ASEE 2014 Zone I Conference, April 3-5, 2014, University of Bridgeport, Bridgpeort, CT, USA.

# Introducing Programming into the Physics Curriculum at Haverhill High School Using the R Language

Katherine Aho and Kavitha Chandra
Department of Electrical and Computer Engineering
University of Massachusetts Lowell
Lowell, MA, USA
Katherine_aho@student.uml.edu;
kavitha_chandra@uml.edu

Ed Roberts
STEM Academy
Haverhill High School
Haverhill, MA, USA
eroberts@haverhill-ps.org

***Abstract*—Computer programming with the R language was integrated into the physics curriculum at Haverhill High School in Haverhill, MA. The goal of this initiative was to improve students' critical thinking skills through the challenges of writing code. This paper provides an overview of how the R language was introduced into the classroom. We will discuss the design of the programming lesson modules and how they are incorporated into the physics curriculum. The difficulties, successes, and failures of this experience will be presented.**

***Keywords—R programming; high school; physics; curriculum integration; K-12 education; computing***

## I. INTRODUCTION

Technology is in abundance in our everyday lives- tablets, smartphones, social media. It is becoming increasingly important to be able to best utilize these devices. This technology is also becoming a central core of our jobs. Programming is now an invaluable skill that employers are seeking in their new job applicants, especially in the fields of Science, Technology, Engineering, and Math (STEM). With technology being the forefront of out daily lives, STEM jobs will only increase. In 2010, there were 7.6 million workers in STEM fields in the United States and STEM jobs are projected to increase by 17% by 2018 [1]. Because of this, it is now even more important to expose students to writing computer programs. Not only does it prepare the students for the new job market, it also teaches the students critical thinking skills that they otherwise may not learn through traditional high school curriculum. The critical thinking skills obtained through the logical process of writing code and computational methods will better equip the students as they enter college and careers [2],[3].

Many institutions have implemented the use of computers, programming, and computational methods into science education since the 1980s [4]. In early studies, Wise [4] found that activities with computers improved student performance. For example, one study showed that using computers to visualize data enhanced the students ability to recognize trends and patterns in graphs. Wilson and Redish [5] acknowledged how the computer revolutionized physics research and investigated how to use computers as tools to emphasize physical concepts. They tested the using computers in physics courses at the University of Maryland and concluded that the students scored better on laboratory reports and felt more satisfied with the nature of the work. They also stated that similar approaches could be used to improve high school level courses.

More recently, the focus has shifted from the use of computers to computational methods in the physics classroom [6]. This shift is done in effort to emulate the modeling and problem solving techniques used by professional scientists and engineers. Using a problem-driven approach to utilize computational methods encourages scientific discovery and increases the student's interest in computing [2], which could possibly influence a student's decision about their future career.

In the fall of 2013, the concept of programming was introduced to two honors and one college prep physics classes at Haverhill High School in Haverhill, MA. The decision to introduce programming to the students was a result of the partnership between the University of Massachusetts Lowell and Haverhill High School through the National Science Foundation (NSF) GK-12 Fellowship program. The role of the graduate fellow assigned to the aforementioned classes was to provide theoretical and educational resources to augment the high school curriculum. Each week, the graduate fellow and the physics teacher collaborated to determine what resources would be available to integrate into the physics curriculum. Often it was difficult to match the resources to the curriculum.

It was concluded that programming is an important resource and a useful skill the 21st century that could be applied

to most any topic in physics. Furthermore, it could be considered as an essential skill in all levels of high school physics- College Prep, Honors, and Advanced Placement. The programming lessons at Haverhill High School were originally used as the lesson modules for the NSF GK-12 Fellow to teach to the classes. However, it was quickly discovered through classroom observation that the programming lessons held a significant impact on the students learning and their interpretation of science.

This paper will give an explanation of how programming with the R language became a part of the high school physics curriculum. In Section II, the goals and objectives of the programming lessons will be listed. Section III will describe why the R language was chosen, how the new programming lessons were added to the curriculum, and an example assignment will be provided. Section IV will explain the successes and challenges faced while performing the programming lessons based on classroom observation and evidence. Section V will discuss plans to move forward based on the experiences so far.

## II. GOALS AND OBJECTIVES

At Haverhill High School, there is an interest in integrating computer programming activities to help meet the following goals and objectives. The ability to learn basic programming skills and applying them to solve problems logically and think critically is the ultimate goal of the initiative.

### *A. Goals*

The goals of incorporating programming into the physics curriculum at Haverhill High School are associated with the core learning skills that the students should have after completing one-year of weekly computer programming lessons. The goals are as follows:

(i) improve students' critical thinking skills through the development of logical and efficient coding

(ii) improve communication skills by frequently using descriptive comments within the code that articulate student understanding and logic

(iii) increase students' understanding of physical concepts by building programs that allow them to vary inputs in the simulations

### *B. Objectives*

Students should be able to achieve the following objectives which are related to specific programming concepts. Some of these concepts are applicable to all programming languages, where as others are more specific to R. However, in general, these objectives address basic programming concepts that should provide the students with the appropriate set of tools to be able to write any program in the future. At the end of the school year, students should be able to:

(i) assign a value to a variable or a constant

(ii) perform basic arithmetic operations using variables or constants

(iii) assign a series of values to a variable using sequences or arrays

(iv) apply the basic commands to control inputs and outputs

(v) apply the basic commands to create plots, including those with multiple sets of data on the same axes

(vi) utilize the appropriate arguments in the plot, and appropriate print commands to effectively communicate results

(vii) perform conditional operations using *if-else* statements

(viii) perform multiple iterations using *for* loops

## III. CURRICULUM IMPLEMENTATION

In previous years at Haverhill High School, technology was integrated into the classroom. However, it was limited to using WolframAlpha or Excel to perform repetitive calculations and create graphs of experimental data. The physics students use Excel regularly to create graphs of experimental data, including performing calculations to add other columns of of calculated data. Because of Excel has a large market share, the students will continue to have many opportunities to use Excel in school and in their careers. There was always a desire to introduce the students to technology that was more robust and more appropriate to modern scientific endeavors. Using a programming languages such as Matlab, Python, or R, satisfies this desire and exposes students to other computational tools. Thus, the reason for integrating computer programming into the physics curriculum.

The R language was chosen over other programming languages for various reasons. The primary reason for choosing R was that it is open source, meaning the license and software is free to download. The advantage of being open source made it easy to install on all of Haverhill High School's MacBook Air computers without the requirement of separate site licenses for each computer. Additionally, it would be accessible to the students if they chose to practice writing code at home. R also offers an intuitive and easy to use interface called RStudio. This interface eliminated the use of a text editor and separate compiler of the source code. A space for displaying plots is included in the RStudio interface. The ability to use the plotting space simplifies the process of analyzing of the results. The graduate fellow had extensive experience in R, providing the students with an technical resource for extra programming help.

### *A. Module Design*

There were two key components that needed to be addressed when designing the lessons on programming: (i) basics of programming and (ii) relating the programs to the physics curriculum. Starting with the basics of programming as a foundation, it was important that the complexity of the assignments was increased with each lesson. With each new program, the students would be required to use all of their

previous programming knowledge in conjunction with the new set of skills that were introduced. These challenges did not pose a problem with relation to the changing physics curriculum because it is already structured with increasing difficulty between assignments. Even though the physics lessons were continually moving to various topics, the students' knowledge of R grew with the weekly lessons, and with that, the complexity of the programs they wrote also grew.

One major challenge that was faced when designing the lesson plans was the difference between level of ability and understanding of physics and programming concepts between the honors level classes and the college prep level classes. Honors students were given the opportunity to think critically for themselves and collaboratively to complete the programming assignments. Although the goals and objectives remain the same, some of the assignments were differentiated to meet the needs of the college prep population. One of the differentiations included giving the college prep students a detailed model which contained examples of sample commands and R language syntax that may need to be used to complete an assignment correctly. Another change was providing them with more detailed instructions and hints to help with some of the more challenging parts of the assignment, such as derivation of necessary physics equations or a more elaborate explanation of the complicated steps of the algorithm. Extra lessons on the basics of programming were added for more practice. In addition, students with a background in programming or those who easily grasped the programming concepts were given the task of helping other students in the class. These modifications proved to be successful in clarifying mistakes and misunderstandings for the students who needed additional instruction. Multiple formative and summative assessments were given to the students to determine their level of understanding with the programming and physics concepts. An assessment rubric was made for each assignment. Details of the formulation of the rubrics are explained and an example is presented in Part D of this section.

### *B. Relation to physics curriculum*

Most of the programming assignments created are directly related to the physics concepts described in the Massachusetts Science and Technology/Engineering Curriculum Framework. These concepts include: kinetic energy, motion of a pendulum, Hooke's Law, conservation of energy and the coefficient of friction.

Additional programs such as, solving a quadratic equation using the quadratic formula and plotting three different polynomial functions on the same axes were meant for the sole purpose of introducing and enforcing basic programming concepts that would be helpful for the other assignments. Even though quadratic equations and polynomials are not strictly physics topics, they are mathematical concepts that are use extensively to throughout the physics curriculum.

### *C. Assignment Example*

As an example, a programing assignment that was used in the classroom is described. The students were required to calculate the velocity of a roller coaster over its second hill for two cases: (i) a frictionless roller coaster and (ii) a roller coaster that has friction. The programming objectives (i), (ii), (iv), and (vii) listed in Section II, Part B were the focus of this particular program. The students were asked to analyze a roller coaster that has two hills of differing heights, given the total energy equation for the frictionless case [8]

$$E = \frac{1}{2}mv^2 + mgh \tag{1}$$

where $E$ is the total energy, $m$ is the mass of the roller coaster cart, $h$ is the height of the roller coaster hills, and $v$ is the velocity of the cart. Given this equation as a starting point, the students are asked to use their skills acquired from algebra and physics to derive the correct equations for the velocity of the roller coaster cart at the bottom of the first hill and at the top of the second hill by hand.

Using the correctly derived equations, the next task was to determine a series of logical steps to determine the velocities based upon user input and implement those steps through writing the code. These steps included: prompting the user for the desired height of second hill; have the program calculate the velocities of the cart; and display the appropriate output. Goal (i) in Section II, Part A was satisfied here.

Next, the students were asked to repeat the process with the coaster under the effect of friction using a modified equation for the total energy. The code was to be modified to prompt the user to enter a coefficient of friction. If the coaster was frictionless, the prompt must inform the user the that coefficient of friction is to be entered as zero. Because the problem now required two cases, the students needed to use a conditional statement to calculate the correct solution. This portion of the assignment acted as an introduction to conditional statements. An example of an *if-else* statement was shown prior to the assignment. The students were allowed to follow the example to help them formulate the correct *if-else* statement for the given assignment.

By giving the user the ability to determine friction or no friction and choose the height of the hills, the program must be designed in a way that is universal enough to give a correct solution for all possibilities. Since different inputs and parameters can be used for this program, goal (iii) in Section II, Part A was satisfied.

Descriptive and helpful comments within the code were mandatory. They are a requirement for all programs because they give the students practice to effectively communicate the functionality of the program to others. It also satisfies goal (ii) listed in Section II, Part A.

*D. Assessment*

The goals and objectives described in Section II were assessed with a rubric that reflected those goals and objectives. The rubric varied slightly with each assignment depending on which goals or objectives were being stressed for a particular assignment. In general, the rubrics focus on the program having descriptive comments, how well the program met the specific concepts objectives, and how the output was presented.

An example of one of the rubrics is shown in Table 1. This rubric was used on an assignment where students studied the relationship between the length and the period of a pendulum. The pendulum length was varied and the period was calculated for all of the length values. A plot of period vs. length was generated to observe the relationship.

TABLE I. Example Rubric

| **Expectation** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|
| *Min. of 3 comments is used within the code.* | > 3 comments | 3 comments | 1 or 2 comments | No comments |
| *Utilize the seq() command to establish the proper range of values* | | | Yes | No |
| *Correctly calculate the period (input and output)* | | Eqn. used correctly calculated period | Eqn. has an error | Not calculated |
| *Graph must display a blue line* | Blue line is used | Line is present, but not blue | Graph exists, but does not show line | No graph |
| *Graph includes appropriate title-, and axis labels with units* | Satisfied | Title exists and have either labels or units but not both | No title and axes lacks labels or units | Title and axes labels or units are missing |
| **TOTAL POINTS** | | | | **12** |

A final programming project will be given at the end of the school year to determine if the students are proficient in all of the goals and objectives. The project will be directly related to a major physics concept and incorporate all of the programming concepts listed in Section II. Details of the project are still being determined.

## IV. CONCLUSION

The project is still within the first year of introducing programming to the physics classroom at Haverhill High School. Each week, new strategies and techniques are developed based on feedback from the previous week. Successes and failures of the programming initiative are discussed in Part A of this section. Plans for improvement are presented in Section V.

The intentions of this work was to explore the possibility of the integrating programming into a physics class and provide the graduate fellow with multiple lesson opportunities to fulfill the requirements of the NSF GK-12 Fellowship. However, through the implementation of these lessons, it was found that teaching programming in a physics class could be treated as a more formal program in the future. The programming lessons provided an opportunity to identify any challenges and develop the best practices to take if a formal program is launched. Plans to continue with programming in the physics class in future years are currently being discussed.

Without the support of the graduate student, it is possible that the physics teacher would have gone forward with a similar pilot project as series of after-school workshops. The teacher would have trained one or more of the current students to serve as teaching assistants or received help from a community member. In the future, students who demonstrated strong programming skills in the current physics class could be offered the opportunity to serve as teaching assistants for the course.

It has been easy to find novel programs that incorporate the physics concepts that are currently being studied. The nature of the equations and concepts in the physics curriculum lends itself well to the partnership between computing and physics. The equations are often easy to implement into a program. Writing a program based on a physics concept allows for increased student understanding on that concept by varying the input parameters and observing the change in the results.

Through classroom observation, it was been found that the students are often very engaged and many look forward to the new programs assigned to them each week. The students having difficulty with some of the concepts do not hesitate to ask questions and spend time after school for extra help. It appears that programming with the R language provided the students with a tool that reinforces physical concepts.

*A. Ability to meet goals*

For many of the students, programming was a foreign concept to them, but many of them picked up the tasks with relative ease. For the students needing extra help, they were allotted extra time after school where the graduate fellow and the physics teacher provide individualized support. Students who did not satisfy the objectives stated on the rubric were given the opportunity to correct their errors. This has proven to be very successful. Most of the students' work improved to receiving full credit on their assignments when they took advantage of this extra help time.

Many of the strategies already in practice have proven to be successful and helpful to the students. Perhaps the most successful tactic is providing the students with a quick reference guide that contains a list of the commonly used commands in R and examples of how they may be implemented in the program. The students depend on this

guide heavily to get ideas about the algorithm and to check the syntax of the commands. In addition to the reference guide, a few short examples are given to demonstrate the functionality and syntax of new commands. The students often refer back to these examples when trying to incorporate the new commands into program. The new commands are added to the reference guide as they are presented. By introducing new commands and requiring more advanced programs every few weeks, the students are constantly being challenged to think critically.

Another strategy that is successful involved the treatment of the more complex programs- those that require multiple logical steps. The students are provided with clear expectations about the program's objectives and the grading rubric a day or two prior to the allotted classroom programming time. Doing so gives the students time to strategize about the program, which includes developing logic and algorithms. During this time students also work on any of the physics content needed for the program, such as deriving the appropriate equations, translating the equations to the R syntax. More instruction is sometimes given to enforce the physical concepts addressed in the program during this time as well.

All computing and programming assignments were completed in class or after-school during designated programming time and therefore had little impact on the completion of other homework assignments. Giving the assignment to the students before the scheduled programming day gave them extra time to strategize their approach and helped minimize some of the time spent to complete the assignment. Although deadlines were set for the assignments, the students were allowed a small amount of extra time complete them because the students only had access to the programs through Haverhill High School's computers and could not easily work on them at home.

One of the major problems faced has been with the nature of the class schedule at Haverhill High School and the graduate fellow's time in the classroom. Typically, programming takes place in one fifty-minute period per week. The students are not usually given a sufficient amount of time to complete the program and subsequently, too much time passes before they are given the change to continue working on the program. In a recent effort to combat this problem, the time to write the programs is sometimes scheduled on consecutive days so the students can have uninterrupted time to complete the program. However, doing this too often interrupts the time spent on traditional physics curriculum.

It is sometimes a challenge to find the correct balance between programming time and physics time. In past years with the absence of programming lessons, the topics in the physics curriculum are not usually covered in their entirety. Therefore, it has been determined that it is preferable to cover fewer topics in greater detail and depth with the added benefit of computational and programming aspects. The hope is that the added benefits will have a greater impact on the students overall learning experience. What is lost in instructional time in content is hopefully mitigated by the gains in critical in computational thinking and allows for further inquiry of the physics content through simulations and graphical analysis. There is a currently a proposal to implement more formal formative and summative assessments and cognitive measurement practices starting in the 2014-2015 school year to determine the impact of programming on the student's learning.

## V. FUTURE WORK

Since teaching programming in the physics classroom is still in its infancy, the possibilities for growth and expansion are unlimited. There are plans to continue teaching programming in the physics classes at Haverhill High School in the future.

A week-long introduction to R tutorial will take place in the beginning of the school year, so the students become familiar with the syntax of the R language. Once the students become familiar with basic syntax of the R language, they will be taught how to develop flow charts to plan the structure of the program's algorithm. Using flow charts may clear up any confusion with logic, especially when the programs get more complex.

Another area for improvement is spending more time on basic programming concepts and examples before adding complexity when the students are not quite ready. Two possible ways to enforce basic programming concepts are: (I) providing short snippets of code and asking the students to analyze what each line of code is doing and (ii) giving the students a skeleton code and ask them to fill in the missing parts to make it run without errors. The latter will also give the students practice in debugging code on their own.

To determine if the students are gaining a better understanding of physics concepts through computing, follow-up questions regarding results presented in the program will be asked. These questions will give the students the opportunity to investigate and better understand the physics principles.

At this time, there is no quantitative data analysis done to verify the success of this programming initiative other than assignment grades. After the final programming project is complete, the plan is to perform some data analysis to determine the success and benefits of the programming lessons. A qualitative survey will be given to the students at the end of the school year to target student understanding and perspective on programming within the physics curriculum. This survey will hopefully serve as a means to determine if student learning was impacted.

Currently there is a proposal to offer a scientific computing course at Haverhill High School for 2014-2015 academic year. Many of the assignments used in the physics classes this year will be used for the scientific computing course.

## ACKNOWLEDGMENT

The authors wish to acknowledge the support of National Science Foundation grants (CCF0649235) and (DGE0841394), and Haverhill Public Schools.

## REFRENCES


[1] D. Langdon, G. McKittrick, D. Beede, B. Khan, and M. Doms, "STEM: Good jobs now and for the future," U.S Department of Commerce, Economics and Statistics Administration, ESA Issue Brief #03-11, July 2011

[2] S. Hambrusch, C. Hoffmann, J.T. Korb, M. Haugan, A.L. Hosking, "A multidisciplinary approach toward computational thinking for science majors," ACM SIGCSE Bulletin - SIGCSE '09, Vol. 41 Issue 1, pp. 183-187, March 2009

[3] R. L. Spencer, "Teaching computational physics as a laboratory sequence,"Am. J. Phys. Vol. 73, No. 2, pp. 151-153, February 2005

[4] K. C. Wise, J.D. Ellis, ed." The effects of using computing technologies in science instruction: a synthesis of classroom-based research," 1988 AETS Yearbook Information Technology and Science Education, Education Resources Information Center, Office of Educational Research and Improvement, U.S. Department of Education, pp. 115-128, 1989

[5] J. M. Wilson, E. F. Redish, "Using computers in teaching physics," Physics Today, pp. 34-41, 1989

[6] H. Gould, "Computational physics in the undergraduate curriculum," Computer Physics Communications, 127, pp. 6-10, 2000

[7] F. Esquembre, "Computers in physics education," Computer Physics Communications, 147, pp. 13-18, 2002

[8] P. G. Hewitt, Conceptual Physics: The High School Physics Program, pp. 149-150, Pearson Education Inc., 2009