TEACHING PHYSICS THROUGH CODING

S. Arabasi

German Jordanian University (JORDAN)

Abstract

Interactive learning has been proven to be more effective. It helps students stay engaged and interested. Coding is a necessary skill that all students need regardless of their field. It helps them visualize the problem and hence understand it more deeply in some cases. In other cases, it is the only way to solve a problem. Undergraduate students usually have a programming language compulsory course in their first year. They are taught a certain programming language from a computer science point of view. They rarely use practical applications in their code and they finish the course wondering about the actual use of such a programming language.

In this paper, we suggest to teach students some elementary coding at the beginning of the class. Afterwards, we ask them to solve some physics problems through coding as the class moves along. This will show the practical use of their programming skills early in their academic life. At the same time, it will enhance their learning. This method has been tested in an introductory physics class and the results of its effect on the general understanding of different topics show an improvement. We believe that this could be the future of teaching, not only in physics but also in different fields of science.

Keywords: Learning technologies, interactive learning, learning through coding.

1 INTRODUCTION

The advent of smartphones and their applications generated interest in coding and programming among young generations. University students nowadays are not only tech-savvy, but also have a keen interest in coding and an overall general knowledge that was not available to previous generations. Schools are trying to catch up with this new digital trend by investing into the necessary infrastructure that will ensure that future generations have what they need to stay relevant in today’s market.

Coding is an essential part of every engineering curriculum. A programming language is usually taught to students in their first year. Schools spend a tremendous amount of money buying licenses to different software that are necessary tools to every engineer. There has been a lot of work lately to encourage students to learn programming skills and investigate their effect on reflective thinking skills towards problem solving [1]. It has been found that when practices emphasize learning by doing together with continuous feedback to be the most efficient means for learning programming skills [2]. Some have promoted social and creative dimensions in learning to code [3]. Duke et al [4], argue that the overall aim of first programming subjects is to encourage students to explore, experiment, experience and finally extend. The goal is to make them explore the art and pleasure through an experiential approach [5]. Students can be equipped with problem solving skills to be used to solve real world problems in a variety of fields along with the computing goals [6], [7], [8]. Nowadays, there are limitless resources that teach you how to code. Many of them are free of charge. Some examples are: freecodecamp.org [9], coursera.org [10], and codeconquest.com [11].

In this paper, we put all the aforementioned factors and use them during a general physics course in mechanics. The benefit is dual; students learn how to code while learning physics. This will help solve real world problems and help the students interact with the topic while learning how to code.

2 METHODOLOGY

Most students that we receive have had a minimum level of experience in programming. The smartphone revolution which became an essential part of every student’s life, made coding to be trendy and fashionable. New students are more acceptable of the idea that programming is very essential to their engineering studies than, say, a decade ago and, hence, they are more willing to learn a new programming language if it will help advance their careers.
In this study, students were asked to solve some physics problems by writing a MATLAB code. MATLAB is licensed in our university for all students and hence getting the software was no hurdle for students. We emphasized that MATLAB, although powerful, is not the only software to be used. Many other open source software options, such as Python, are available and the student is encouraged to look into them.

Students first are introduced to MATLAB in class and some examples were shown to them. The following problems are given to the students and are asked to solve them by writing a code:

1. An object is moving such that its position changes according to the following equation: \( x = 2t^2 - 4t + 5 \), plot the position versus time graph, then find the turning point on the graph. Write some code to solve for the turning point. Plot the velocity versus time graph and determine the turning point on the graph. Finally, plot the acceleration versus time graph.

2. Two cars A and B. A moves with a constant velocity while B moves with a constant acceleration. Car A passed car B and immediately car B, which was initially at rest, started moving. When does car B overtake car A? Plot the graph of position versus time for both cars, and find their intersection on the graph. Write some code to find this intersection point without graphs.

3. An object is tossed upwards with an initial velocity of 4.9 m/s. Determine the time it takes the object to reach a height of 1 m. Explain why you have two solutions.

4. A projectile is thrown with an initial velocity of 10 m/s at an angle of 53°. Find its horizontal and vertical position after half a seconds. Find its maximum height and range. When the object is 2 m high how far is it horizontally from the launching point.

5. Show that two projectiles launched at two angles that are complimentary will have the same range.

6. Calculation and analysis of kinetic friction force of an object sliding on incline plane.

7. Determination of static friction force of a car moving with constant speed on a uniform circle.

These exercises are, although carefully chosen, yet not the only ones that could be used. Any physics problem can be used. However, we used problems that students are familiar with. Most of these problems have been covered in class or in the homework. Hence, the physics side of the problem is somehow familiar to the student.

A survey has been given to participating students after finishing the exercise. The response to each question ranges between strongly disagree to strongly agree.

1. The exercise has helped me understand the physics behind the problem more deeply

2. Coding has helped me gain some insight about the real problem

3. This exercise has made programming more fun and relevant

4. Coding method is more efficient (in comparison to the classical way of teaching) to understand physics problems

5. I would like this method to be used in other science courses

These questions help us understand students’ attitudes towards our practice and probe into how effective this whole exercise is.

Our aim is to engage the student into the physical problem and to create links between the physics world and the programming world, a world they can easily relate to and feel they can use to their advantage.

3 RESULTS

As mentioned earlier, the student has seen some or all of these problems at some point in the class. We believe that by coding these problems we enforce the concept and force the student to think deeply about the problem gaining some insight along the process. In this section, I will summarize the students experience and attitude towards this exercise.

Most students have a hard time understanding graphical interpretations of physical quantities. Generating a graph helped them understand what the graph represents. Some students were forced to question their concept and think deeply to have a problem solved. In general, students’ attitudes
towards programming were positive if it helps solve a real problem. They struggled with syntax learning at the beginning but with many tutorials available online they were able to overcome these problems.

The key outcome of this exercise is that we forced the student to think critically. First year students usually think that they are not ready yet to do something on their own. They often ask for guidance even when it is something that has been repeated in class. This exercise has forced them to take charge and think on their own. It has taught them independence.

The sample of students is not large enough to have a statistical significance. The number of students who volunteered for this exercise was 10 students with only 5 fully committing. However, it was an indication that such an exercise is indeed productive. All of the participants strongly agreed that such an exercise helped them develop a deeper understanding of physical concepts. All of them recommended that such an exercise to be used in other science courses.

Two students in particular have shown keen interest in the whole process and have written an advanced code. One had absolutely no knowledge MATLAB prior to this exercise, yet she was able to learn the basics and write her own code regarding motion in one dimension (Problem 1). The other one has had a previous, yet not very deep, knowledge of MATLAB. He took this exercise a step further and developed a whole animation of projectile motion (Problem 4). Both students have shown real growth in their understanding of kinematics. The resulting outcome of their code is shown in Fig. 1 and Fig. 2, respectively.

![Figure 1. Coding outcome of student 1.](image1)

![Figure 2. Coding outcome of student 2.](image2)
4 DISCUSSION

The goal of this paper is to introduce such a practice in physics and science classes for first year students. We absolutely encourage coding and programming to be a part of every science course. It is a powerful tool that can pay dividends when students advance in their studies. We believe that introducing such a practice in physics courses, not only will help students learn their physics, but also will help them develop their programming skills and show them some practical use of coding.

We believe that students’ understanding has to grow from a place familiar to them. This place in our case is starting from something simple and then developing a more advanced scheme. The pedagogical process takes the following path:

- Analyse the physical problem
- Understand the governing equations
- Determine the dependent variable and independent variable(s)
- Turn the governing equations into lines of code
- Determine the outcome of your code, whether it is a specific value, a graph or an animation
- Does the outcome of your code match your expectation? Analyse and question the outcome of your code
- Link your physical understanding to what you see as the outcome

5 CONCLUSIONS

In this paper, we emphasize the importance of teaching students the programming skills they ought to learn during their engineering education in their first year. We suggest that this should be done during their very first science classes. Different physics problems were given to students and were asked to write a software program to solve the problem. These problems were chosen carefully so they can add to their basic understanding of the topic while coding the problem. By going through the process, student will develop a deeper understanding of the physical problem, gaining some critical thinking skills along the way. A survey has been handed to the students to indicate their attitude towards this method. Students have shown improvement in their understanding of the physics concepts involved and expressed a positive attitude towards the use of this method other science classes.

ACKNOWLEDGEMENTS

The author would like to thank the deanship of scientific research and graduate studies at German Jordanian University for their help and financial support.

REFERENCES


