education sciences

MDPI

Article

Expanding Models for Physics Teaching: A Framework for the Integration of Computational Modeling

Rebecca Elizabeth Vieyra [1,*], Colleen Megowan-Romanowicz [2], Kathi Fisler [3], Benjamin S. Lerner [4], Joe Gibbs Politz [5] and Shriram Krishnamurthi [3]

1 Department of Physics, University of Colorado Boulder, Boulder, CO 80309, USA
2 American Modeling Teachers Association, Sacramento, CA 95820, USA; colleen@modelinginstruction.org
3 Department of Computer Science, Brown University, Providence, RI 02912, USA; kathryn_fisler@brown.edu (K.F.); shriram@brown.edu (S.K.)
4 Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA; blerner@ccs.neu.edu
5 Jacobs School of Engineering, University of California, San Diego, CA 92093, USA; jpolitz@eng.ucsd.edu
* Correspondence: rebecca.vieyra@colorado.edu

Abstract: Teaching computation in science courses can enhance science education, but doing so requires that teachers expand the vision of their discipline beyond the traditional view of science presented in most curricula. This article describes a design-based research (DBR) program that included collaboration among high school teachers and professional development leaders in physics and computer science education. Through three years of professional development and teacher-led development, field testing, and refinement of integrated curricular resources, we have combined instructional modeling practices, physical lab materials, and computer programming activities. One of the outcomes is a co-created framework for the integration of computational modeling into physics that is sensitive to teachers' interests and expressed needs in addition to learning goals. This framework merges two evidence-based approaches to teaching: Bootstrap:Algebra, a web-based computing curriculum that emphasizes using multiple representations of functions and scaffolds that make the programming process explicit, and Modeling Instruction in physics, an approach that emphasizes the use of conceptual models, modeling practices and representational tools. In doing so, we uncover the need to balance teachers' visions for integration opportunities with practical instructional needs and emphasize that frameworks for integration need to reflect teachers' values and goals.

Keywords: computation; integration; modeling; physics; programming

Citation: Vieyra, R.E.; Megowan-Romanowicz, C.; Fisler, K.; Lerner, B.S.; Politz, J.G.; Krishnamurthi, S. Expanding Models for Physics Teaching: A Framework for the Integration of Computational Modeling. *Educ. Sci.* **2024**, *14*, 861. https://doi.org/10.3390/educsci14080861

Academic Editors: Michael C. Hynes and Craig Berg

Received: 26 May 2024
Revised: 23 July 2024
Accepted: 31 July 2024
Published: 8 August 2024

1. Introduction

Numerous studies have designed frameworks for the integration of computational thinking (CT) into core science courses. The literature explores the educational expectations of practicing scientists [1,2], uncovers the CT-related activities of science teachers [3], and discusses the compatibility of computer science and natural science principles [4]. A synthesis of the literature on frameworks for CT-integrated science [5] illustrates the diversity of each of these frameworks. While Weller et al. [5] proposed a researcher-synthesized framework, verified through their observations of high school CT-integrated physics classrooms, we foreground the voices of teachers as they design and implement CT-integrated physics for themselves.

In this article, we describe the evolution of a framework for CT-integrated introductory high school physics teaching, with emphasis on computational modeling (CM). CT broadly encompasses those concepts and skills that are foundational to problem solving in the computer science discipline [6], which, in the sciences, may include components such as data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices [3]. We define CM as a subdomain of CT that emphasizes the representation of relationships in code, programs, and simulations—wherein the code, programs, and simulations are the models that represent our understanding of those relationships (see Table 1). For this article, we focused specifically on CM that involved the student as a programmer.

Table 1. Core definitions.

Computational Thinking (CT)	"...solving problems, designing systems...by drawing on the concepts fundamental to computer science" [6].
Computational Modeling (CM)	The representation of relationships in code, programs, and simulations. CM is a subdomain of CT [3].
CM-integrated physics	The representation of physical relationships in code, programs, and simulations in the context of a core physics course.

Our study was a three-year design-based research investigation with 75 secondary physics teachers across the United States. We illustrate how computationally supported teachers contributed to our understanding not just of what is theoretically possible for CM-integrated physics, but what is realistically possible based on teachers' interests, constraints, and needs.

Our program co-developed, tested, and refined the curriculum resources and framework with secondary physics teachers. The genesis of the collaboration was the convergence of a calls from federal and professional societies to integrate computing into STEM [1,7], a response through expressed desire of physics teachers for computational resources to use in their secondary physics courses, the appearance of an accessible approach to teaching students to program, and the rise of Physics First [8] as a viable high school science sequence. This convergence produced a venue to test computational integration in a much lower stakes environment for students and teachers than advanced physics courses, where programming has historically been incorporated in preparation for college.

We responded to calls to "be explicit about the nature of [STEM] integration, the types of scaffolds and instructional designs used" [7] by focusing on the experiences of secondary physics educators as both learners and teachers. While researchers found that CM-integrated physics can enhance the understanding of fundamental principles, modeling complexity, and emphasizing physical relationships over physics topics [9], this benefit is not always sufficient to catalyze teacher uptake of integrated approaches. We aimed to address this challenge of CM-integrated physics with secondary physics teachers by merging two evidence-based teaching approaches: Bootstrap:Algebra [10–13], a curriculum that emphasizes using multiple representations of functions and scaffolds that make the programming process explicit, and Modeling Instruction (MI) [14–16] in physics, an approach that emphasizes the use of conceptual models, modeling practices and representational tools.

After two years of teacher-led development of integrated curricular resources and more than a year of dissemination (Table 2), we co-created with teachers a proposed framework for CM-integrated physics that illustrates what educators believe they can teach, and their students can learn. This paper lays out the development of this project, recounts the teachers' experiences as they enacted an integrated curriculum, and describes the resulting framework for integration.

Table 2. Study Timeline.

Academic Year	Purpose	Participants	Outcomes
2016–2017	Curriculum development	30 experienced MI teachers	• Initial Computational Modeling in Physics Framework • Identified curriculum developers
2017–2018	Curriculum revision and continued development	22 experienced MI teachers, in addition to 8 from prior year	• Identified three MI teachers with strongest curriculum-writing and MI workshop-leading abilities to bring coherence to the curricular products
2018–2019	Workshop deployment with revised curriculum	23 teachers: 50% experienced MI, 50% new to MI	• Developed revised Computational Modeling in Physics Framework • Finalized curricular materials

2. Prior Work on Computational Modeling Integration in Physics and Research Gaps

Our work is situated within calls for enhancing students' computational thinking (CT) skills through science education [17,18]. Within the Next Generation Science Standards, it is generally understood that CT in science "enable[s] the representation of variables, the symbolic representation of relationships and the predictions of outcomes" [19]. Computational skills are well-represented in analyses of skills for STEM-enabled professionals in the workforce [20,21]. Even so, research on CT in science disciplines in K-12, especially as it relates to what teachers think about how CT can be practically integrated into their practice, is limited.

Weintrop et al. [3] constructed a taxonomy of CT in math and science courses composed of four practices: data, modeling and simulation, computational problem-solving, and systems thinking. They also presented teacher-produced math and science lessons that included these CT skills. However, many of these examples explored concepts through front-end simulations such as PhET Interactive Simulations [22] or graphical analysis of data. Science teachers rarely incorporated the broader set of CT skills traditionally associated with Weintrop et al.'s [3] problem-solving practices, such as preparing problems for computational solutions, programming, creating computational abstractions, or troubleshooting and debugging.

One factor that makes integration of science and CT—especially that with a programming focus—challenging is that most students struggle to relate the more iterative models of computation with the more causal models of physics [23]. Teachers find disciplinary content integration across the STEM fields to be a challenge [24,25], particularly as it relates to the "...balancing and exploiting the trade-off between the domain-generality of CT...and the domain-specificity of science representations present a significant educational design challenge" [26].

Beyond those mentioned, several other projects have developed curricula and assessments for integrated CT and science generally [2,27] and physics specifically [5,23,28–32]. However, this literature often presents the integrated material as an end-product, with a limited description of its development process.

The integration of computation in physics has received substantial attention in higher education, with the potential for identified benefits of integration to be applicable at the secondary level. Sherin [33] was one of the first researchers to identify the utility of CM-integrated physics as a representational tool, and Langbeheim et al. [31] looked at the use of CM-integrated physics as a representational tool used in partnership with hands-on laboratory experiences. Sherin found in the context of his study, using his selected programming environment, that students conceptualized physics differently depending on the representational tools that they used. He concluded that "algebra-physics can be characterized as a physics of balance and equilibrium, and programming-physics as a physics of process and causation" [33]. Sengupta et al. [4] and Basu et al. [26] have developed an intervention that integrates computing into 6th-grade physics, focusing on CT as a representational tool and as a problem-solving process. A limited number of studies [31,34] have looked at how CM can be used as an integrated approach alongside hands-on experiences. However, what is noticeably absent from the literature base are case studies that illustrate the complexities and often indirect pathways to developing integrated material that considers the goals (and the voices) of teachers themselves, or an exploration into how these benefits at the higher education level could support learners in secondary schools.

3. Materials and Methods

Our work focused on the following research question: What do teachers believe should be the characteristics of an integrated computational modeling in a physics instructional framework? Our research question was directed toward understanding high school teachers' vision and goals for science education, including students who might not be bound for college or STEM careers.

We adopted Cobb et al.'s approach [35,36], a type of design-based research (DBR), founded on the idea of design experiments [37–39] that "are iterative, situated, and theory-based attempts simultaneously to understand and improve educational processes" [39]. DBR studies are appropriate when there is limited research on the desired practice.

On the spectrum of studies described by Cobb et al. [36], this effort approximates an organization design study, as it involved multiple stakeholders (physics teachers, the organization that supports their professional growth and development, physics education researchers, computer science education researchers, and curriculum developers). This approach relies on design and iterative re-design processes. This program was multi-faceted, including curriculum-creation, community-building, classroom implementation, and assessment. The focus of this article is on the iterative evolution of the program's core principles as they pertain to the integration of physics and CM that underlay all program elements—a framework for integration.

At the teachers' behest, we sought to produce a framework for improved physics teaching and learning, seamless integration of computational modeling into high school physics, and a proof of concept that this could be accomplished for a general introductory course for 9th grade students (ages 14–15). In collaboration experienced MI physics teachers, the authors engaged in multiple design and implementation cycles, ultimately producing a three-week teacher workshop, a well-tested, comprehensive set of curriculum resources for a one-semester course, and a framework to guide integration of computational modeling into any high school science course. This iterative process enabled us to assess and address the problems of practitioners as they arose, to construct a theory of action with respect to how students learned, and to devise a generalizable approach for integration of computing and science.

Teams of teachers convened to develop curriculum and design professional development in 2016 and 2017, with 12 teachers in the first year and a total of 30 in the second year. Development workshop participants included experienced MI physics teachers and several prior MI workshop leaders and curriculum developers. During these workshops, teachers learned to program in Pyret (see https://pyret.org/index.html, (accessed on 1 August 2024)) during the first week and developed a curriculum to use in their classrooms for the remaining two weeks. Deployment workshops took place in the summers of 2017 and 2018 for additional 45 teachers from across the U.S. both years. These workshops were led by two to four teachers, all experienced Modeling workshop leaders, who participated in the development workshops. They refined a subset of the most coherent and classroom-tested CM-integrated physics and engaged the new teacher participants in professional development to prepare them to use the materials with their students. Between summer workshops, teachers met monthly to share feedback on implementation and offer revisions to the curriculum.

The focus of this article is the framework and its development, not on its evaluation. However, it would not be particularly interesting if it were not effective at all. The external evaluation of this project in the years including and following this study (2017–2021) used quantitative data from teachers who participated in the professional development experience and used the materials across at least one academic year. These teachers demonstrated a statistically significant improvement in their attitude toward CM from pre-test to post-test (F (1,217) = 17.65, $p < 0.05$). Multivariate modeling also demonstrated small but statistically significant improvements in teachers' mathematical understanding of functions (Λ (3,209) = 0.06, $p < 0.05$), in addition to attitude (Λ (3,209) = 0.07, $p < 0.05$). Teacher interviews by a different group of researchers [40] has also revealed significant differences in conceptualization of physics and computational modeling integration, influencing teacher self-efficacy and implementation of the materials. While students' attitudes toward CM fluctuated significantly by teacher and by year and saw small decreases overall (F (1,4461) = 8.78, $p < 0.05$), students demonstrated gains in understanding of physics (Λ (3,4109) = 0.11, $p < 0.05$) and functions (Λ (3,4109) = 0.01, $p < 0.05$). These results show that there is benefit to further exploring this framework and the materials it has produced.

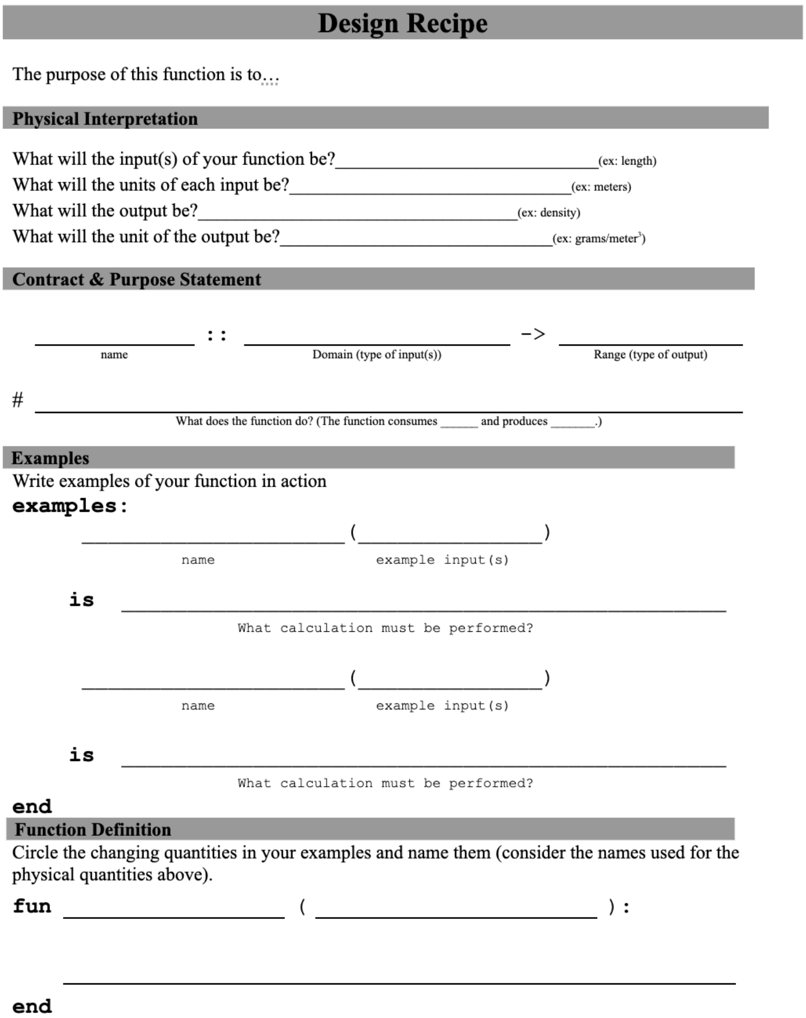
4. Program Integration: Modeling Instruction and Bootstrap:Algebra

4.1. Theory Base for Modeling Instruction and Bootstrap:Algebra

Given the criticism that theory is often underutilized in DBR studies [39], we present the grand theories behind the existing programs we merged, as well as the program design. Both MI and Bootstrap:Algebra rely on domain-specific instructional theories (for physics, mathematics, and computer science education) that are socio-constructivist in nature—they attend to collaborative learning, making their thinking visible through representations, testing ideas, and providing scaffolding for challenging ideas.

MI [41,42] is an approach to teaching physics that is anchored in the construction, revision, and application of models and multiple representations [15,16,43]. The approach's effectiveness has been documented with data from more than 30,000 students [42,44], and has been shown to improve student attitudes [42,45], conceptual understanding [46], and reformed instructional practice [47]. For more about MI workshops and a comprehensive discussion of the design and efficacy of MI, see Barlow et al. [48], Brewe [49], and Jackson et al. [42].

Bootstrap:Algebra [10–13] was developed to integrate computing into algebra courses through a shared foundation in functions. Bootstrap:Algebra guides students to solve word problems and write programs using a step-by-step approach called the Design Recipe (Figure 1). This approach has been used in Bootstrap's teacher PD workshops since 2007 and is embodied in the textbook *How to Design Programs* [50]. It illustrates abstraction, testing, and modeling with variables, all of which are core topics in CS standards, though the first two are generally overlooked in early computing curricula.


Design Recipe

The purpose of this function is to…

Physical Interpretation

What will the input(s) of your function be?__________________(ex: length)
What will the units of each input be?__________________(ex: meters)
What will the output be?__________________(ex: density)
What will the unit of the output be?__________________(ex: grams/meter³)

Contract & Purpose Statement

________ :: ____________ -> ________
name / Domain (type of input(s)) / Range (type of output)

\# ________________________
What does the function do? (The function consumes _____ and produces ______.)

Examples

Write examples of your function in action

```
examples:
    ______________ (__________)
         name       example input(s)
  is ________________________________
       What calculation must be performed?
    ______________ (__________)
         name       example input(s)
  is ________________________________
       What calculation must be performed?
end
```

Function Definition

Circle the changing quantities in your examples and name them (consider the names used for the physical quantities above).

```
fun ____________ ( ______________ ):
    ________________________________
end
```

Figure 1. Design Recipe worksheet to create functional representations.

4.2. Integrating Modeling Instruction in Physics with Bootstrap:Algebra

MI and Bootstrap:Algebra's strengths in CT complement one another. The collaboration between these two initiatives fits Weintrop et al.'s [3] more general computational thinking in science framework as a starting point for developing our computational modeling in physics framework. We recognized that MI utilized Weintrop et al.'s [3] data and systems thinking practices and modeling and simulation practices more generally. Complementarily, Bootstrap:Algebra emphasized Weintrop et al.'s [3] modeling and simulation and computational problem-solving practices. We envisioned how programming tools in Bootstrap:Algebra could support physics education by serving as another representation of models, adding programming as another item in MI's toolbox that already included physical models, graphical models, algebraic models, and visual models. We initially considered two important aspects that defined what was our shared vision for the integration of computational modeling into physics. We also explored how educators might want to integrate these fields, especially given our own experience as educators that high school teachers often attempt to bring programming into their teaching in a relevant way, but would like more support doing so.

First, our integration addressed Science and Engineering Practices #2 (Developing and using models) and #5 (Using mathematics and computational thinking) of the Next Generation Science Standards [18]. The *K-12 CS Framework* [51] states that "[p]rogramming can assist in the collection and representation of data in mathematics or science classrooms". While we wanted to leverage programming to better facilitate what physics already needed to teach, we also felt that its integration could enhance the accessibility of higher-level physics to introductory learners while also providing needed clarity on what elements of computational modeling students could learn as they develop physics understanding [52].

Second, recognizing the wide availability of computer simulations for science, we instead designed a program that focused on computational models and modeling. Although simulation (using pre-crafted models to represent the world) has received substantial attention [22], modeling requires a distinct investment of intellectual capacity because it entails the creation (and use) of models [53]. In a computing context, modeling entails engagement in identifying key parameters, defining functions over parameters, and articulating the processes embodied within the code.

4.3. Program Evolution

We recognized the need to balance training in CM with opportunities for MI master teachers to apply CM in their classroom practice. In the summers of 2016 and 2017, we recruited experienced MI physics teachers, each of whom had previously gone through a 3-week workshop (Mechanics I) and had implemented the approach in the classroom for at least one year. The development experience employed research-validated practices in curriculum development, attending to national standards, learning outcomes, and activity-assessment alignment [54]. Workshops were aligned with evidence-based practices in PD [55], including a focus on content knowledge, active learning, and a cohort-style 90 h workshop.

4.4. Evidence of Teachers' Interests, Constraints, and Needs in CM-Integrated Physics

To better understand how these experienced MI physics teachers' conceptualizations of CM-integrated physics evolved, two of this article's authors collected participant-observer field notes [56] and surveys throughout the development workshops and monthly virtual meetings throughout the following academic years. During the workshops, two of this article's authors and two research assistants conducted both interviews and think-aloud problem-solving interviews [57].

What teachers produced during the workshops, used in their classrooms, and shared for future teacher deployment workshops shed light on what they believed they could teach and what their students could learn. Table 3 describes the CM-integrated physics content of several core modules developed by teachers in our program, along with examples of the activities used to integrate the two areas. The full set of student resources is publicly accessible through our project website at https://www.compadre.org/precollege/CMP/ (accessed on 1 August 2024).

Table 3. Examples of integrated computational modeling activities in each curricular unit.

Unit	Physics Content	CM Content	Example Integrated Activity
1	Qualitative Energy	System state; Data types; Contracts; Using Functions	Students observe a simulation of a bouncing ball and note changes in state. Two parameters (starting height and energy loss) are modified to change the model for energy transformations.
2	Constant Velocity	Defining functions of one argument; Testing; Differential representations (functions representing state update)	Students create functions to represent the motions of two cars moving toward each other. Students use a prediction generated from the computer simulation to test the setup with real toy cars experimentally.
3	Uniform Acceleration	Multiple coordinated state updates; Defining functions of multiple arguments	Students attempt to model accelerated motion differentially for each interval of time accurately. Students learn that average velocity over a time interval is equal to the instantaneous velocity at the middle of the time interval.
4	Balanced Forces	Booleans; Conditionals; Testing conditional functions	After investigating the motion of an air-levitated soccer ball, students model an air hockey puck's motion as it moves from one frictionless half of the table where the air is working to the other half where it is not working friction is present. Students modify the function to get the puck to land within a target.
5	Unbalanced Forces	Synthesis of computational concepts and differential representations	Students model thrust and drag on a falling coffee filter that extends learning from a hands-on coffee filter lab on terminal velocity.

A defining feature of the CM-integrated physics materials is the emphasis on differential (rather than closed-form) models. This element arises from Bootstrap:Algebra's modeling approach, which the teachers then adapted to a MI physics context. In an example modeling cycle, students first identify the physical constants (e.g., gravity), static information (e.g., the height of a bridge), and dynamic information (e.g., the position of a moving car) that arise in the given scenario (these become constants and variables/parameters in the computational model). Students then work out examples of the differential changes from one moment in time to the next; if a scenario involves an external stimulus (e.g., striking a hockey puck), students show changes before and after the stimulus occurs. Based on the examples, students design and test one or more algebraic functions representing the differential changes to the dynamic variables.

The differential approach is a more general way of describing systems because it applies equally to systems for which there does not exist a known closed form. For instance, systems like accelerated motion are not only quite easy to express but are also insightful—the speed of a falling ball in the next instant is a function of its speed in the current instant—in the differential form (Table 4). This approach is potentially easier to comprehend than using traditional time-parametric equations that require students to understand polynomials, fractions, and square roots to solve problems. The differential approach allows students to tackle higher-level physics despite not yet having the math mastery that traditionally may be an obstacle. Having students become familiar with the differential form—which is intentionally evocative of differential equations—in this gentle setting prepares learners for future study of calculus and advanced physics.

Table 4. Representations for uniform acceleration through algebraic formulae and functions.

	Parametric (Time-Based)	Differential
Algebraic Formula	$x_f = x_i + v_i\ t + \frac{1}{2}\ a\ t^2$ $v_f = v_i + a\ t$ . . .combined, these equations yield a third formula typically used in physics. . . $v_f^2 = v_i^2 + 2\ a\Delta x$	$x_{n+1} = x_n + v_{n\ (average)}\Delta t$ $v_{n+1} = v_n + a\Delta t$

Computational Function (in the Pyret language)	`fun x-at-t(t):`
	`x-init + (v-init * t) +`
	`(0.5 * a * t * t)`
	`end`
	
	`fun v-at-t(t):`
	`v-init + (a * t)`
	`end`
		`fun next-x(x, v-ave):`
		`x + (v-ave * delta-t)`
		`end`
		
		`fun next-v(v):`
		`v + (a * delta-t)`
		`end`

5. Results: A Teacher-Driven Framework for Integration

5.1. An Initial Framework (2016–2017 Academic Year)

To consolidate how we saw teachers integrate CM into physics, we built a framework to define how physics teachers, with the support of physics education and computer science education experts, could integrate CM into their curriculum and instruction. Discussions and 30–45 min interviews with each of the teachers from the first cohort provided insights into the proposed framework. As a first foray into CM of physics, participants were asked how they would consider incorporating this into their teaching, especially where it would fit along the MI continuum of model development and deployment (use). After sketching out what we thought were the most prominent features of the integration, we then presented our draft framework to teachers at a monthly meeting and made modifications.

First, teachers distinguished between simulation and modeling, noting that they each had their benefits at different parts of the MI cycle. Some teachers preferred non-computational activities to develop physics concepts first: "I would want to define velocity in the context of the classroom. We want to use the program as another [representation of the] model, but we're going to collect data and define what velocity is". Some teachers gravitated toward modeling activities (over simulations alone) that they felt could improve their pedagogical practice by giving students new outlets for problem solving: "There are some kids who can solve the problem with algebra but cannot do the graph. Every new representation exposes new conceptions, and I ran into that with my students this past year. There were some kids who showed me misconceptions when they were doing programming that did not show up on pencil and paper tasks".

Second, teachers called out the observable shift in students' thinking from almost exclusively time-parametric (algebraic) to differential. Teachers referenced that students were learning to "think about the world in terms of functions", and "thinking about what data you can get from the world", such as initial conditions that are quantifiable. Teachers drew a mental parallel between this choice and what they had already been doing in their classrooms with MI, which included using vector diagrams representing the position and velocity of objects at moments in time called motion maps. In describing the models she uses, one teacher mentioned how she had students correlate representations of motion: "It's like algebra and graphing (which is time-parametric), and motion mapping (which is differential)".

Teachers were also eager to think more about where else in the curriculum it might make sense to take such a differential worldview and how this might lend more support to students' initial tendencies to solve problems differentially. In a problem involving a faster bicyclist passing a slower bicyclist, one participant noted:

> "There's always that one group that the first thing they want to do is not do the [graphical or algebraic] representations, but want to make a motion map to solve the problem. They want to figure out what the position of each dot is for each second. [. . .] In all my years of doing this, when a kid has done that, I've always said, 'Ok, I guess you can do that. If that's what you need, then cool. But, it's always been kind of patronizing. . .then 'graph it, solve the equations.' Those kids, what they want to do is to write a computational simulation".

The need for a shift to differential thinking brought to light one of the advantages of teaching CM-integrated physics. Initially, it caused teachers to think about the problems and tasks they assigned to students in different ways. Typically, physics problems articulate a set of initial conditions and ask the solver to determine the condition at some particular point in time. Learning to think in terms of systems and state variables enables the problem solver to better manipulate the elements, operations, relations, and rules [58] of the model under investigation—it makes them better modelers.

Our initial version of the framework, which included these points of emphasis identified by teachers above, was produced in the early winter of 2017 (Table 5). These initial concepts strongly reflected each of Bootstrap:Algebra and MI's pedagogy and design. The framework had three themes. The Underlying Concepts of Integrated Computation theme captured ideas about computation as a process, the Affordances/Drawbacks of Computational Modeling theme focused on choosing a computational model to represent phenomena or solve problems, and the Computational Modeling Cycle theme articulated the steps that students take to define a specific computational model through code.

Table 5. Our initial Computational Modeling in Physics Framework.

Computational Modeling in Physics Framework
Items of high priority identified by teachers are starred (*).

- **Underlying Concepts of Integrated Computation**
    - Repetitive nature of computer simulations *
    - Role of parameters
    - Program execution
    - Error interpretation *
    - Defining physical variables and connecting them with real-world measurements
    - Alternative representations *
- **Affordances/Drawbacks of CM**
    - Identifying appropriate situations for CM *
    - Selecting the most appropriate representation
    - Explaining why CM is appropriate *
    - Explaining limitations of CM
- **Computational Modeling Cycle**
    - Defining the physical system *
        - Relevant Physical Constructs
            - Physical setup
            - Initial conditions
            - Constants (static)
            - Variables (dynamic)
            - Relationship (how constants and variables influence one another)
- Defining the goal of modeling the system
- Planning the program *
- Writing the functions
    - Running and using the simulation *
    - Refining or expanding on the existing model

However, as the academic year progressed, we realized that teachers emphasized some parts of programming over others. Teachers wanted more focus on the practices, emphasizing modeling processes. Many teachers struggled to find the time needed to dedicate to computing as a new representational tool, demonstrating a need to critically think about what to leave out, both in physics and computing. The framework was also not at the right level of granularity: this framework had been written to reflect the Design Recipe worksheets, but those worksheets were used in the context of specific problems that indirectly reinforced key modeling and programming concepts, such as when and how to capture a physical element as a static or a dynamic value. A more valuable framework would make these indirect concepts explicit so that they could be measured before and after students worked with our CM resources.

We also observed two major content-related obstacles to the implementation of the materials. First, many teachers lacked a holistic, concise storyline—for both physics and computing concepts and processes—that could interweave both disciplines. A few teachers were able to implement large swaths of the materials, and those who did so needed to develop additional materials to bridge the gap between units. Second, teachers found programming-related material to have a large learning curve. Too much time was spent on learning the underlying details of programs that were not relevant to physics, such as creating visual elements.

5.2. *Refining the Framework (2017–2018 Academic Year)*

In the following year, we made significant improvements to the curricular resources. We also realized that we needed to expand the scope of expertise on the curriculum development team. At our second development workshop, we made two design changes. Our response to the first obstacle was to invite back eight of the teachers in the original cohort and bring an additional 22 master MI teachers from across the country to join the development, including teachers who had more in-depth experience writing curriculum storylines. In response to the second obstacle, the Bootstrap:Algebra team developed libraries that hid much of the computing environment's inherent but unwanted details from students' work.

Reducing the cognitive load of the programming to allow more focus on the physics was a major change. For example, while writing a complete simulation entails rendering the physical state into an image, students learning physics need to concern themselves only with the system dynamics. Accordingly, we separated the physically relevant programming tasks from the programming-only tasks and hid the latter in "starter files" that students could use as a black box. Similarly, while Pyret makes it easy to extract the sequence of simulated states as a table for further processing, the initial export is in terms of data structures that are useful to the simulation, but irrelevant or unmeaningful to the students. Accordingly, the starter files post-process the data table into a simpler form that students use to analyze their results. For students who are interested in how the complete simulations are built, or how such data cleaning and processing can be performed, these topics are covered in Bootstrap's curricular projects, Bootstrap:Reactive and Bootstrap:Data Science.

Seeing that some teachers in this second development workshop still struggled with the programming tasks even after a week of being introduced to Bootstrap:Algebra, we dedicated the last day of the first week to teaching Bootstrap:Algebra with teachers in the student role. As a surprise, this revealed that many teachers struggled with understandings of the function concept from mathematics—a topic that would become a focal point for the following weeks and years.

Over two weeks, this group of 30 teachers collaborated on units of their choosing to revise and further develop curricular supports initiated the year prior. The Bootstrap:Algebra team helped teachers with programming, from fundamental understandings of functions to designing elaborate programs. Teachers participated in board meetings twice a day, led by author Megowan-Romanowicz, in which groups shared their progress with the whole cohort. Groups illustrated their unit development and how it fit into the overall physics and CM storylines by creating a visual display (Figure 2). At the end of the workshop, groups co-taught the materials they had developed to their peers and explicitly discussed how CM was integrated into the development and deployment models of the MI approach, covering introductory mechanics topics.

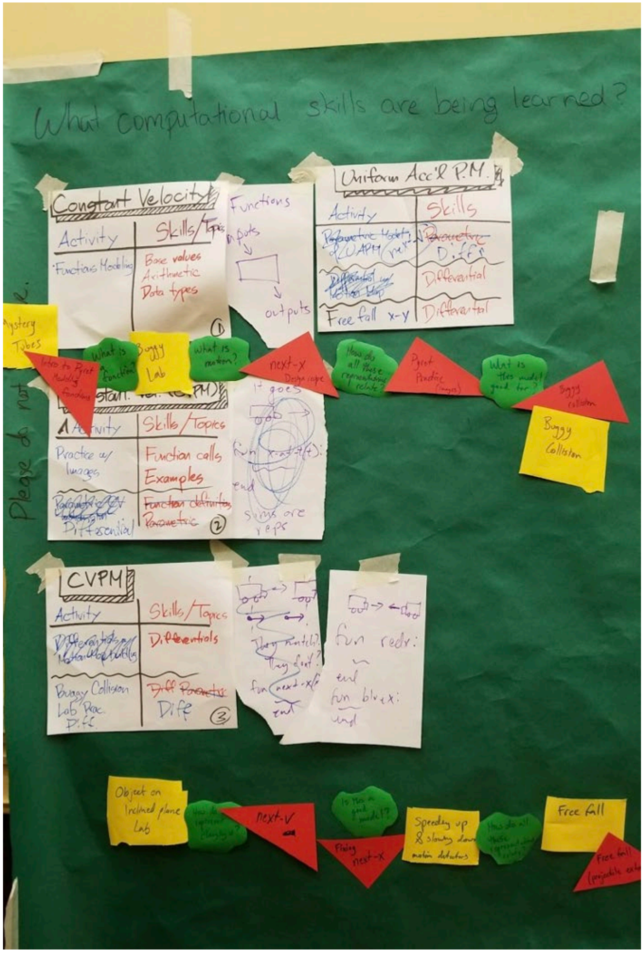


Figure 2. Initial teacher-developed alignment of computational modeling activities skills for the Constant Velocity Model and Uniform Acceleration Model units, along with hands-on laboratory activities (yellow rectangles), computing concepts (red triangles), and core questions (green clouds).

To boost classroom implementation over the following academic year, we asked each participant for a letter of administrative support, and we provided a stipend for evidence of implementation. About half of the 30 teachers from the summer 2017 workshop implemented the materials, with the remainder citing a lack of confidence or insufficient class time.

Recognizing the need to refine the materials, we selected three teacher participants who demonstrated significant programming and curriculum development skills. Two of these individuals were also experienced MI workshop leaders, suggesting a heightened awareness about what students need and what teachers need to take up innovative approaches. In early 2018, these teachers further refined the materials and developed the storyline using concept maps. Figure 3 illustrates the concept map for Unit 3, the Accelerated Particle Model, and demonstrates how CM is integrated into model development and deployment. Concept maps showing the progression of integrated concepts and skills are available from https://www.compadre.org/precollege/CMP/, (accessed on 1 August 2024).

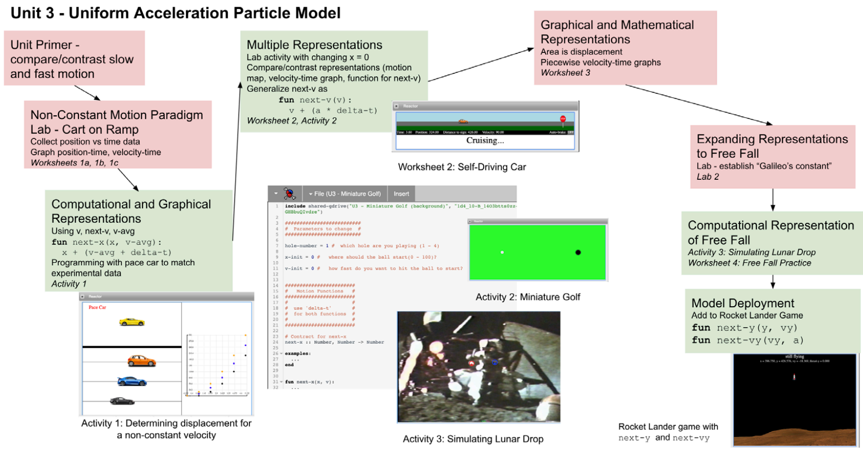


Figure 3. Final conceptual flow map for the Uniform Acceleration Model including physics components (pink), computational modeling components (green), and associated activities.

5.3. Finalizing Revisions (2018 and Beyond)

By early 2018, monthly conversations with teachers about what they were emphasizing in their use of the integrated materials clear that the original framework merited further revision. However, we waited to observe how the new curriculum materials would be received as part of a dissemination workshop with a combination of 23 teachers, both Modelers and non-Modelers. The summer 2018 workshop was modified from the prior years in two ways. First, given many teachers' struggles with functions, we began with three days of pure Bootstrap:Algebra to strengthen their understanding of functions. Second, as a dissemination workshop, rather than a curriculum-development workshop, the participants were expected to learn as students and reflect as teachers (Figure 4).

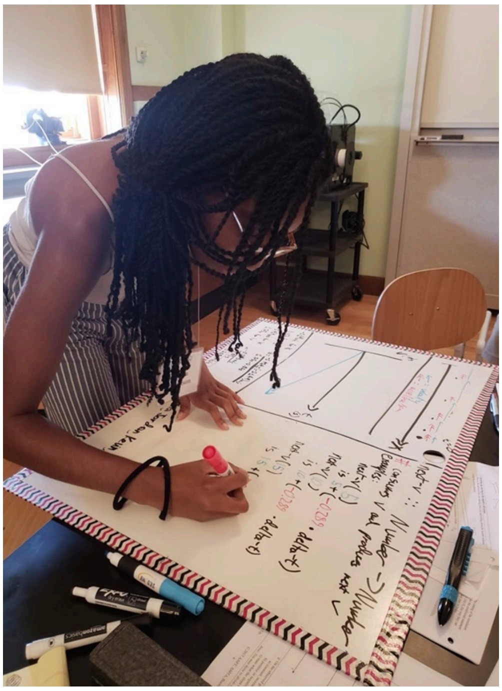


Figure 4. A teacher participant uses a whiteboard to display her understanding of acceleration with motion maps, graphs, algebraic equations, and a computational model (computer function).

As the workshop proceeded, we observed how teacher workshop leaders explicitly called out aspects of the framework for integration and which elements were most discussed by the participants. In conjunction with our observations that teachers struggled with functions, these observations led to the development of a revised framework that more explicitly focused on mathematical modeling with functions and that was more detailed to reflect the cycle of activities we saw in the curricular resources.

5.4. A Final Framework

To address the research question, "What do teachers believe should be the characteristics of an integrated computational modeling in a physics instructional framework?", we present the revised framework in Table 6. The framework has four themes: computational modeling, programming, prediction, and mathematical foundations. This revised framework moved from general themes that teachers believed characterized CM-integrated physics instruction toward more explicit student performance expectations that teachers believed served as evidence that students could carry out CM-integrated physics. In effect, while the initial framework addressed the question, "What is CM-integrated physics?", the revised framework addressed the question, "How do we know students are engaged in CM-integrated physics?"

Table 6. Revised Framework for Integrated Computational Modeling in Physics.

1. **The (Computational) Modeling theme**
    a. Able to recognize the data (including experimental data) that underlies a scenario, classifying each datum as *constant* (static), *dynamic*, or *stimulus* (driven by a user or external event)
    b. Able to identify which information about a scenario is irrelevant for a particular computational question
    c. Able to articulate or express a model that updates dynamic information in discrete steps, based on constants and stimuli; this model may be a function abstracted from experimental data
    d. Able to identify when a computational model is and is not appropriate for a given scenario
    e. Able to contrast, coordinate, and transition among computational and other representations of physical phenomena
    f. Able to identify errors in a proposed computational model for a given scenario
    g. Able to understand when each of an algebraic or differential representation of a model is appropriate for questions about a particular scenario
    h. Able to extend a model with new information to handle either more information or a variation in an existing model
2. **The Programming theme**
    a. Able to write a program to capture a specific model
    b. Able to articulate where each of constant, dynamic, and stimuli information is reflected in a program that captures a specific model
    c. Able to run/simulate a model to determine the state of the scenario after a given number of steps
    d. Able to locate and remove errors in a program intended to capture a specific model
    e. Able to identify where and how to edit an existing programmatic model to reflect a specific extension to a scenario
3. **The Predictive theme**
    a. Able to simulate/run a computational model to determine the value of dynamics after a given number of discrete steps
    b. Able to describe the behavior that results from a computational model
    c. Able to identify interesting instances under which to test a computational model of a scenario
4. **The Underlying Mathematics of Functions theme**
    a. Able to define a function that corresponds to a graph or table of observations
    b. Able to determine whether a collection of observations is consistent with a defined function
    c. Understands that functions capture computations that can be performed over different parameter values
    d. Able to compute the result that a function produces given values for its parameters

These new themes aligned better with the concepts and skills that teachers believe they are teaching. For example, general concepts about computing—such as the role of parameters and how functions evaluate—moved from the abstract-sounding Underlying Concepts concept to themes for the skills that use those underlying concepts. Specifically, some Underlying Concepts moved into each of the (Computational) Modeling theme (part of errors and defining variables), the Programming theme (parameters and part of errors), and the Predictive theme (understanding how models execute). There is still a separate Programming theme, but it is framed as a set of concrete modeling skills, rather than general CT concepts.

The original section on Affordances/Drawbacks of Computational Modeling is now part of the (Computational) Modeling theme, reflecting that computational representations are meant to be studied in the context of other representations as part of modeling. This theme is where the deep integration occurs, combining steps to create a model through a program (items 1a and 1c) and show when a computational model is appropriate (items 1d and 1e) while noting concerns such as program evolution and maintenance (item 1h) which are more typical of CT standards despite being a key part of scientific practice. While many teachers entered the development workshop attempting to write separate but parallel student performance objectives for using CM as a tool for science practice and science as an application area for CM practice, they eventually settled more directly on integrated objectives that reflected CM as a representation to display understanding about science. The revised framework also explicitly separates the simulation (Prediction) theme from the (Computational) Modeling theme, which we felt was important for noting the difference between simulation and modeling. Calling out the Underlying Mathematics of Functions theme explicitly helps convey the assumptions of our approach.

Making cross-disciplinary dependencies explicit also sets up some of our aspirations for transfer: we believe that integrated curricula that leverage concepts from math, computing, and physics could reinforce learning in each discipline on its own. We say this with a deep respect for the challenges of transfer in practice [59] and a solid understanding of what is required to achieve it from our work in CM/mathematics transfer in Bootstrap:Algebra [12,13]. This understanding leads to our embracing differential models and programming based on functions rather than variables, loops, and assignments.

6. Recommendations for Developing Integrated Frameworks

6.1. Teachers Must Balance Their Vision of Integration Opportunity with Instructional Needs

Our work revealed tensions between teachers' perceived benefits and limitations of MI and CM. We initially observed that teachers became preoccupied with procedural aspects of CM such as the steps to creating a program that models the physical world and explicit conversations about when and why CM should be used. Teachers grasped at activities in which physics and CM content and practices appeared to overlap, with little consideration for prioritization of knowledge or consideration for how it would benefit physics understanding rather than supplementing students' CM skills. As the program evolved, the possibilities for integration were dwarfed by time constraints. We also realized that some teachers struggled with a limited understanding of mathematical functions—something that was only revealed to us once we stepped away from the development process to reexamine teachers as learners. Tensions such as these are reported widely in STEM-related integration projects [60].

Only after attempting implementation were teachers able to refine what they valued and rationalize the integration of particular computational skills. These findings are reflected in the evolution from our initial framework, moving from a prescription to a process-oriented framework. The result was an increased focus on problem solving as a practice.

6.2. Frameworks Should Reflect Teachers' Values and Goals

Professional development programs should work with teachers as partners to incorporate their visions for integration into the frameworks they promote, with room for the framework to evolve as teachers learn more about their own and their students' interests, concerns, and needs. We found that attention to teachers' values and goals for instruction refined what we initially saw in terms of pathways for integration. This finding illuminates the risks inherent to excluding teacher voice through top-down projects that present teachers with fully prepared materials, which can exacerbate the tensions within integrated disciplines.

Our final framework reflects teachers' goals for integration and the critical features of the curricular materials they developed. We softened it to focus more on CM as a practice to promote an understanding of models of physics, as opposed to having programming as an ancillary goal. As such, the revised framework and resources aim to teach science content and processes using programming and software as tools and teaching CM as an approach to creating representations that display an understanding of how physical systems behave.

Critically, the framework reflects goals that speak to two affordances of integration. First, teachers recognized that there were areas in the content where CM could address challenging areas of physics pedagogy, such as the shift from constant velocity to uniform acceleration. This transition is notoriously difficult for introductory physics students who still have only an emergent grasp of parabolic functions. Second, the CM could allow the exploration of previously inaccessible physics scenarios at the introductory level, such as those that cannot be expressed in continuous functions or otherwise require analysis of complex interactions. For example, the motion and energy of a bouncing ball or gas molecule in a box cannot be easily analyzed without computational modeling due to their abrupt changes in direction and energy loss. Similarly, interesting phenomena such as the motion of rockets that lose mass throughout their trajectory are typically considered too complicated for introductory students. This framework displays how teachers believe the CM integration benefits physics as the core content comprehension of physics.

6.3. Distinguishing Teacher versus Student Needs as CM Learners

Professional development providers should help teachers explicitly recognize that their learning challenges in CM-integrated material may differ from those of their students. Teachers report that this shift from a time-parametric to a differential view of change was easier for students than it was for themselves. This finding came as a surprise to many of this project's lead researchers, as there was an expectation that the physics teachers in this program—most of whom had physics degrees—would be comfortable with differential expressions given their academic background. However, it is possible that these teachers' experience with secondary teaching, which has historically only emphasized time-parametric expressions, overshadowed teachers' prior academic experiences. Once teachers adopted a differential view of change in a physical system, the mathematical simplicity enabled them to address scenarios and guide their students in tackling problems that were outside the scope of introductory physics that would ordinarily require advanced algebra or calculus.

7. Discussion

We encouraged teachers to conjoin the CM skills defined in two established approaches (MI and Bootstrap:Algebra) by exposing teachers to both approaches. Our framework shifted in response to teacher feedback over multiple years. Our approach to developing this framework is not just the result of combining CM and physics content and processes, but of combining two distinct cultures of educators with their pedagogies, values, and identities. Our framework addresses the "need to develop a common language to describe [our] work" and is built upon an "iterative model of educational improvement" [7]. Our framework evolution resulted from learning what we did not know that we did not know—particularly, that many physics teachers were uncomfortable with functions. We did not know what teachers would value about the programming process, but we also did not expect them to integrate another discipline they were not masters of without significant investment in their learning, as integration would otherwise have likely been superficial [60]. Our work demonstrated the importance of balancing program leaders' vision with that of teachers, not just because teachers might see opportunities and obstacles not envisioned by program leaders, but to promote ownership and, ultimately, implementation [61].

Our work is different from others in multiple ways. First, it is evident from Weller et al.'s [5] literature review that few extant frameworks explicitly incorporate teacher voice (although Weller et al.'s own project uses an observational approach to note what teachers attend to in CT-integrated physics). Sengupta et al.'s [4] system design framework for agent-based CM builds off computing and science principles and practices. However, it does not explicitly illustrate teachers' perspectives on integration or their willingness to learn or implement the material. Weintrop et al.'s [3] taxonomy was built on a literature review and examples of CT-integrated lessons presented by math and science teachers who already used CT. Our own framework did not arise from abstracting over existing integrated materials, which might be limited by the perspective of the teachers who developed them.

Significant differences between our framework and those presented by Sengupta et al. and Weintrop et al. [3] lies in our explicit integration of computational and non-computational representations. In our framework, students explore observations across representations as part of helping them understand the affordances and limitations of computational models. Another difference is one of granularity: Weintrop et al.'s [3] taxonomy lists practices, while our framework lists skills that go into such practices. Our framework also explicitly calls out the mathematical foundations on which modeling depends.

8. Conclusions

Introductory high school physics courses should motivate confidence and an interest in the sciences and engineering. Supporting teachers with computational modeling competencies, the confidence to use them, and a teacher-created framework that reflects their own needs and aspirations present both structure and flexibility for both teachers and students. Giving teachers this measure of control and agency has been an important factor in their persistence in the program.

This DBR study illustrates emergent themes for the integration of CM in physics: it reflects teachers' vision for opportunities for integration tempered by practical classroom limitations, as well as emergent themes aligned with teachers' goals and values. Along the way, our teachers offered insights that might be informative to the anticipated process and expected products from other integrated CT efforts. The framework presented here results from merging key ideas from two established approaches (MI and Bootstrap:Algebra), iterating and refining materials through close collaboration with teachers and experts from each CT and physics education. By providing a conceptual framework based on differential modeling, and exercises that combine computational and non-computational representations, we present one view of a tight integration that we hope is useful to physics teachers and those who work to prepare and support them.

Author Contributions: Conceptualization, all authors; methodology, all authors; software, K.F., S.K., B.S.L. and J.G.P.; investigation, all authors; data curation, R.E.V. and C.M.-R.; writing—original draft preparation, K.F., S.K., C.M.-R. and R.E.V.; writing—all authors; project administration, R.E.V.; funding acquisition, R.E.V., C.M.-R., S.K. and K.F. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by 100Kin10 and the National Science Foundation under Grant No. 1640791. Publication of this article was funded by the University of Colorado Boulder Libraries Open Access Fund.

Institutional Review Board Statement: This project received IRB approval from the University of Maryland-College Park (IRBNet #1072769), effective 17 July 2017.

Informed Consent Statement: Informed consent was obtained from all individual participants included in this study, per the guidelines set forth by the University of Maryland-College Park.

Data Availability Statement: Qualitative data is unavailable due to complexity and, in some cases, privacy restrictions. Quantitative data is available upon request from the corresponding author.

Acknowledgments: The authors would like to acknowledge contributions to this work from the cohort physics teachers who participated in the workshops, to collaborators Fernand Brunschwig, Yi Li, and STEMteachersNYC for making the initial workshops possible through the arrangement and hosting of these workshops, and to evaluator Brandon Helding for his review of earlier versions of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. AAPT. *AAPT Recommendations for Computational Physics in the Undergraduate Physics Curriculum*; American Association of Physics Teachers: College Park, MD, USA, 2016; Available online: https://www.aapt.org/resources/upload/aapt_uctf_compphysreport_final_b.pdf (accessed on 1 August 2024).
2. *NSF #1647018*; Workshop to Develop an Interdisciplinary Framework for Integrating Computational Thinking in K-12 Science, Mathematics, Technology, and Engineering Education. Education Development Center: Waltham, MA, USA, 2018.
3. Weintrop, D.; Beheshti, E.; Horn, M.; Orton, K.; Jona, K.; Trouille, L.; Wilensky, U. Defining Computational Thinking for Mathematics and Science Classrooms. *J. Sci. Educ. Technol.* **2016**, *25*, 127–147. [CrossRef]
4. Sengupta, P.; Kinnebrew, J.S.; Basu, S.; Biswas, G.; Clark, D. Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Educ. Inf. Technol.* **2013**, *18*, 351–380. [CrossRef]
5. Weller, D.P.; Bott, T.E.; Caballero, M.D.; Irving, P.W. Developing a learning goal framework for computational thinking in computationally integrated physics classrooms. *arXiv* **2021**, arXiv:2105.07981.
6. Wing, J.M. Computational thinking. *Commun. ACM* **2006**, *49*, 33. [CrossRef]
7. National Academy of Engineering & National Research Council. *STEM Integration in K12 Education: Status, Prospects, and an Agenda for Research*; The National Academies Press: Washington, DC, USA, 2014.
8. AAPT. *AAPT Statement on Physics First*; American Association of Physics Teachers: College Park, MD, USA, 2002. Available online: https://www.aapt.org/Resources/policy/physicsfirst.cfm (accessed on 1 August 2024).
9. Chabay, R.W.; Sherwood, B.A. Computational physics in the introductory calculus-based course. *Am. J. Phys.* **2008**, *76*, 307–313. [CrossRef]
10. Bootstrapworld. Bootstrap. Available online: http://www.bootstrapworld.org (accessed on 1 August 2024).
11. Schanzer, E.; Fisler, K.; Krishnamurthi, S. Bootstrap: Going beyond programming in after-school computer science. In Proceedings of the SPLASH Education Symposium, Indianapolis, IN, USA, 26–31 October 2013.
12. Schanzer, E.; Fisler, K.; Krishnamurthi, S. Transferring skills at solving world problems from computing to algebra through Bootstrap. In Proceedings of the ACM Technical Symposium on Computer Science Education, Kansas City, MO, USA, 4–7 March 2015.
13. Schanzer, E.; Fisler, K.; Krishnamurthi, S. Assessing Bootstrap: Algebra students on scaffolded and unscaffolded word problems. In Proceedings of the ACM Technical Symposium on Computer Science Education, Toronto, ON, Canada, 15–18 March 2018.
14. Wells, M.; Hestenes, D.; Swackhamer, G. A modeling method for high school physics instruction. *Phys. Teach.* **1995**, *63*, 606–619. [CrossRef]
15. Hestenes, D. Toward a modeling theory of physics instruction. *Am. J. Phys.* **1987**, *55*, 440–454. [CrossRef]
16. Hestenes, D. Modeling instruction in mechanics. *Am. J. Phys.* **1987**, *55*, 455–462.
17. Lee, I.; Grover, S.; Martin, F.; Pillai, S.; Malyn-Smith, J. Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *J. Sci. Educ. Technol.* **2019**, *29*, 1–8. [CrossRef]
18. NGSS Lead States. *Next Generation Science Standards: For States, by States*; National Academies Press: Washington, DC, USA, 2013.
19. Osborne, J. Teaching scientific practices: Meeting the challenge of change. *J. Sci. Teach. Educ.* **2014**, *25*, 177–196. [CrossRef]
20. Malyn-Smith, J.; Lee, I. Application of the occupational analysis of computational thinking-enabled STEM professionals as a program assessment tool. *J. Comput. Sci. Educ.* **2012**, *3*, 2–10. [CrossRef] [PubMed]
21. Magana, A.J.; Couthinho, G.S. Modeling and simulation practices for a computational thinking-enabled engineering workforce. *Comput. Appl. Eng. Educ.* **2016**, *25*, 62–78. [CrossRef]
22. PhET Interactive Simulations. 2022. Available online: https://phet.colorado.edu/ (accessed on 1 August 2024).
23. Aiken, J.M.; Caballero, M.D.; Douglas, S.S.; Burk, J.B.; Scanlon, E.M.; Thoms, B.D.; Schatz, M.F. Understanding student computational thinking with computational modeling. In Proceedings of the AIP Conference Proceedings: 2012 Physics Education Research Conference, Philadelphia, PA, USA, 1–2 August 2012; American Institute of Physics: College Park, MD, USA, 2013; Volume 1413, pp. 46–49.
24. Stubbs, E.A.; Myers, B.E. Part of what we do: Teacher perceptions of STEM integration. *J. Agric. Educ.* **2016**, *57*, 87–100. [CrossRef]
25. Wang, H.-H.; Moore, T.J.; Roehrig, G.H.; Park, M.S. STEM integration: Teacher perceptions and practice. *J. Pre-Coll. Eng. Educ. Res.* **2011**, *1*, 2.
26. Basu, S.; Biswas, G.; Sengupta, P.; Dickes, A.; Kinnebrew, J.S.; Clark, D. Identifying middle school students' challenges in computational thinking-based science learning. *Res. Pract. Technol. Enhanc. Learn.* **2016**, *11*, 13. [CrossRef] [PubMed]
27. Orban, C.M.; Teeling-Smith, R.M. Computational thinking in introductory physics. *Phys. Teach.* **2020**, *58*, 247–251. [CrossRef]
28. Aksit, O.; Wiebe, E.N. Exploring force and motion concepts in middle grades using computational modeling: A classroom intervention study. *J. Sci. Educ. Technol.* **2020**, *28*, 65–82. [CrossRef]
29. Basu, S.; Biswas, G.; Kinnebrew, J.S. Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Model. User-Adapt. Interact.* **2017**, *27*, 5–53. [CrossRef]
30. Chabay, R.W.; Sherwood, B.A. *Matter & Interactions*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
31. Langbeheim, E.; Perl, D.; Yerushalmi, E. Science teachers' attitudes towards computational modeling in the context of an inquiry-based learning module. *J. Sci. Educ. Technol.* **2020**, *29*, 785–796. [CrossRef]
32. PICUP: Partnership for Integration of Computation into Undergraduate Physics; American Association of Physics Teachers: College Park, MD, USA, 2018. Available online: https://www.compadre.org/PICUP/ (accessed on 1 August 2024).

33. Sherin, B.L. A comparison of programming languages and algebraic notation as expressive languages for physics. *Int. J. Comput. Math. Learn.* **2001**, *6*, 1–61. [CrossRef]
34. Farris, A.V.; Dickes, A.C.; Sengupta, P. Learning to interpret measurement and motion in fourth grade computational modeling. *Sci. Educ.* **2019**, *28*, 927–956. [CrossRef]
35. Cobb, P.; Jackson, K.; Dunlap, C. Design research: An analysis and a critique. In *Handbook of International Research in Mathematics Education*, 3rd ed.; English, L.D., Kirshner, D., Eds.; Taylor and Francis, Inc.: Abingdon, UK, 2015; pp. 481–503. [CrossRef]
36. Cobb, P.; Jackson, K.; Smith, T.; Sorum, M.; Henrick, E. Design research within educational systems: Investigating and supporting improvements in the quality of mathematics teaching and learning at scale. *Natl. Soc. Study Educ.* **2014**, *112*, 320–349.
37. Collins, A. *Toward a Design Science of Education*; Center for Technology in Education: New York, NY, USA, 1990. Available online: http://cct2.edc.org/ccthome/reports/tr1.html (accessed on 1 August 2024).
38. Collins, A. Toward a design science of education. In *New Directions in Educational Technology*; Scanlon, E., O'Shea, T., Eds.; Springer: Berlin/Heidelberg, Germany, 1992; pp. 15–22.
39. Disessa, A.; Cobb, P. Ontological innovation and the role of theory in design experiments. *J. Learn. Sci.* **2004**, *13*, 77–103. [CrossRef]
40. Vieyra, R.; Himmelsbach, J. Teachers' disciplinary-boundedness in the implementation of integrated computational modeling in physics. *J. Sci. Educ. Technol.* **2021**, *31*, 153–165. [CrossRef]
41. Megowan-Romanowicz, C. Whiteboarding: A tool for moving classroom discourse from answer-making to sense-making. *Phys. Teach.* **2016**, *54*, 83–86. [CrossRef]
42. Jackson, J.; Dukerich, L.; Hestenes, D. Modeling instruction: An effective model for science education. *Sci. Educ.* **2008**, *17*, 10–17.
43. Johnson-Laird, P.N. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*; Harvard University Press: Cambridge, MA, USA, 1983.
44. Hestenes, D. Findings of the Modeling Workshop Project, 1994–2000. One Section of an NSF Final Report. 2000. Available online: https://davidhestenes.net/modeling/R&E/ModelingWorkshopFindings.pdf (accessed on 1 August 2024).
45. Brewe, E.; Kramer, L.; O'Brien, G. Modeling instruction: Positive attitudinal shifts in introductory physics measured with CLASS. *Phys. Rev. Spec. Top Phys. Educ. Res.* **2009**, *5*, 013102. [CrossRef]
46. Etkina, E.; Warren, A.; Gentile, M. The role of models in physics instruction. *Phys. Teach.* **2006**, *44*, 34–39. [CrossRef]
47. Cabot, L.H. *Transforming Teacher Knowledge: Modeling Instruction in Physics*; University of Washington: Washington, DC, USA, 2008.
48. Barlow, A.T.; Frick, T.M.; Barker, H.L.; Phelps, A.J. Modeling Instruction: The Impact of Professional Development on Instructional Practices. *Sci. Educ.* **2014**, *23*, 14–26.
49. Brewe, E. Modeling theory applied: Modeling Instruction in introductory physics. *Am. J. Phys.* **2008**, *76*, 1155–1160. [CrossRef]
50. Felleisen, M.; Findler, R.B.; Flatt, M.; Krishnamurthi, S. *How to Design Programs: An Introduction to Programming and Computing*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018. Available online: https://mitpress.mit.edu/9780262534802/how-to-design-programs/ (accessed on 1 August 2024).
51. K–12 CS Framework. 2017. Available online: https://k12cs.org/ (accessed on 1 August 2024).
52. AAPT. *Advancing Interdisciplinary Integration of Computational Thinking in Science: Conference Report*; American Association of Physics Teachers: College Park, MD, USA, 2020; Available online: https://www.aapt.org/Resources/upload/Computational_Thinking_Conference_Report_Final_200212.pdf (accessed on 1 August 2024).
53. Anderson, L.W.; Krathwohl, D. (Eds.) *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*; Longman: Harlow, UK, 2001.
54. Krajcik, J.; McNeill, K.L.; Reiser, B.J. Learning goals-driven design model: Developing curriculum material that align with national standards and incorporate project-based pedagogy. *Sci. Educ.* **2007**, *92*, 1–32. [CrossRef]
55. Garet, M.S.; Porter, A.C.; Desimone, L.; Birman, B.F.; Yoon, K.S. What makes professional development effective? Results from a national sample of teachers. *Am. Educ. Res. J.* **2001**, *38*, 915–945. [CrossRef]
56. Emerson, R.M. Observational field work. *Annu. Rev. Sociol.* **1981**, *7*, 351–378. [CrossRef]
57. Ericsson, K.A.; Simon, H.A. *Protocol Analysis: Verbal Reports as Data*; MIT Press: Cambridge, MA, USA, 1984.
58. Lesh, R.E.; Doerr, H.M. *Beyond Constructivism: Models and Modeling Perspectives on Mathematics Problem Solving, Learning, and Teaching*; Lawrence Erlbaum Associates Publishers: Mahwah, NJ, USA, 2003.
59. Bransford, J.D.; Schwartz, D.L. Rethinking transfer: A simple proposal with multiple implications. *Rev. Res. Educ.* **1999**, *24*, 61–100.
60. Pang, J.; Good, R. A review of the integration of science and mathematics: Implications for further research. *Sci. Sci. Math.* **2000**, *100*, 73–82. [CrossRef]
61. Kelley, T.R.; Knowles, J.G.; Holland, J.D.; Han, J. Increasing high school teachers' self-efficacy for integrated STEM instruction through collaborative community of practice. *Int. J. STEM Educ.* **2020**, *7*, 14. [CrossRef]