Teachers' intended learning outcomes around computation in high school physics

Daniel P. Weller
Department of Physics & Astronomy, Michigan State University, 567 Wilson Rd., East Lansing, MI, 48824

Marcos D. Caballero
Department of Physics & Astronomy, Michigan State University, 567 Wilson Rd., East Lansing, MI, 48824
CREATE for STEM Institute, Michigan State University, 567 Wilson Rd., East Lansing, MI, 48824 and
Department of Physics & Center for Computing in Science Education, University of Oslo, N-0316 Oslo, Norway

Paul W. Irving
Department of Physics & Astronomy, Michigan State University, 567 Wilson Rd., East Lansing, MI, 48824

As an effort to integrate computational thinking into high school science classrooms across Michigan, we administered a year-long professional development series for physics teachers to learn how to program and utilize computer simulations as learning activities in their curricula. Near the end of their first year in the series, *N*=7 instructors were interviewed to explore their intended learning outcomes, design process, and implementation of computational activities. We conducted a thematic analysis on qualitative interview data to identify major themes from the teacher perspective. Herein, learning outcomes are articulated to investigate the professional development series' effectiveness at helping teachers prepare activities aligned with science standards and their specific aims. Our analysis provides insight regarding the lack of specificity in learning outcomes, the absence of assessment on computational content, and difficulty navigating the space of crosscutting concepts around computation. This research informs the augmentation of computational physics professional development workshops, communicates recommendations for practitioners, and documents teachers' intended learning outcomes when integrating computation in K-12 education spaces.

2019 PERC Proceedings edited by Cao, Wolf, and Bennett; Peer-reviewed, doi.org/10.1119/perc.2019.pr.Weller

I. INTRODUCTION: COMPUTATION IN K-12 SCIENCE

The growing desire to integrate computation into K-12 science classrooms is reflective of its unabating prevalence in modern science professions. There have been numerous calls to prioritize teaching computational thinking to all students. For example, the Next Generation Science Standards (NGSS) emphasize "using mathematics and computational thinking" as one of the eight main scientific practices that all students should experience [1]. Furthermore, in their *Guidelines for High School Physics Programs*, the American Association of Physics Teachers outlines that "a reasonable attempt should be made to keep up with rapidly changing technology" by including computer-based instruction (i.e., "computer simulation/modeling") [2]. Despite these pleas to incorporate computation in education, there is a lack of scholarly literature on how to best implement these ideas in the classroom [3, 4].

NGSS guidelines are somewhat vague when describing learning outcomes around computational thinking, and teaching computation is even more complicated when focused in a physical science classroom. For example, the standards state that students in grades 9-12 should be able to "create and/or revise a computational model or simulation of a phenomenon, designed device, process, or system" [1]. These nebulous learning outcomes, combined with the dearth of evidence-based pedagogical strategies for teaching computation in anything other than computer-based courses, presents a challenge for teachers wishing to align with NGSS and other high school physics program guidelines.

In an effort to accommodate the aforementioned calls, our team has developed and hosted a professional development series: Integrating Computation in Science Across Michigan (ICSAM). In this series, high school science teachers learned how to program (with Glowscript, a free browser-based VPython language) and teach computational modeling to their students. First, teachers participated in a five-day summer workshop where they prepared to implement coding in their classes with equitable learning in mind. Participants left the summer workshop with a repository of computational activities that they developed themselves. Throughout the year, the teachers returned every two months for workdays where they addressed successes, challenges, and patterns of inequity experienced while teaching computation. Now that the first year of ICSAM is complete, we hope to make improvements based on our research and teachers' reflections. Thus, this work will communicate recommendations for the design of future professional development series.

The ICSAM program was modeled after the Partnership for Integration of Computation into Undergraduate Physics (PICUP) faculty development workshop, except modifications were made to better fit the priorities of our program (e.g., equity, programming, pedagogy) [5]. Whereas PICUP dedicated an entire day to thinking about learning objectives around computation, such a session was omitted from ICSAM to focus on other relevant topics like equitable learning. Throughout our interactions with the teachers in follow-up workdays, we hoped to better understand their computational learning outcomes for our particular computational integration approach.

The research detailed herein investigates high school physics teachers' intended learning outcomes when integrating computation into their classrooms. We utilize teacher interviews and thematic analysis to identify major themes from the teacher perspective. In doing this work, we aim to (1) assist practitioners when designing activities aligned with their specific learning outcomes and standards, (2) provide recommendations for developers of computational physics workshops, and (3) elucidate teachers' learning outcomes involving computation.

II. INTERVIEW AND ANALYSIS METHODOLOGY

We utilized a semi-structured interviewing approach to study teachers' intended learning outcomes around computation in their high school physics classes [6]. The interview protocol was initially formed by selecting relevant questions from peer-reviewed sources [7–9]. Then, our research group (10+ members) workshopped, discussed, and provided feedback on the protocol to enhance the precision, wording, and organization of questions. The finalized interview protocol focused on teachers' computational backgrounds, course-wide learning objectives, activity-specific learning goals, activity design processes, and implementation in the classroom. Teachers were asked beforehand to have one activity in mind to discuss in our open-ended conversations, which typically lasted about 60 minutes in total.

Interviews were conducted with N=7 participants near the end of their first year implementing computation in the classroom. See Table I for the classes and activities discussed in interviews. We would like to note that Abe's and Carl's interviews both were discussed over two separate sessions owing to scheduling constraints. The teachers had varied experience teaching high school physics; Abe taught for 7 years, Carl and Emmett for 14 years, and the rest of the teachers had more than 20 years of experience. Generally, their self-identified confidence level with respect to programming was divided into three bins: low (Carl and Diane), medium (Abe and Burt), and high (Emmett, Frank, George). Interviews were audio-recorded and transcribed for subsequent analysis.

A thematic analysis was conducted to identify major themes in the interviews [10, 11]. Our analysis procedure, similar to that of Bartos and Lederman, explicates teachers' intended learning outcomes and relates them to shared curricular features that acted as affordances or constraints for student learning [12]. First, interviews were read to generate an initial set of codes around interesting discussion points. A set of preliminary themes was crafted from these ideas to encapsulate the space of teacher perspectives. Lastly, the set was reviewed by three other researchers to clarify and resolve the finalized themes.

TABLE I. Details from interviews with high school science teachers.

Teacher	Class	Computational Activity
Abe	Physical Science 2 Physics	Placing objects to spell name 1D elastic collisions Levers & mech. advantage
Burt	Physics	1D motion & kinematics
Carl	Physics	Momentum & 1D collisions Universal gravitation
Diane	Physics	Captain America in free fall
Emmett	Physics	Universal gravitation
Frank	AP Physics C [a]	Hooke's law & energy transfer
George	AP Physics C	Vertical spring-mass Bungee jumper

[a] AP Physics C is a calculus-based mechanics course.

III. LEARNING OUTCOMES AROUND COMPUTATION

The following subsections communicate teachers' intended learning outcomes discussed in interviews. Since the interview focused on both course-wide learning objectives and activity-specific learning goals the results for each have been presented separately.

A. Course-level learning objectives

Typically, the starter question to invite the teachers to explore big-picture outcomes was, “What are your overall learning objectives for students involving computation in your course?” Teachers responses fell into four related but distinguishable themes: developing computational thinking practices, utilizing computation as a tool for doing physics, learning physics through computation, and impacting student affect around computation. The following sections describe these objectives from the teacher perspective.

1. Developing computational thinking practices

Some teachers' defined their course-level objectives strictly with respect to computational thinking practices like programming, debugging, or creating simulations.

Burt: “I'm hoping that my students develop some level of proficiency in writing VPython code in the classroom...I'd like to see the students taking ownership of writing their own lines of code.”

Emmett: “I want [students] to be comfortable writing code so that they are familiar with it as a problem solving skill...So that they could say, ‘Maybe we can put this into a simulation really quickly and see what we get.”’

In the above two responses, we notice that teachers are specifically focusing on familiarizing their students with computer programming skills. On the other hand, Carl (shown below) focused heavily on troubleshooting and debugging throughout his physics course.

Carl: “I just want them to get deeper into the code to identify what different lines do and then be able to spot an error...I relied on them to troubleshoot the programs if they had issues when they executed them.”

Lastly, although a few teachers touched on the idea of using computational models to find and test solutions, the following quote by George describes this idea quite clearly.

George: “Seeing these kids be able to say, ‘Let's just change it and see what happens, and then we'll see what happens after,’ is exactly what I want them to do...That iterative process of thinking will really help them not only in this class, but also as they move on.”

Thus, objectives regarding computational practices mostly involved writing code, troubleshooting, and modeling.

2. Utilizing computation as a tool for doing physics

Several teachers underlined their learning objectives based around perceptions of the interaction between computation and physics. They wanted students to understand that computation is a tool for doing physics, as opposed to something that is there simply to gain experience working with code.

Abe: “I think the big picture they'll get from the computation is that it is just a physics equation. We see that in my class' computation assignment they're working on now. There's literally part of their assignment that says, ‘line 11, write that piece of code as a physics equation,’ which I think is important for them to see.”

From a big picture perspective, Abe describes that he just wants his students to observe that the equations being used to model the physics are effectively the same equations when translated into code. In the next extract, Diane elaborates further on the relationship between physics and computation.

Diane: “As far as with the physics, just to see that there is another way we can do it. We do it in graphs, we do it in motion, we do it in video, we do it on paper, and now we've got the computation as well.”’

The above quotes demonstrate Diane and Abe outlining all of the tools that students have used previously to do physics and that computation is now just another one of those tools.

3. Learning physics through computation

The themes of “utilizing computation as a tool for doing physics” and “learning physics through computation” are related but distinguishable. Viewing computation as a tool for doing physics is different than actually using that tool develop a better understanding of physics in students. The following quotes demonstrate the various ways teachers viewed computation strictly for the purpose of learning physics.

George: “The goals of computation are for them to be able to write or re-work the code, using their physics knowledge, to understand the physics more deeply.”

Emmett: “Because it requires you to think about what’s going on in this step-by-step manner, I was hoping that exposure would solidify their physics understanding.”

This may be a relatively unsurprising theme given that all of the interviewees are physics teachers, but surprisingly, only these two teachers articulated course-level objectives related to learning physics via computation. Most teachers did not discuss coding as a way of learning physics until discussing tasks of a smaller grain size (i.e., specific in-class activities).

4. *Impacting student affect around computation*

The theme that emerged the strongest from interview data pertains to impacting students’ attitudes and feelings toward computation. The following two teachers’ objectives for physics and AP Physics, respectively, skirt between two interconnected ideas: it will be advantageous for students to have computational experience for their future endeavors, and to enable this, teachers should strive to reduce the intimidation factor associated with coding.

Carl: “Our goal is that when the students go on, they’ve got a little understanding of coding so they’re not afraid of it and they’ve got something to build on. That was really our major objective...I think if they got an entry and they just know the logic of it, it opens doors that otherwise would remain closed. They’d be afraid of it. They’d feel intimidated.”

In the above quote, Carl projects negative associations and feelings when it comes to a future where his students will be working with programming. The potential for fear around computation is indicated twice and a concern about intimidation is brought up. Intimidation and the providing a positive experience with computation is repeated below.

Frank: “It isn’t really a tie into the college board course objectives. I guess it’s a personal course objective of mine that I want them exposed to some of these skillsets...Then if they’re going out to take advanced classes that have more of this, at least they bring that exposure to it. It’s not new, it’s not intimidating, they’ve seen this before, so I think it could make the whole science experience better for them.”

Frank declares that this learning outcome around reducing the intimidation factor is a personal objective as opposed to one guided by education standards. Again, there is a projection toward some future experience for the student, and Frank attempts to reduce the bar to successfully engage in these future computation-based activities. This projection forward is not always so strongly associated with negative feelings. Instead, Diane simply focused on the feeling of accomplishment and providing students with a positive experience.

Diane: “I think a bunch of different things. One, a sense of accomplishment, like, ‘Yes I got it to do it.’...And also just see what a program is and know what people are talking about when they talk about a language."

In general, interviewees who focused heavily on the positive experience also tended to discuss their learning outcomes for computation and physics separately. This highlights two different teaching perspectives of integrated computation versus computation and physics as distinct disciplines. Overall, the diversity of learning objectives discussed by different teachers demonstrates the complex, multi-faceted nature of implementing computation in the classroom. While most teachers had one large overarching objective, some teachers covered several themes. Ultimately, very few of the teachers mentioned education standards as justification for integrating computing in their courses.

B. Activity-specific learning goals

Teachers were asked before interviews to consider one of their computational activities that would be suitable for grounding our discussion of learning outcomes. After describing the activity, each teacher was asked, “What are your learning goals for students in this activity?” Whereas teachers’ course-level objectives were relatively broad, their learning goals for one chosen activity tended to be more refined and aligned with NGSS. For example, in George’s vertical spring-mass activity, he clearly stated his learning goals.

George: “The main goal of this is for [students] to see that the total energy is conserved...An aspect of using the physics principles that they haven’t had before is to insert them or change them within actual computer code; that is something that’s completely new. And then there are goals about the computation that are completely different than the physics goals, like iterative thinking or understanding computer code as a concept. So, this is a many-layered thing.”

In this case, George highlights a disciplinary core idea (conservation of energy), a computational thinking practice (iterative thinking), and a possible crosscutting concept (applying physics principles in code). The first two themes are exactly consistent with the language used to describe performance expectations in the NGSS energy unit, whereas the latter does not fit well within any explicitly defined crosscutting concepts [1]. In order to view something like inserting physics principles into code and then making changes (as described in the previous quote) as a crosscutting concept, then one would have to see computation as its own separate knowledge domain. This ambiguity around crosscutting concepts further underscores the aforementioned duality of integrating computation into physics versus physics and computation as two separate entities.

Another emergent pattern was that teachers' goals seemed to focus on different themes based on how early computational learning activities took place in the curriculum.

Abe: "What I'm looking for is them to get the idea that copying and pasting code is a nice, easy, convenient way to get an object. Then also changing the position of the object and working with the axis attribute, where they can tilt things...Some get excited anytime that they can create something and have it be theirs. And then as a teacher, you kind of just build off that with high fives." *(Placing objects to spell name activity)*

Abe clearly communicates his attention to students' interest and self-efficacy around computation. He gives them the chance to claim ownership over their creation, and then, he follows that up with high fives to reinforce that they have achieved something positive. This focus on self-efficacy was true of learning activities positioned toward the beginning of the curriculum for other teachers as well. Early on, learning goals related to basic computational practices like programming, but teachers also made explicit references to students having a positive experience when first introducing the new material. As curricula progressed, teachers' goals morphed to focus on computational practices and physics concepts.

IV. DISCUSSION AND IMPLICATIONS

Our data from teacher interviews demonstrate an evolution of learning outcomes based on where computational learning activities fall in the curriculum. In the beginning of a course, a heavy focus is placed on computational practices and providing a positive experience. These outcomes disappear for deeper conceptual physics and computational thinking outcomes as the curriculum progresses. An on-boarding process is reasonable to expect, especially for those teachers who are focused on providing a positive overall computational experience for their students. However, this also could contribute to a divide between computation and physics content, rather than relating the two subjects. Additionally, most teachers referred to their course-wide objectives with vague wording, which could indicate a potential shortcoming of the professional development series.

For the theme of developing computational thinking practices, NGSS guidelines are relatively unclear. Accordingly, it is difficult to attribute the teachers' motives to specific education standards. In George's case, his course-level objective could be mapped to the NGSS outcome of "using a model to predict relationships in systems" [1]. However, the practices discussed by other teachers, like programming (Burt) and debugging (Carl), perhaps more closely fit the taxonomies of other computational thinking frameworks [13, 14]. Unfortunately, any computational thinking framework is highly context-dependent, and in turn, the practice-based learning outcomes vary significantly from one programming environment to the next. In general, we broadly classified writing code and similar ideas as the practice of computational thinking, but more work needs to be done to particularize the language around this topic for our specific context. Hence, teachers' descriptions related to computational thinking practices were underdeveloped from this perspective.

Regarding recommendations for future computational physics workshops, our analysis revealed several areas for improvement. First off, we find it important to have an explicit session about learning outcomes to get participants to think about these ideas from the onset of their professional development. This will support teachers in creating learning activities in accordance with their specific aims, as well as education standards. Moreover, participants should begin by brainstorming about how their previous experiences and knowledge may be viewed as computational thinking to impress on them that this is not an isolated knowledge domain. One approach could start by grounding the conversation around a canonical physics problem (e.g., projectile motion) and then conceiving how computational thinking practices apply to that problem. Then, a conversation about the affordances and constraints of computing technology is a natural next step. Finally, developers should explicitly discuss computational learning outcomes that previous teachers have considered and outline potential progressions to reach those outcomes. These suggestions hypothesize possible routes to facilitate future professional development events.

The work presented in this report serves as a foundation for a wealth of future research. For instance, the vast prevalence of learning outcomes aimed at impacting student affect motivates the development of a computational attitudes survey as a method of measuring student gains. Currently, there is a marked lack of strategies for assessing students' and teachers' attitudes toward computational material. Such work could have important implications on buy-in toward these learning approaches. Additionally, computational learning environments offer a unique context for studying equitable learning and self-efficacy with respect to demographic dimensions. For example, do different learners engage with computational content differently along lines of gender, race, or other social markers? Equity research is needed to engage historically under-represented groups in these important learning opportunities. Another promising area of research surrounds how teachers' intended learning outcomes evolve over the course of their professional development. Our future research will attempt to address these research areas.

ACKNOWLEDGMENTS

We would like to thank the National Science Foundation (DRL-1741575) for funding and the teachers for graciously donating their time and effort to our research. We also thank Paul Hamerski and Theodore Bott for insightful discussions.

[1] National Research Council, "A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas," Washington D.C.: The National Academies Press (2012).

[2] American Association of Physics Teachers, "Guidelines for High School Physics Programs," (2002).

[3] S. Grover and R. Pea, "Computational Thinking in K-12: A Review of the State of the Field," Educ. Res. **42**(1), 327 (2013).

[4] L.D. English, "Advancing Elementary and Middle School STEM Education," Int. J. Sci. Math. Educ. **15**(1), 1 (2017).

[5] N. Chonacky and D. Winch, "Integrating computation into the undergraduate curriculum: A vision and guidelines for future developments," Am. J. Phys. **76**(4), 327 (2008).

[6] V.K. Otero and D.B. Harlow, "Getting Started in Qualitative Physics Education Research," Rev. PER **2**(31) (2009).

[7] A.J. Magana, S.P. Brophy, and G.M. Bodner, "Instructors' Intended Learning Outcomes for Using Computational Simulations as Learning Tools," J. Eng. Educ. **101**(2), 220 (2012).

[8] A.S. Pawlak, "Collaborative Learning Environments in Introductory Physics," Ph.D. Dissertation (2018).

[9] M.D. Caballero, N. Chonacky, L. Engelhardt, R.C. Hillborn, M.L. del Puerto, and K.R. Roos, "PICUP: A Community of Teachers Integrating Computation into Undergraduate Physics Courses," The Physics Teacher **57**(6), 397 (2019).

[10] V. Braun and V. Clarke, "Using thematic analysis in psychology," Qual. Res. Psych. **3**(2), 77 (2006).

[11] R.E. Boyatzis, "Transforming qualitative information: Thematic analysis and code development," Thousand Oaks, CA: Sage Publications, Inc. (1998).

[12] S.A. Bartos and N.G. Lederman, "Teachers' Knowledge Structures for Nature of Science and Scientific Inquiry: Conceptions and Classroom Practice," J. Res. Sci. Teach. **51**(9), 1150 (2014).

[13] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, and U. Wilensky, "Defining Computational Thinking for Mathematics and Science Classrooms," J. Sci. Educ. Tech. **25**(1), 127 (2016).

[14] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," AERA, 1 (2012).