Why Computational Modeling in Physics?

The US lags behind developed nations in student STEM achievement, affecting economic competitiveness [1]. The "technology gap" between rich and poor, and between the advantaged and the underrepresented is increasing [2]. Despite the urgent need for a workforce skilled in computer science, relevant courses are not equally accessible to all students, especially in low-income schools that often serve minority populations [3]. Research has shown that taking computer science is a strong predictor of selecting a STEM major at a four-year college [4], and that the effect is slightly stronger for underrepresented minority groups than for White or Asian students. However, teacher preparation and engaging curricula, especially curricula that integrate the naturally synergistic fields of computational modeling, science, and engineering, are lacking. The "Computer Science for All" initiatives of President Obama [5] and the CS10K Community supported by the NSF [6] address the problem by incentivizing more study of computer science. However, we argue that the more challenging, and potentially rewarding, possibility is to leverage the connections between computer science and the physical sciences.

Many school districts are adopting the Next Generation Science Standards [7]**, which require students to create, refine and use models - a central practice for scientists and engineers.** The ability to use new technologies to "manage and make meaning from the large amounts of data they produce is becoming a defining feature of scientific work in the 21st century and thus critical to computational thinking in mathematics and science, underscoring the importance of developing computational thinking data practices in the classroom" [8]. Given the reality that fewer than half of teachers of physics have a major or minor in physics, many teachers are unprepared to teach their students how to do so [9]. This problem is further exacerbated by the fact that there has been very little research at the K-12 level to guide physics teachers in how to effectively incorporate computational modeling into their coursework in a way that supports what they were prepared to do: to teach science.

Very few resources exist to support secondary teachers or students with computational modeling in physics. There has been extensive and ongoing research done on computational modeling in physics at the undergraduate level spearheaded by the AAPT, including NSF DUE 1432363 "Surveying the State and Implications of Computational Physics in Courses for Physics Majors," NSF DUE 1505278 "Fostering Introduction of Computation into Physics Courses: A Local Communities Approach," and NSF DUE 1524963 "Integrating Computation into Undergraduate Physics – Building a Sustainable Community through Faculty Development". However, relatively little has been done either to prepare secondary physics teachers to teach computational modeling or to support students in their learning of it. Successful students require good teachers, especially students from poor backgrounds in under-resourced schools. Computer Science for All – all students in all schools - is one approach with a worthy goal. Science teachers attempting to pursue this face three interlinked problems: 1) glaring gaps in their own knowledge of computer science, 2) an absence of computer science teaching workshops tuned to their needs, and 3) a dearth of curriculum materials that connect directly with the often-mandated scientific content and that the use of programming to solve problems and build models, not merely to create an effect on the screen.

When computational modeling is included only in advanced physics courses, the achievement gap widens for underrepresented minorities. The data on who takes advanced physics courses is clear: physics students at advantaged schools are twice as likely to take advanced physics (Advanced Placement and 2nd year courses) as students at disadvantaged schools [14]. Likewise, physics students at disadvantaged schools are three times more likely to take it through a Physics First course. A recent survey of New York City demonstrated that AP Physics course enrollments are low among Black and Hispanic populations, and that AP Physics is significantly more likely to be available in those schools that have triple the average percent makeup of White and Asian students [15]. Females are another underrepresented group in advanced physics courses: only 32% of AP Physics C students were female in

2009. The only STEM-related AP courses that had a lower representation of women at the time was Computer Science A, with 19% females, and Computer Science B, with 13% [16]. Including computational modeling in advanced physics courses only widens the achievement gap, because Black and Hispanic students are disproportionately served by disadvantaged schools, and because women are significantly less likely to be enrolled in advanced physics.

While there is a relatively high use of educational technology – including computers – in physics education for the purpose of collecting experimental data and using simulations [17]**, teachers and students are typically "end users."** Though beneficial to computational thinking broadly, neither teachers nor students get the exposure to programming environments that define many of the technical aspects of work in CS. Misguided efforts at teaching coding, in which learning a programming language is an end in itself, have also not served the larger purpose of enabling students to engage in computational thinking to understand the physical world and how it works and thus be ready to succeed in the academic, scientific, and engineering worlds. There is a lack of cohesion among the way students are exposed to and learn mathematics, physics and computer science. As expressed by a 2010 report form the National Academies [18], it is vital to merge approaches that incorporate general computer literacy, programming skills, and games and simulations, and to draw parallels between computational thinking, mathematics, and engineering. The report makes an excellent case for the need to incorporate computation in the context of an algebra-based science course.

While there continues to be discussion about the definition of computational thinking [18]**, there are significant efforts to define computational thinking in the context of science courses.** A comprehensive, contextualized definition developed by Weintrop et al. [8] includes an insightful taxonomy of computational thinking in math and science courses, including a productive elucidation of modeling, that is instrumental for all scientific model building, especially via the computer, and for efforts to help students practice computational thinking and gain perspective on their thinking. The taxonomy divides computational thinking into four types of practices (Figure 1): data, modeling and simulation, computational problem-solving, and systems thinking.

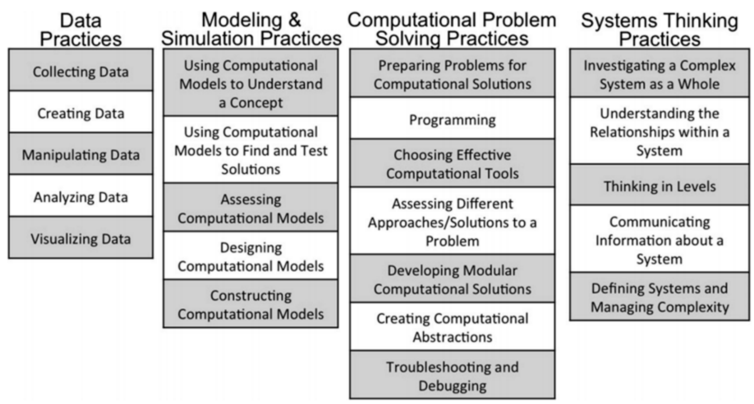

Figure 1: Weintrop's Hierarchy of Computational Thinking in Math and Science Courses

The use of computational modeling to understand physics concepts and to solve problems can engage students in <u>many</u> computational thinking practices, not just "Modeling & Simulation Practices" as defined by Weintrop. The *Framework for K-12 Science Education* report [19], which serves as the intellectual basis for the Next Generation Science Standards, is founded on practices associated with data, modeling, problem-solving, and systems thinking. The use of computers to apply these practices is essential in many STEM degrees, including undergraduate physics. However, of the science-contextualized computational modeling lessons identified and studied by Weintrop, the majority were narrowly used for exploring concepts through front-end simulations, not for analysis of data, problem-solving, or systems thinking, suggesting that teachers and students are missing out on vital opportunities for using computational thinking to think critically about problems. Additionally, the lack of clear exposure to computational problem solving (including coding) fails to give students the real exposure that they need to see what is "under the hood" of many simulations.

Why Bootstrap and Modeling?

This funded work (NSF #1640791) will merge Bootstrap's programming model and engage existing Modeling Instruction physics teachers to incorporate computational modeling in their courses. Bootstrap (Figure 2) and Modeling Instruction in Physics are two separate existing programs that, if used together, have the potential to overcome many of the challenges associated with the lack of resources for teachers and the existing lack of equity for students. In what follows, we describe the Bootstrap program and Modeling Instruction.

Bootstrap [20] is a computational modeling professional development program for teachers accompanied by student curriculum, an instructional approach, and rooted in pedagogy [21] that was designed to help students learn mathematics (algebra and geometry) through the programming of their own video game. Students typically begin by designing simple side-scrolling games involving a *player*, a *target* to be achieved, and *obstacles* to be avoided.

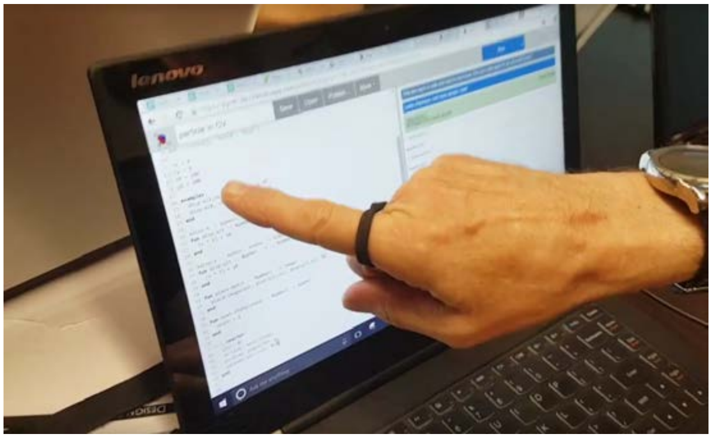

Figure 2: Pyret editor used in Bootstrap 2.

Bootstrap provides 3-5 days-long professional development to teachers, and assistance with implementation in schools. Currently, Bootstrap offers two levels of professional development for teachers (Bootstrap 1 and 2). Pedagogically, teachers are taught during the professional development experience to help students to construct a variety of models when building code (often, with the end goal of modeling motion). For example, mathematical understandings about orders of operation for numbers and variables are visualized using nested "Circles of Evaluation," and this is then translated into code. The simple motion programmed into the game frame-by-frame is represented by functions in the code, which support mathematics as well as physical understandings about motion. Visual interactive games are an end-product of student work, and significant research has already been done demonstrating the high value of interactive simulations in science engagement and understanding through inquiry [22] [23]. Student learning is scaffolded through a "Design Recipe" as they attempt to solve word problems through mathematical modeling: this design recipe ensures that students understand the domain, range, and examples of the function they intend to code. Importantly, the underlying programming environment used by Bootstrap differentiates itself from other approaches in that functions written by students are "side-effect-free," and more accurately reflect the nature of the algebra than other environments. This instructional approach is rooted deeply in work associated with *Programming by Design* [24] and the resulting publication *How to Design Programs* [25]. The scaffolded exposure of students to coding and programming through the development of algebraic ideas by developing mathematical, diagrammatic, and graphical models very closely mimics many of the research-based techniques used by physics teachers.

All Bootstrap material is closely aligned to the existing Computer Science Teacher Association standards [26]**,** while also supporting Common Core Standards for Mathematical Practice [27]. Because modeling relationships in physics requires an understanding of simple algebraic relationships, Bootstrap can serve a three-way purpose to help students understand computational modeling (Table 1), physical modeling, and mathematical modeling. Bootstrap 1 helps students to learn and reinforce physics-relevant algebra skills such as word problem-solving, coordinate planes, order of operations, variables, functions, domain and range, and the Pythagorean Theorem. Bootstrap 2 presents students with an opportunity to develop an understanding of complex functional relationships, randomness, and trigonometry.

Table 1: Computational Modeling Content in Bootstrap

Bootstrap 1	Bootstrap 2
• Numbers • Strings • Images • Defining Functions • Unit Testing • Boolean Logic • Multi-Input Functions • Mixed-Type Functions	• Event-Driven Programming • Data Structures • Whole-Program Design • Data Modeling • Encapsulation • Connections to recursion, lists, and algorithms

Evaluations of Bootstrap's program demonstrate students' understanding of computational modeling through the creation of their computer game products, in addition to significant gains in algebra [28]. Bootstrap is very accessible to teachers who often have no experience with computer science at all, and students are highly engaged by the curriculum [28]. This year alone, over 400 teachers have taken Bootstrap's workshops. Since 2011, Bootstrap has impacted more than 13,000 students. Importantly, of the teachers in the workshops, the majority had no degree in either math or computer science, yet their fidelity to program implementation is very high, suggesting that Bootstrap is accessible to both teachers and their students regardless of background.

Modeling Instruction, initially developed at Arizona State University [29], is a model-based instructional approach to teaching science through inquiry, and includes a set of resources that has continually demonstrated significant gains in student conceptual knowledge [30]. Modeling Instruction is a research-based method created for secondary physics education [31], and was developed through the support of seven separate NSF grants (NSF MDR-895461, NSF ESI-9353423, NSF PHY-9819461, NSF DUE 9910458, NSF ESI-0138561, NSF DUE-9952706, and NSF I^3-0930103). It has been recognized by Change the Equation as a featured STEMWorks program with the highest possible ranking of "Accomplished" [32].

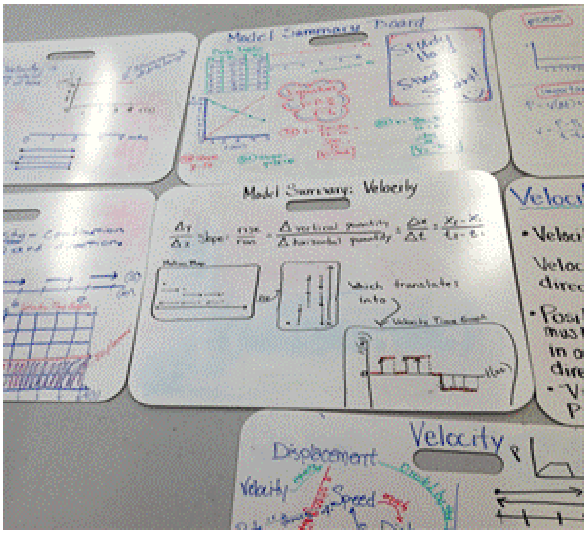

Figure 4: Example mathematical, graphical, diagrammatic, and verbal models in kinematics.

Currently, there are over 8,000 "Modelers" (teachers who use Modeling Instruction) in the U.S. and around the world impacting almost a million students annually. This popular pedagogy resulted in the formation of the American Modeling Teachers Association (AMTA) [33] in 2005. The AMTA has a sustaining online community and offers 60+ two- to three-week-long workshops each year (80-120 hours), primarily in the U.S., with virtual follow-ups throughout the year. STEMteachersNYC, founded in 2011, supports a tri-state community of 550+ teachers and presents about 10% of the 60+ two- to three-week Modeling workshops and annually conducts 20+ weekend workshops led by teachers. A number of universities from the Physics Teacher Education Coalition [34], a group of 300+ physics teacher education universities across the U.S., use Modeling Instruction materials through in-person or online formats as part of their discipline-specific teacher preparation.

The Modeling approach to physics content implicitly emphasizes many computational thinking skills. The "models" to which Modeling Instruction refers are the fundamental conceptual models that students develop, refine, and deploy (e.g. "Uniform Motion" and "Force-Particle" models in mechanics, and "Particle" and "Wave" models in light). In developing these conceptual models, students use inquiry approaches to collect data to build graphical, diagrammatic, algebraic, and verbal representations (Figure 4). These understandings are frequently supported by visual or even physical models. Each modeling cycle (Table 2) includes two phases and support many computational thinking skills [8]: (1) Model development: students perform experiments to understand physical relationships, and (2) Model deployment: students apply, test, and refine their model to make predictions. In both phases, data practices and systems thinking practices can be found.

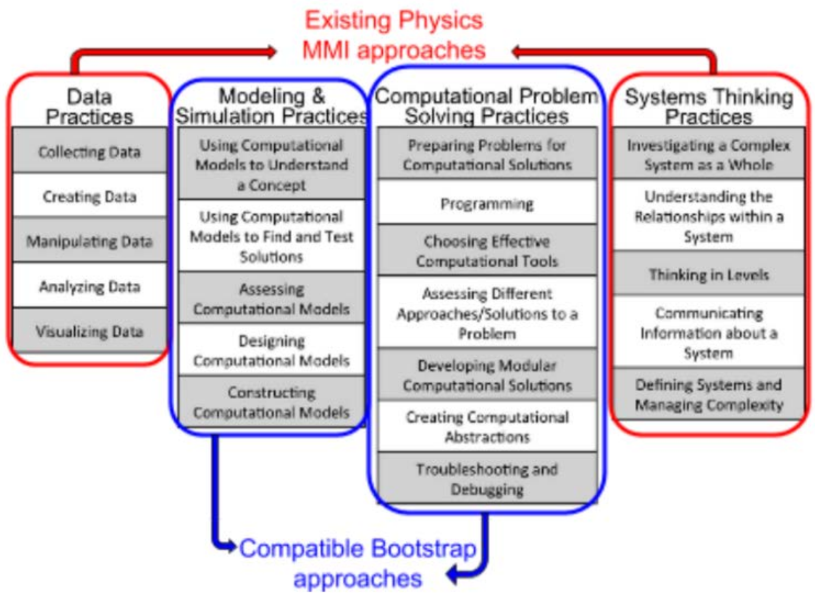

Figure 4: How Bootstrap and MMI can be integrated to cover Weintrop's hierarchy.

Bootstrap resources closely support methods used in Modeling instruction to have students represent their understandings through the development and refinement of graphical, algebraic, and visual models. In a typical Modeling cycle, students collect data, display data graphically, derive algebraic expressions from the graph, and continue to refine and apply their models through the use of a variety of representations, including drawn graphics (ex: motion maps, pie charts, bar graphs) and simulations. Bootstrap can be incorporated into nearly all of these components using many of Weintrop's defined computational thinking skills (Figure 5). For example, during the study of accelerated motion of a rocket, students might experimentally collect data from video analysis of a real rocket. Bootstrap programming can be incorporated to help students use the collected data (*Data Practices*) to develop mathematical models to describe this motion iteratively (*Modeling & Simulation Practices*), and

Table 2: Computational Thinking Skills in the Modeling Phases

Model Development	Model Deployment
• Data Practices o Collecting o Creating o Manipulating o Analyzing o Visualizing	• Systems Thinking Practices o Investigating complex systems as a whole o Understanding the relationship within a system o Thinking in levels o Communicating Information about a System o Defining Systems and Managing Complexity

to modify variables – such as the mass of the rocket - in a computational environment to predict the effect of those changes through coding (*Computational Problem Solving Practices*). The students can also increase the complexity of the system, such as adding in air friction for a vertically-moving rocket at low and high speeds and through air masses of varying densities (*Systems Thinking Practices*).

Together, these two programs provide math and science as appropriate contexts [35] **for the learning of computational skills by *all* students.** Notably, significant gains in student achievement in students who learned algebra through the Bootstrap program were demonstrated in schools with a majority of racial and ethnic minority and low income students [28]. Early research also suggests that Modeling Instruction is an effective approach to equity in introductory physics, with one study at a predominantly Hispanic-serving institution demonstrating that the Modeling approach significantly increases a student's chance of success in physics compared to traditional teaching, yielding the same rate of success for Hispanics as for non-Hispanics [36].

Together, the proposal partners will work to understand how computational thinking can be fully integrated into introductory Physics First courses and to examine its impact on students' self-efficacy and performance. Although there are some attempts to include computer usage in physics, what is currently being done is not enough to build a "CS-ready" identity in physics teachers and their students. Comprehensive research on the ways that computing is incorporated and even defined in science classroom settings suggests that data collection tools and simulations are often used but they do not expose students to the programming that run these products [8]. For example, students might collect and display data on a graphical analysis tool, perform video analysis of an object in motion, or even work with a virtual simulation showing projectiles at various launch speeds and angles. However, while these studies might encourage some aspects of computational thinking, students do not engage in programming or gain an understanding of how simulations actually work. Additionally, teachers often lack a framework for how to effectively incorporate computational tools along the full spectrum of conceptual model development [37]. Simulations are frequently reserved for "single exposures," often to reinforce, enhance, or assess concepts at the end of a learning cycle, not throughout the full progression of learning. It is only through deeply embedded computational modeling in the conceptual development of science models (model development and model deployment) that science teachers and their students will develop a "CS-ready" identity.

These broad computational skills can be made explicit, applied to computational physics modeling problems, and quantified through their application in the Bootstrap programming environment. To examine its impact on students' self-efficacy and performance, this proposal will answer two overarching questions about engaging science teachers and their students with computational thinking: (1) How does science instruction that integrates computational modeling impact student performance and confidence in the application of computational modeling to solve problems in physics? (2) How does engaging Physics First teachers to incorporate computational modeling in their teaching practice impact their curriculum and instruction?

Appendix 1: Elements of Integration of Computational Thinking (CT), Modeling, and Physics

Bootstrap CT	Integrated CT/STEM Skills	Modeling Physics First
Instructional Approach / Skills 1. Identification of physical constants and variables in an existing system 2. Identification of both the static and dynamic aspects of a problem/system 3. Writing test cases 4. Developing abstractions 5. Selecting programming constructs that perform needed computations 6. Strategic debugging of programs	*CT-STEM Hierarchy* (Weintrop, et al) 1. Data Practices 2. Modeling & Simulation Practices 3. Computational Problem Solving Practices 4. Systems Thinking Practices	*Instructional Approach / Skills* 1. Model Development 2. Model Deployment
CT Content 1. Numbers, strings, images 2. Defining functions 3. Unit testing 4. Boolean logic 5. Multi-argument functions 6. Tabular data 7. Tabular processing 8. Event-driven programming 9. Lists/Data structures	*NGSS SEP #5* (Using Mathematics and Computational Thinking) 1. Decide if qualitative or quantitative data are best... 2. Create and/or revise a CM... 3. Use mathematical, computational, and/or algorithmic representations... 4. Apply techniques of algebra and functions... 5. Use simple limit cases to test mathematical expressions, computer programs, algorithms, or simulations... 6. Apply ratios, rates, percentages, and unit conversions…	*Physics Content* 1. Descriptive Models (Motion) 2. Causal Models (Newton's Laws) 3. Qualitative Energy Models (Energy) 4. Particles Models (Matter) 5. Wave Models (Light & Sound)

Appendix 2: Examples of Potential Integrations of Computational Modeling (CM)

MMI™ Physics First Models	Physics and Bootstrap-integrated CM tasks	NGSS Connections
Descriptive Models • Constant Velocity • Uniform Acceleration	Develop a CM that iteratively displays changes in position as a result of constant velocity or uniform acceleration in one and/or two dimensions. • *Students fluidly display motion via CM and other models - graphically, mathematically, using vector diagrams, etc. - to solve problems, such as determining final speed of a falling rock or maximum height of a vertical rocket..* • *Students apply functions derived from experimental data (xf = v t + xi and vf = a t + vi) to more complicated situations, such as multi-part motion or relative motion.*	Disciplinary Core Ideas • PS2: Motion and Stability - fundamental understandings of motion Science & Engineering Practices • #2: Developing and Using Models: Students develop and/or use multiple types of models to provide mechanistic accounts and/or predict phenomena, and move flexibly between model types based on merits and limitations. • **#5: Using Mathematics and Computational Thinking**: Students create and/or revise a computational model or simulation of a phenomenon, designed device, process, or system. Apply techniques of algebra and functions to represent and solve scientific and engineering problems.

Causal Models • Balanced Forces • Unbalanced Forces • Central Forces & 2D Motion	Develop a CM that displays particle interactions of two or particles in one dimension. • *Students predict the resulting motion of two colliding objects of various mass and speed, and students predict the effect of a central force acting on an initially linearly-moving particle).* • *Students apply functions derived from experimental data (Fnet = ma, pf = pi) to more complicated situations, such as multi-particle interactions.*	Disciplinary Core Ideas • PS2: Motion and Stability: A. Newton's Second Law accurately predicts changes in the motion of macroscopic objects. Momentum is defined for a particular frame of reference; it is defined as mass times the velocity of the object. If a system interacts with objects outside itself, the total momentum of the system can change; however, any such change is balanced by changes in the momentum of objects outside the system. Science & Engineering Practices • #2: Developing and Using Models: Students develop, revise, and/or use a model based on evidence to illustrate and/or predict the relationships between systems or between components of a system. • **#5: Using Mathematics and Computational Thinking**: Students create and/or revise a computational model or simulation of a phenomenon, designed device, process, or system. Apply techniques of algebra and functions to represent and solve scientific and engineering problems.
	Develop a CM that displays the effects of an object undergoing two or more forces at the same time (i.e. accelerating rocket under the influence of gravity, thrust, and drag). • *Students compare and contrast situations in which friction is truly negligible or has a significant impact on motion.* • *Students revise their understandings about free fall in systems where friction is negligible to systems in which friction is significant.*	Disciplinary Core Ideas • PS2: Motion and Stability: A. Newton's Second Law accurately predicts changes in the motion of macroscopic objects. Science & Engineering Practices • #2: Developing and Using Models: Students evaluate merits and limitations of two different models of the same proposed tool. • #4: Analyzing and Interpreting Data: Students evaluate the impact of new data on a working explanations and/or model of a proposed process or system.
Qualitative Energy Models • Mechanical Energy • Energy Transfer	Develop a CM that displays energy transformations in a simple, closed system. • *Students create representations of energy transformations in simple situations where energy transformations are not always 100% efficient.*	Disciplinary Core Ideas • PS3: Energy: A. Energy is a quantitative property of a system that depends on the motion and interactions of matter and radiation within that system. That there is a single quantity called energy is due to the fact that a system's total energy is conserved, even as, within the system, energy is continually transferred from one object to another and between its various possible forms. B. Conservation of energy means that the total change of energy in any system is always equal to the total energy transferred into or out of the system. Science & Engineering Practices • #2: Developing and Using Models: Students develop, revise, and/or use a model based on evidence to illustrate and/or predict the relationships between systems or between components of a system. • **#5: Using Mathematics and Computational Thinking**: Students create and/or revise a computational model or simulation of a phenomenon, designed device, process, or system. Apply techniques of algebra and functions to represent and solve scientific and engineering problems. Apply ratios, rates, percentages, and unit conversions in the context of complicated

		measurement problems involving quantities with derived or compound units.
Particle Model of Matter • Evidence for Particle Model • Interacting Particle Model	Develop a CM that displays multiple particles as they interact with one another in 2D. • *Students create representations of energetic particles in motion and represent variations in heat and temperature.*	Disciplinary Core Ideas • PS1: Structure and Properties of Matter: A. The structure and interactions of matter at the bulk scale are determined by electrical forces within and between atoms. • PS3: Energy: A. Energy is a quantitative property of a system that depends on the motion and interactions of matter and radiation within that system. Science & Engineering Practices • #2: Developing and Using Models: Students develop a complex model that allows for manipulation and testing of a proposed process or system. • **#5: Using Mathematics and Computational Thinking**: Students create and/or revise a computational model or simulation of a phenomenon, designed device, process, or system. Apply techniques of algebra and functions to represent and solve scientific and engineering problems. Apply ratios, rates, percentages, and unit conversions in the context of complicated measurement problems involving quantities with derived or compound units.
Waves Models • Oscillating Particles • Mechanical Waves • Sound	Develop a CM that displays the effects of superposition and interference of waves as a way to transfer, convert analog/digital data, and encode information. • *Students send and receive "messages" through created representations of analog data as superimposed waves, Students use appropriate selection procedures to determine how to convert the analog signal into a digital signal without losing important information.*	Disciplinary Core Ideas • PS4: Waves and Their Applications in Technologies for Information Transfer: A. Information can be digitized; in this form it can be stored reliably in computer memory and sent over long distances as a series of wave pulses. Waves can add or cancel one another as they cross, depending on their relative phase, but they emerge unaffected by each other. • ETS: Engineering, Technology, and the Application of Science Science and Engineering Practices • #2: Developing and Using Models: Students develop and/or use a model to generate data to support explanations, predict phenomena, analyze systems, and/or solve problems. • #4: Analyzing and Interpreting Data: Students consider limitations of data analysis when analyzing and interpreting data. • **#5: Using Mathematics and Computational Thinking**: Students create and/or revise a computational model or simulation of a phenomenon, designed device, process, or system.

References

[1] N. R. Augustine, Rising above the gathering storm: Energizing and employing America for a brighter economic future., Washington, DC: National Academies Press, 2005.

[2] D. P. Driscoll, "Technology and engineering literacy framework for the 2014 National Assessment of Educational Progress.," U.S. Department of Education, Washington, DC, 2014.

[3] Y. Kafai, K. Peppler and G. Chiu, "High tech programmers in low-income communities: seeding reform in a community technology center.," in *Proceedings of communities and technologies 2007*, London, 2007.

[4] A. Lee, "Determining the effects of computer science education at the secondary level on STEM major choices in postsecondary institutions in the United States," *Computers and Education,* vol. 88, no. October, pp. 241-255, 2015.

[5] M. Smith, "Computer Science for All," White House, 30 January 2016. [Online]. Available: https://www.whitehouse.gov/blog/2016/01/30/computer-science-all. [Accessed 19 March 2016].

[6] "CS10K," National Science Foundation, [Online]. Available: https://cs10kcommunity.org/. [Accessed 19 March 2016].

[7] NGSS Lead States, Next generation science standards: For states, by states., Washington, DC: National Academies Press, 2013.

[8] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille and U. Wilensky, "Defining computational thinking for mathematics and science classrooms," *Journal of Science Education Technology,* vol. 25, no. 1, pp. 127-147, February 2016.

[9] S. White and J. Tyler, "Who teaches high school physics? Results from the 2012-13 nationwide survey of high school physics teachers.," American Institute of Physics, College Park, MD, 2014.

[10] R. Chabay and B. Sherwood, Matter and Interactions, 4th Ed., New York, NY: Wiley, 2015.

[11] M. D. Caballero, J. Burk, J. Aiken, B. Thoms, S. Douglas, E. Scanlon and M. Schatz, "Integrating numerical computation into the Modeling Instruction curriculum," *The Physics Teacher,* vol. 52, no. 1, pp. 38-42, January 2014.

[12] B. Sherwood, "VPython: 3D programming for ordinary mortals," [Online]. Available: www.vypython.org. [Accessed 22 March 2016].

[13] D. Sherer, P. Dubois and B. Sherwood, "VPython: 3D interactive scientific graphics for students," *Computing in Science & Engineering,* vol. 2, no. 5, pp. 56-62, 2000.

[14] S. White and C. Langer Tesfaye, "High school physics courses and enrollments: A 2012-13 nationwide survey of high school physics teachers," American Instute of Physics, College Park, MD, 2014.

[15] "Issues of Equity in Physics Access and Enrollment," National Association of Black Physicists, 6 August 2015. [Online]. Available: http://vector.nsbp.org/2015/08/06/issues-of-equity-in-physics-access-and-enrollment/. [Accessed 19 March 2016].

[16] S. White and C. Langer Tesfaye, "Female students in high school: Results from the 2008-2009 nationwide survey of high school teachers," American Institute of Physics, College Park, MD, 2011.

[17] S. White and C. Langer Tesfaye, "High school physics textbooks, resources, and teacher resourcefulness," American Institute of Physics, College Park, MD, 2014.

[18] National Academies, "Report of a workshop on the scope and nature of computational thinking," in *National Academies Press*, Washington, DC, 2010.

[19] National Academies, A Framework for K-12 Science Education, Washington, DC: National Academies Press, 2011.

[20] "Bootstrap," [Online]. Available: http://www.bootstrapworld.org/. [Accessed 22 March 2016].

[21] E. Schanzer, K. Fisler, S. Krishnamurthi and M. Felleisen, "Bootstrap: Transferring skills at solving word problems from computing to algebra," in *Proceedings of SIGCSE*, Association of Computing Machinery, 2015.

[22] E. Moore, T. Herzog and K. Perkins, "Interactive simulations as implicit support for guided-inquiry," *Chemistry Education Research Practices,* vol. 14, no. 3, pp. 257-268, 2 April 2013.

[23] N. Finkelstein, W. Adams, C. Keller, P. Kohl, K. Perkins, N. Podolefsky, S. Reid and R. LeMaster, "When learning about the real world is better done virtually: A study of substituting computer simulations for laboratory equipment," *Physical Review Special Topics,* vol. 1, no. 1, pp. 010103-1 - 010103-8, 6 October 2005.

[24] S. Bloch, J. Clements, M. Felleisen, R. Findler, K. Fisler, M. Flatt, V. Proulx and S. Krishnamurthi, "Program by Design," [Online]. Available: http://www.programbydesign.org/. [Accessed 22 March 2016].

[25] M. Felleisen, R. Findler, M. Flatt and S. Krishnamurthi, How to design programs, Boston, MA: MIT Press, 2001.

[26] C. S. T. Association, "CSTA K-12 Computer Science Standards," Computer Science Teachers Association, 2011.

[27] C. C. S. S. Initiative, "Common Core Standards for Mathematical Practice," [Online]. Available: http://www.corestandards.org/Math/Practice/. [Accessed 19 March 2016].

[28] W. McClanahan, S. Pepper and M. Polin, ""I program my own videogames": An evaluation of Bootstrap," McClanahan Associates, 2016.

[29] M. Wells, D. Hestenes and G. Swackhamer, "A modeling method for high school physics instruction," *American Journal of Physics,* vol. 63, no. 7, pp. 606-619, 1995.

[30] D. Hestenes, "Notes for a modeling theory of science, cognition, and instruction," in *Proceedings of the 2006 GIREP Conference*, 2006.

[31] C. Megowan-Romanowicz, "Helping students construct robust conceptual models," in *Models and modeling: Cognitive tools for scientific enquiry*, London, Springer Netherlands, 2011, pp. 99-120.

[32] "STEMWorks," [Online]. Available: http://www.changetheequation.org/stemworks.

[33] "American Modeling Teachers Association," 2016. [Online]. Available: https://modelinginstruction.org/.

[34] "Physics Teacher Education Coalation," [Online]. Available: http://www.phystec.org/. [Accessed 19 March 2016].

[35] Committee for the Workshops on Comp. Thinking, "Report of a workshop on the pedagogical aspects of computational thinking," National Academies Press, Washington, DC, 2011.

[36] E. Brewe, V. Sawtelle, L. Kramer, G. O'Brien, I. Rodriguez and P. Pamela, "Toward equity through participating in Modeling Instruction in introductory university physics," *Physical Review Special Topics - Physics Education Research,* vol. 6, no. 1, pp. 010106-1 - 010106-12, 2010.

[37] D. Rehn, E. Moore, N. Podolefsky and Finkelstein, N, "Tools for high-tech tool use: A framework and heuristics for using interactive simulations," *Journal of Teaching and Learning with Technology,* vol. 2, no. 1, pp. 31-55, 2013.