective of students and professors on the use of computing in the teaching of university physics (Weintrop, et al., 2016).

Computational integration should involve changing a course's curricula and curriculum to incorporate computational modeling. In this way, students learn to program together with learning science, in a new way, through simulation and computational modeling. Computing provides an avenue for discipline over the scientific practices to be learned in the classroom (Fennell, et al., 2019)(Schweinle, Meyer, & Turner, 2006)(Caballero, Fisler, Hilborn, Romanowicz, & Vieyra, 2020)(Irving, McPadden, & Caballero, 2020)(Kapon, Laherto, & Levrini, 2018).

[1] Universidad de lasAméricas, Facultad de Ingeniería y Ciencias Aplicadas, Carrera de Ingeniería Industrial. Quito, Ecuador, franyelit.suarez@udla.edu.ec, http://orcid.org/0000-0002-8763-5513

[2] Universidad Nacional Experimental Politécnica Antonio José de Sucre, Vice Rectorado Puerto Ordaz, Venezuela, luis.rosales2@gmail.com, https://orcid.org/0000-0002-7787-9178

[3] Universidad José Carlos Mariátegui, Moquegua – Perú, jdelapaz@ujcm.edu.pe, https://orcid.org/0000-0003-1096-1457

Students in the first semester courses at universities often experience psychological stress associated with developing new problem-solving methods. From a broader perspective, some studies have highlighted some associated barriers when using computational techniques to teach physics in university introductory courses. The main computational difficulties encountered in terms of a skill set are the elaboration of algorithms and codes; this includes syntax, semantics, structure, and style uses. (Gupta, Elby, & Danielak, 2018).

The application of computational tools in the study and teaching of physics requires the mastery of logical-mathematical skills and some degree of experience with translating ideas into algorithms and codes. Both are difficult to master quickly because the computer provides intricate comments or absolute rejections. Much of the research on students' experiences with programming focuses on the challenges they face. Some authors compiled studies focused on learning difficulties in handling a programming language. Students are concerned about learning syntax, variables, error messages, and understanding code (Bosse & Gerosa, 2016).

An essential feature is that instructors should ensure that initial experiences with computing should include some students achieving success. It is easy to get the wrong programming on the computer if you need to learn how to interact with the compilers and the expected messages when errors are detected in the code.

A recent study raises how computational activities might be challenging in education. Learning a general domain programming language and then using it for scientific modeling is a significant challenge. They found that certain features, such as the problem-solving process and syntactic complexity of programming languages, can be harnessed for learning (Jenkins, 2002).

In an investigation into the impact of a Python-based university-level computational integration, it was found that students were enthusiastic about learning computing, although the integration did not have a significant benefit for learning, until students had learned computational tools and were able to leverage their proficiency with certain lab tools and data analysis techniques. (Malmi, Sheard, Kinnunen, Simon, & Sinclair, 2020)(Basu, et al., 2016).

Some research highlighted several other benefits that computing brings to the study of physics, they focused their work on scenarios where modeling is used and argued that calculus highlights relationships between physics concepts, creates dynamic visual models, and can be used to explore complex real-world physics problems due to its computing power. In addition, they explained that students who use computing are learning to use tools that scientists use, which makes learning physics more interesting. One of the problems of computational modeling in physics teaching is the great effort for students to invest time to become familiar with the software. (Serbanescu, Kushner, & Stanley, 2011)(Brewe, 2008)(Knight M., 2015)

Materials and Methods

This work is described as a case study because of the variation in data sources and because the goal is to study the computational experiences of college students. In particular, an interpretive perspective is adopted by focusing on students, teachers, and their perspectives.

The analytical and interpretive approach lends itself well to studies focusing on how people experience and interpret a phenomenon instead of the phenomenon itself. The aim is to explore how students assume the introduction of computational tools in their physics classes. An interpretive case study is ideal for exploring this deeply and qualitatively.

When determining the data sources, the reality of the case study is delimited to the students themselves and the teacher within the classroom (Dyson & Genishi, 2005).

Surveys of students and teachers revolved around classroom activities, in how the teacher had integrated computing into physics class. Teachers must relate their experiences so that the educational community and researchers have their conclusions and opinions about the work.

The work was carried out in the fundamental physics courses taught during the second semester of 2021, taking importance in the experience as researchers and expert programmers of the teachers of the physics subjects of the introductory courses. First, they were taught the theoretical concepts of physics, then the computational activities and the use of programming in Matlab language (Irving, Obsniuk, & Caballero, 2017).

The work involved developing and explaining a computer program with applications of a random physics topic to the students. The preliminary explanation is error-free and accurately executed for a demonstration to students. However, for the practical execution, the teacher delivers a code with certain erroneous or incomplete lines, which must be attended to by the students, following the instructions given and considering the desired output. The idea is that students knowing the physical theories previously taught, and having a base of the code, can correct and improve the errors of the program so that it runs without failures and the required physical solutions can be obtained.

Survey protocols were developed for the two groups selected for the study. The questions aimed to obtain and discuss their perceptions about the physics class and the computational strategies implemented. Also, notes were taken on the observations in the classroom of both groups of students working on the computational activity during the class.

Results

Stress in students accompanies new physics learning experiences using computational strategies in problem-solving. When students had a mastery of theory and relevant physics concepts, computation forced them to think to translate their knowledge of physics into a computer language. This experience was accompanied, in some cases, by frustration. Both frustration and stress make some students feel unprepared to take on this new challenge in teaching and learning physics.

Other students resisted learning new tools for solving physics problems, and it was observed that instructors could not devote enough time for students to get used to a programming language within the physics subject. Several students stated that the computational activities were too complicated, so they usually copied another student's code.

Some students manifested programming skills, but it was a minority. In the problem-solving aspect of physics, they demonstrated mastery and skills when implementing the algorithms. These activities involved more than just learning physics. It was about building new skills and letting the students' creativity shine.

Not all physics topics are adapted, so solutions to problems are made by implementing a computer program. The criteria evaluated for classroom implementation are shown in Figure 1.

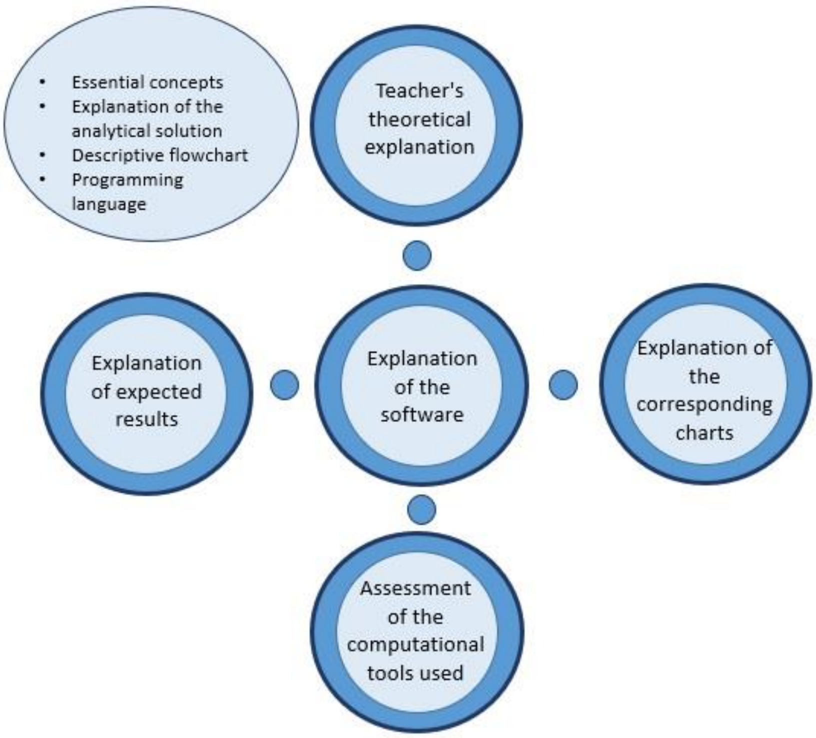

Figure 1. Criteria to be considered in the evaluation of the practical application carried out.

Source: Own

The work was carried out with a group of students of the subject of mechanical physics with 16 students, and another group with an activity of electromagnetism, with 23 students. The previous results are shown in Tables 1 and 2. The methodological procedure was the same in both groups.

Table 1. Results of the survey-interview of the group with the activity of mechanical physics.

Number of students	Activity
12	They used algorithms to find the analytical solutions.
3	Previous programming experience.
9	Experience in commercial software management.
8	They accepted the challenge of programming to find the solutions.
14	Interested in learning a programming language.
8	Interested in developing codes to solve problems.
10	Generated additional stress.
7	They were not interested in incorporating programming into their physics learning.
14	They tried, but were frustrated by not achieving results in the expected time.
1	Experience in projectile launch simulation.

For the 23 mechanical engineering students evaluated on the use of programming in the simulation of problems related to electromagnetism, the results obtained through the assessment tool can be observed in Table 2.

Table 2. Results of the survey interview in the group with the activity of electromagnetism.

Number of students	Activity
17	They used algorithms to find analytical solutions.
7	Previous programming experience.

11	Experience in commercial software management.
17	They accepted the challenge of programming to find the solutions.
23	Interested in learning a programming language.
17	Interested in developing codes to solve problems.
8	Generated additional stress
9	They were not interested in incorporating programming into their physics learning.
8	They tried but were frustrated by not achieving results in the expected time.
2	Experience in solving electromagnetism problems

In both cases, the professor has experience in research and programming. Becoming an expert in programming is like learning to communicate with a new language. With proper guidance or computational experience, students can explore questions and build code. Experience that will undoubtedly mark his professional life in the future.

Another common feature in the surveys was how the programming made you forget that the problem was physics. Some students stopped thinking about physics to develop computational skills, which led to feelings of inadequacy in physics or computing.

Some students who experienced stress and frustration stated that programming is difficult, even more so when used in the teaching-learning process of physics. Stress is usually caused by interpreting codes and errors when compiling the program. This has previously been documented with students learning physics through Python (Kennedy, 2002).

Interpretations of Implementation

In surveys with students, they showed motivation and experienced it for themselves. For example, they argued that the purpose was to understand better programming concepts, which helped her see the connection between equations and actual physical phenomena.

Some students presented the benefits of computing, emphasizing that one of those benefits is the visualization and strengthening of physics concepts. Others expressed a similar perspective: translating ideas into codes was a way to learn physics concepts. While others focused on the benefits of interacting with code, others discussed how creating code was constructive to their formation. Some students demonstrated to themselves that they understood the value in meticulously translating physical equations into computer codes and incorporating software feedback. This allowed them to participate in the activities in a way that they felt helped them learn physics.

Discussion

From surveys of students, it is inferred that they faced a variety of difficulties and new challenges when computing was integrated into their physics classes. Some related specifically to the programming language, compilers, program runs and interpretation of codes and development of solutions.

It is normal for barriers to exist in introducing the learning level of computing to solve physics problems, focusing mainly on the additional skills that students need to learn, such as knowledge of computing and elements associated with programs or codes, such as syntax, semantics, and algorithms.

Relevant skills were conceptual understanding of physics, pseudocode writing, computational thinking, connecting ideas between mathematics, physics, and computing, understanding the purpose of using computation beyond analytical problem solving, and learning programming. How to write comments on your code. Several researchers interviewed in this study were self-taught programmers, further demonstrating that there is a need for these types of skills to be introduced into physics curricula.

It was also observed that students experienced nervousness when coding the solutions and linking them as functions and parameters, often resulting in students seeing this as a problem. For example, when students realize that their code contains an error, they are prone to give up and not finish programming. In addition, they worry that they take a lot of time and effort to identify and correct.

A much talked about part of surveys is understanding the syntax and how it binds to error messages and strategies to address them. These are new skills that students don't have unless they've previously taken a programming or simulation class.

Computing can help some students generate interest in physics. Classroom teachers for study establish that computational programming is an opportunity to connect with physics more authentically.

Conclusions

This article describes the challenges science and engineering students face in an introductory physics class taught using computational tools. The main feelings found before the change of attention to physics problems were stress and frustration, demotivation with the physical subject, and difficulties in elaborating, implementing, and interpreting numerical solutions. Connections were also found between student descriptions, the teacher's way of delivering classes, and student interest.

While this study is a first step, a deeper study is needed on the difficulties students face and how to support them in what is intended by teaching physics courses differently than the traditional, incorporating computational tools.

It is necessary that engineering students understand the needs of integrating computing in a large part of the subjects, since industrial needs point to the use of digital tools for the development of automated systems with scientific applications.

An example of the importance of student perspectives was that some expressed that they were uncomfortable with coding when the program was being developed. Others were more willing to learn a programming language or had some previous experience in software management. For researchers, this study is a call to action to prepare future researchers. Computer physics teaching courses continue to be a significant challenge as they become synonymous with doing science by learning.

With this come the complexities and difficulties of the new way of teaching physics to college students. The curriculum needs to adapt it to the new schemes and new times in the digital age and the need to understand the experiences of students in this new environment. There is a need for further exploration, particularly in how the integration of computing into physics takes shape and how the difficulties and frustrations of learning a new programming tool affect students.

For university institutions, this study is a call to consider many factors when designing or modifying the curriculum for computational integration in physics education, particularly programming and computer code-building. Pedagogical strategies for teaching computing, which means redesigning existing curricula.

Computing in physics courses is essential for the next generation of scientists, and the teacher must understand the application of computation as a necessary tool for the student.

There is a need to develop accessible approaches to computational models that reflect science so students can do science using calculation tools. Study how computing changes problem-solving attitudes promotes learning standards by implementing computational integration in physics teaching, and supports teachers as content developers.

References

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. Research and Practice in Technology Enhanced Learning .

Bodin, M., & Winberg, M. (2012). Role of beliefs and emotions in numerical problem solving in university physics education. Physical Review Special Topics – Physics EducationResearch.

Bosse, Y., & Gerosa, M. (2016). Why is programming sodifficult to learn: Patterns of Difficulties Related to Programming Learning Mid-Stage, . ACM SIGSOFT Software Engineering Notes , 41.

Brewe, E. (2008). Modeling theory applied: Modeling Instruction in introductory physics, . American Journal of Physics , 25-37.

Caballero, M. (2015). Computation across the curriculum: What skills are needed. Proceedings of the Physics Education Research Conference (pp. 102-111). MD: College Park.

Caballero, M., Fisler, K., Hilborn, R., Romanowicz, C., & Vieyra, R. (2020). American Association of Physics Teachers. The American Association of Physics Teachers, College Park.

Dyson, A., & Genishi, C. (2005). On the Case: Approaches to Language and Literacy Research. Teachers College Press. New York: Teachers College Press.

Fennell, H., Lyon, J., Magana, A., Rebello, S., Rebello, C., & Peidrahita, Y. (2019). Designing hybrid physics labs: combining simulation and experiment for teaching computational thinking in first-year engineering. IEEE.

Fidan, M., & Tuncel, M. (2019). Integrating augmented reality into problem based learning: The effects on learning achievement and attitude in physics education. Computers & Education, 142.

Gupta, A., Elby, A., & Danielak, B. (2018). Exploring the entanglement of personal epistemologies and emotions in students' thinking, . Physical Review Physics Education Research, 14.

Hamerski, P., McPadden, D., Caballero, M., & Irving, P. (2022). Students' perspectives on computational challenges in physics class. Physics Education, 1-27.

Irving, P., McPadden, D., & Caballero, M. (2020). Communities of practice as a curriculum design theory in an introductory physics class for engineers. Phys. Rev. Phys. Educ. Res, 16.

Irving, P., Obsniuk, M., & Caballero, M. (2017). P3: A Practice Focused Learning Environment. European Journal of Physics, 055701.

Jenkins, T. (2002). On the Difficulty of Learning to Program . 3rd Annual LTSN-ICS Conference Proceedings, (pp. 53–58). London, United Kingdom.

Kapon, S., Laherto, A., & Levrini, O. (2018). Disciplinary authenticityand personal relevance in school science. Science Education , 102.

Kennedy, K. (2002). Top 10 Smart technologies for Schools: Artificial Intelligence,. Retrieved from https://www.youtube.com/watch?v=MMjVjirQuFM

Malmi, L., Sheard, J., Kinnunen, P., Simon, K., & Sinclair, J. (2020). Theories and Models of Emotions, Attitudes, and Self-Efficacy in the Context of Programming Education,. Proceedings of the 2020 International Computing Education Research Conference,. New Zealand.

Schweinle, A., Meyer, D., & Turner, J. (2006). Striking the Right Balance: Students' Motivation and Affect in Elementary Mathematics. The Journal of Educational Research, 99.

Serbanescu, R., Kushner, P., & Stanley, S. (2011). Putting computation on a par with experiments and theory in the undergraduate physics curriculum,. American Journal of Physics.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computationa Thinking for Mathematics and Science Classrooms. Journal of Science Education and Technology , 25.