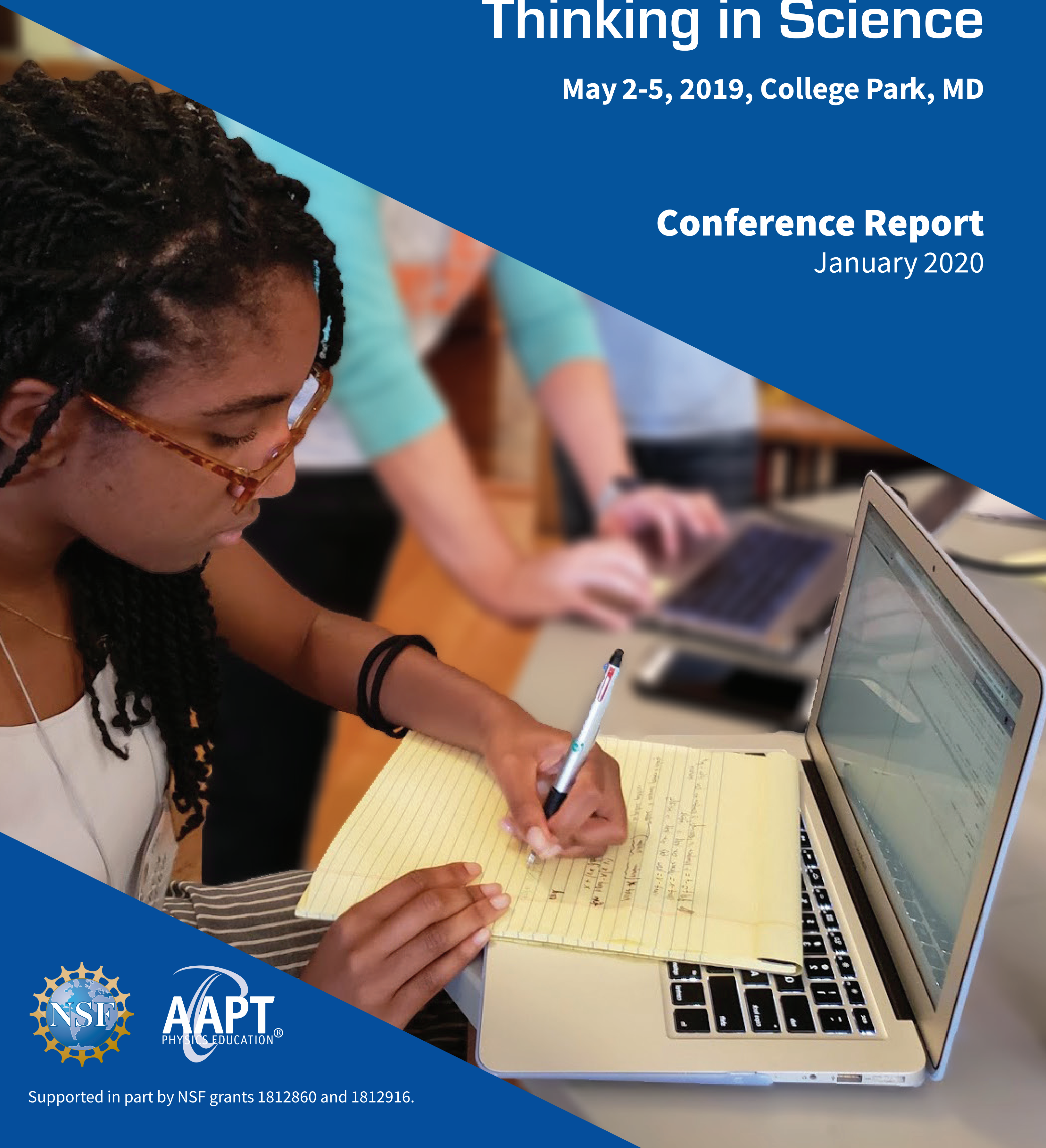
Advancing Interdisciplinary Integration of Computational Thinking in Science

May 2-5, 2019, College Park, MD



Conference Report

January 2020

Supported in part by NSF grants 1812860 and 1812916.

Conference Leadership Team

Sean Brophy
Purdue University

Marcos D. Caballero,** PI
Michigan State University

Kathi Fisler**
Brown University and Bootstrap

Michael Hicks
University of Maryland

Robert C. Hilborn,** Chair, PI
American Association of Physics Teachers

Colleen Megowan Romanowicz,** Co-PI
American Modeling Teachers Association

Kelly Roos
Bradley University

Rebecca Vieyra,** Co-PI
University of Maryland

**Report authors

Table of Contents

Executive Summary

This report was generated by participants in the May 2-5, 2019 *Advancing Interdisciplinary Integration of Computational Thinking in STEM* Conference. It provides a summary of the discussions at the conference and the recommendations drawn from those discussions. There was no attempt to arrive at a consensus, and, in fact, the report highlights the disparate ways practitioners and education researchers in different disciplines interpret and implement computational thinking. In short, the participants came to realize that there is no single construct that can be labeled "computational thinking," and that trying to build a consensus definition is not a productive way to foster computational skills and concepts for student learning. As a result, this report holistically refers to "computing" as those concepts and practices that include computational activities either with or without actual computers.

The sciences are rich in tools and methods. New tools provide novel views and generate fresh insights, while new methods yield results that were not previously attainable. Science often progresses with developments in tools and methods. Into this scientific world, computing has injected a flood of novel tools and methods. The computer, as a tool, enables rapid calculation and simulation at scales well beyond human capabilities. Computing offers the ability to create and manage data structures and algorithms that make previously intractable problems feasible to address and allows for entirely new kinds of problems to be solved. Computing has revolutionized all the sciences at the highest level. Yet this revolution has only just begun. As computing itself takes major leaps in sophistication (e.g., the advent of powerful new forms of artificial intelligence), the effects on the sciences will only grow.

This report focuses on *integrating* computing into K-12 science education. Integration means introducing enough computing into existing disciplinary education programs so that students can do work relevant to that discipline. An integrated approach is rich with promise and fraught with challenges—this report explores both of these aspects of integration, providing a roadmap for educators, investors, policymakers, and other stakeholders who are considering the role of integrated computing in the sciences.

Themes

The conference participants came to consensus on five key themes, which became the basis for our recommendations. The themes are the following:

- **Integration of computation must emphasize values native to the discipline in which computing is being integrated and demonstrate a clear alignment with existing standards.** As a result, integration may provide only a very limited exposure to computing, especially as small amounts of it can suffice to make a disciplinary point. In this case, the approach of computing in service of the core discipline might be at odds with using integration to fully satisfy K-12 computer science standards, which often emphasize software development.
- **Educational leaders need to recognize that relevant computing content differs across the sciences, ruling out a "one size fits all" notion of integrating computing in science.** For example, physics might emphasize the power of building and simulating physical scenarios, while biology might focus more on data analytics or the structure of biological data.
- **Diversity, Equity and Inclusion must be built into all efforts to integrate computation with science education.** Integration should take place in contexts that enable *all* students to avail themselves of these opportunities (e.g. through required courses), especially given the lack of diversity (racial, gender, socio-economic) in many advanced science and computing classes. Only 65% of K-12 students have access to a computer at school (National Center for Education Statistics, 2018); access is a particular problem in underserved rural and urban schools. Even when computing is offered, educators must ensure that course recruitment processes, teaching practices, and learning environments are fully inclusive and inviting. Likewise, professional development opportunities must be extended to teachers who are from traditionally minoritized backgrounds.

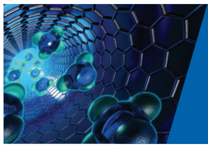
- **K-12 teachers need sustained professional development and support to learn and teach science while leveraging computing.** Integrating computing requires a significant mindset shift for the majority of science teachers, who themselves might not know how to address science problems with computers. Sustained funding for projects and changes to pre-service education are needed to move beyond the current limitations of short-burst in-service professional development.
- **Research is needed to understand and assess computational integration. There are relatively few theories of how computation impacts science learning. There are also very few useful assessments for charting progress.** Work from the science disciplines and computing needs to be wedded to concepts and theories from education, learning science, cognitive science, sociology, and other disciplines to achieve these ends.

Recommendations

Building on the above themes, conference participants developed a series of recommendations, organized into four cluster areas, for science education stakeholders . These areas are:

1. **Professional Development in Teaching and Learning**
2. **Student Learning and Curriculum**
3. **Assessment**
4. **Policy, Implementation, and Sustainability**

Lastly, while this report focuses on the integration of computation with science, we acknowledge that our recommendations also have implications for STEM (science, technology, engineering, and mathematics) and STEM-linked fields more broadly. The *Next Generation Science Standards* (NGSS Lead States, 2013) explicitly include standards for technology and engineering. The *NGSS* also have natural links to the *Common Core State Standards* (*NGSS*) (National Governors Association, 2010) for math and language arts (for example, reading in technical fields). While the conference attendees primarily represented the life and physical sciences, and our report discusses computing in relation to science, we urge readers to consider implications for computational integration across all STEM fields.



I. Overview

The May 2-5, 2019, *Advancing Interdisciplinary Integration of Computational Thinking in STEM* Conference (supported in part by NSF grants 1812860 and 1812916) brought together 40 computer science, physical science, Earth science, life science, and general STEM education faculty and high school teachers, including several whose scholarly work is in computational thinking education. The goal of the conference was to understand how computation is implemented through integration in participants' courses and to articulate a set of research questions whose answers could provide evidence for the effectiveness of various modes of implementation.

The conference was motivated by the significant shifts occurring in the practices of science and mathematics. Scientists increasingly leverage computing in a variety of ways. Whether to build models, create simulations, make predictions, detect patterns, or process large quantities of data, computation has become a standard tool for practicing scientists. The inclusion of mathematical and computational thinking as one of the eight science and engineering practices in the *Next Generation Science Standards* (NGSS Lead States, 2013) speaks to this transformation.

Simultaneously, 33 states have adopted or are in the process of adopting computing standards for K-12 (Code.org, 2018). Schools in several states are under mandate or have received strong suggestions to include computing in their curricula. Many schools and districts, however, lack personnel, resources, or time in students' schedules for stand-alone computing courses. Taken together, these forces suggest a hard look at integrating (some) computing into science instruction (Schanzer, Krishnamurthi and Fisler, 2019). Integration offers an approach that fortifies teachers' disciplinary expertise in science with the basics of computing to provide students with contextualized learning of an otherwise often abstract skill set. At the same time, computing provides a platform to both broaden and deepen students' conceptual understanding of the disciplinary content.

Integrating computing into science classes is no easy task. While some teachers are already using software tools to run simulations, integrated computing should ideally go deeper, preparing teachers and students to leverage computing to build, execute, and analyze their own models and data, in line with modern scientific practice. Achieving this vision raises considerable challenges around preparing teachers, getting teachers interested in learning and integrating computing, creating curricula, fostering equity, assessing impact, obtaining infrastructure, and promoting school, district, and state policies that create a climate for integrated computing. Furthermore, it raises questions about whether the costs (of many forms) are actually worth the effort, while not harming existing learning in the sciences.

This report summarizes conference discussions and presents a series of recommendations for future research and action. While this report is written primarily with an eye towards research that might be funded at the federal level, we also address policy considerations that impact the potential for integrating computing into science classes.

What do we mean by computational thinking?

Throughout this report, we describe the disparate ways that practitioners and education researchers in different disciplines interpret and implement *computational thinking*. The conference participants came to realize that computational thinking is not a single construct, and that there are many complementary ways of talking about the cluster of concepts, skills, and practices associated with computational thinking. For example, Wing (2006) emphasizes the broad aspects of computational thinking such as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science." As an example of an alternative approach, Weintrop and colleagues (2016) focus on *practices* associated with computational thinking. That approach, at least in principle, brings attention to behaviors that might be measured and assessed. The International Society for Technology Education and the Computer Science Teachers Association (ISTE and CSTA, 2011) have developed an "Operational Definition of Computational Thinking for K-12 Education," which, in part, has a well-defined emphasis on computers: "Formulating problems in a way that enables us to use a computer and other tools to help solve them," but also includes more general skills such as "confidence in dealing with complexity" and "tolerance for ambiguity."

The broad approaches to defining computational thinking have the danger of allowing educators to claim they are teaching computational thinking when they teach general problem-solving skills and foster attitudes such as "tolerance for ambiguity" with nary a computer in sight. However, we also want to avoid claiming that coding alone is the same as computational thinking. In this report, we use "computing" (and, occasionally, "computation" and "computational thinking") more holistically to refer to the broad range of concepts, skills, and practices that include computational activities that bring the affordances of computers to the field of science education.

Attending to Equity

The question of equity in integrated computing instruction is sufficiently important and pervasive that we highlight it up front, and then again within the individual sections on various aspects of this work. Historically, computing has had low engagement of different identity groups, such as people of color, people with disabilities, people who are emergent bilingual, people who are lesbian, gay, bisexual, transgender, intersex, queer, or Two Spirit [LGBTIQ2S], women, and people at the intersections of these groups. We must make direct efforts to address inequality and inaccessibility in order to make participation in computing reflect the society in which it is embedded. Addressing issues of equity at the intersection of computation and science education is particularly important, as these disciplines have documented, systemic inequities. Biased beliefs, structural inequalities, and state policies (Margolis, Estrella, Goode, Holme, & Nao, 2017) have led to the most heavily male-skewed AP exams being those in physics and computer science, with Black and Hispanic students having the lowest pass rates (Ericson, 2019). By working to integrate computation in science education without taking diversity, equity, and inclusion into account, we could inadvertently exacerbate those inequities.

For the purposes of this report, we take equity to mean that all students and teachers get the support they need to be successful. For this to happen, individuals and systems must work to interrupt inequitable practices, examine biases, and create inclusive environments, while discovering and cultivating the unique talents and interests that every student possesses (National Equity Project, 2017). These principles will lead to equitable outcomes in the context of learning, teaching, research, service and employment, including closing representation and participation gaps within our K-12 and university communities.

Equity requires an intentional commitment to strategic priorities, resources, respect, and civility, and ongoing action and assessment of progress towards achieving specified goals. In this report, we begin to articulate the necessary actions that the computational teaching and learning community needs to take. By asking a series of questions, we aim to bring the conversation about equity to the forefront of research, instructional design, pedagogy, resource management and utilization, and assessment work in the computational teaching and learning community. Key framing questions, which appear in various guises within the remaining sections, include:

- What are the barriers to equitable computing education?
- How must policy change to reduce and ultimately eliminate barriers for equitable implementation of computing?
- How do curricula and pedagogy need to change for equitable participation in computing?
- What does computing education look like for students with disabilities?
- What are the impacts on equity when computing is integrated with the sciences?

Conference Organization and Agenda

The conference agenda (Appendix A) provided attendees (Appendix B) with opportunities to think broadly about the scope of the integration of computation in the sciences, to showcase examples of initiatives that integrate computing into the sciences, and to review extant findings from the research community on integration. Small and whole group discussions by the attendees were embedded as a core feature in each day's activities, which allowed the participants to identify emergent themes and to prioritize goals for research, programming, and policy. At the end of the conference, the participants formed working groups to write guidelines and recommendations for these themes.

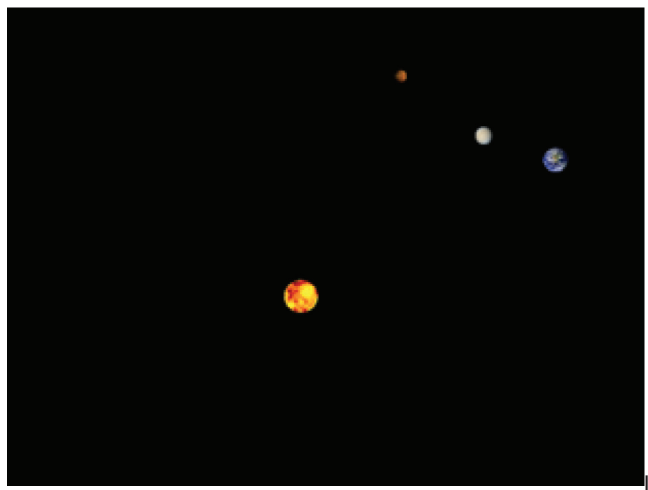
II. Illustrating Computing in the Science Classroom

While integration can look very different in different classrooms, illustrative scenarios can reveal the potential of bringing computing into disciplinary science courses. The following scenario is *one* example of how a conference attendee integrates computing into his physics class.

Lucas Walker is an experienced high school physics teacher who teaches juniors and seniors. He exudes confidence in his instructional ability, as well as his pedagogical choice to use the Modeling Instruction approach. Up until recently however, he struggled to find a way to teach Newton's Law of Universal Gravitation that was consistent with his pedagogical philosophy. This law, which can be expressed algebraically as $F_g = G\frac{m_1 + m_2}{r^2}$, cannot really be developed or tested empirically in a typical classroom laboratory investigation. In recent years, he had simply conveyed the law to his students through a reading.

But where did it come from? Newton derived it mathematically from his own laws of motion and from Kepler's laws of planetary motion. Kepler inferred his laws by painstakingly fitting the data on observed orbits of planets to possible calculated orbits until he hit upon a formula that made consistently correct predictions.

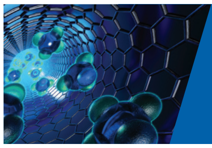
Walker knew that he couldn't conveniently replicate this process in his classroom, but he saw that the construction of a computational model could provide his students with a 21st century experience that paralleled Newton's and Kepler's approaches. His students already knew how to write functions to calculate "next-position" and "next-velocity" for an object in accelerated motion, so he set up a situation in which they would have to define the force function describing the gravitational interaction between the Sun and the planets.



The resulting simulation (Figure 1) was pre-programmed to include the Sun, Mercury, Venus, and Mars. The planets' motions were set to a default state in which they had an initial velocity moving upward on the screen, and in which gravitational interactions between the planets and the Sun were absent. Running the program with the default values caused the planets either to go off the screen or to fall into the Sun. Walker asked his students to predict and test variables' relationships to one another so they could find a function that would place the planets in stable elliptical orbits.

Walker commented about his instruction, "And what was really cool is that when I finally did it with students, it went exactly how I was hoping it would. I gave them the opportunity to brainstorm variables that might make a difference. They tried a bunch of different functions, and then finally as a full class, we were able to find a formula that worked. When we finally plugged it in on the big screen and we had the formula that made the planets' orbits stable, the class burst into a round of applause. It was this huge payoff ...treating gravity as a function, which they could then manipulate in their programming language and conceptually go through the same process that took Kepler years and years and see, via trial and error, that Newton's Law of Gravitation actually can produce the motions we observe in our solar system, and THAT's why we should believe it—not because some textbook says so."

This vignette illustrates the power of computing in science on multiple levels. Computing has the potential to promote the use of active pedagogies for otherwise hard-to-teach topics, to help students break down a complex phenomenon into its respective parts, and to efficiently perform multiple calculations and to display the results visually. Perhaps most importantly, computing can transform difficult concepts into exciting challenges.



III. Professional Development and Teacher Learning

Meaningful integration of computational thinking with science learning at the K-12 level will require a comprehensive approach to providing education, ongoing support, and resources for science teachers. Because teaching is a profession that is often built upon an "apprenticeship of observation" (Lortie, 2002), the way teachers teach is often strongly linked to how they observed their own teachers teach them. As a result, teaching approaches are often very resistant to change, despite major advancements in education research (Meltzer & Otero, 2015). For the vast majority of teachers currently teaching K-12 science, there were few, if any, computational thinking activities in the science courses they took in middle and high school, and even fewer in the science methods courses they took as a part of their teacher preparation. As such, teachers must first experience computational practices as learners in the disciplines they teach before they can design and deliver effective learning experiences for their students.

On the teacher-learning side, professional development must provide teachers with a solid foundation in both computing (i.e., designing a process or algorithm, implementing it on a computer, and assessing that it works) and modeling (i.e., defining and using function- and process-based computational models of physical phenomena or systems). Professional development must help teachers build self-efficacy, confidence, and interest in leveraging computing in science and math education. Low self-efficacy can hinder implementation of reform-based instruction (Grant & Hill, 2006). For example, when the *Framework for K-12 Science Education* (National Research Council, 2012) and subsequently the *Next Generation Science Standards* were released in 2012 and 2013, the majority of science teachers felt uncomfortable and ill-prepared to meet the requirement to teach engineering design (Haag & Megowan, 2015). Since most science teachers have had little experience with computation in their pre-service courses, we must work to equip them with the concepts, skills, practices and pedagogy needed to incorporate computational elements into their science courses. Attention to building teachers' confidence as well as competence will be critical in effecting change at the classroom level and facilitating the integration of computing in science.

On the classroom-implementation side, teachers will need materials that equip them with robust pedagogies to guide and to facilitate student thinking and learning when computation is integrated into their courses. Materials must be adaptable to different classroom settings and available infrastructure. The equitable distribution of classroom computing resources is not yet a reality in many underserved rural and urban schools. As mentioned in the Executive Summary, only 65% of K-12 students have access to a computer at school (National Center for Education Statistics, 2018). Limited computer access should not be an additional variable that further divides students into "successful" and "not successful" groups. Previous research demonstrates that teachers believe that a one-to-one computer-to-student ratio is necessary to incorporate modeling and technologies that facilitate computing (Gonczi, Maeng, & Bell, 2017). However, effective collaborative strategies could reduce the need for one piece of technology per student. Therefore, teachers need to engage in computing within shared computer contexts and consider how to integrate computation into their own classrooms when resources are less than ideal. Those supporting teacher knowledge and growth need to model and facilitate computing practices in various person-to-computer ratios and facilitate teacher reflection and planning to ensure all students develop computing skills irrespective of computer resources and infrastructure.

The professional development experiences described here will need to be developed for and deployed with both pre-service and in-service teachers. Teacher expertise and confidence will not develop quickly or cheaply. Sustained and ongoing professional development, with multiple activities over several years, will be required to create a transformational shift in how teachers envision science education that leverages computing. We offer the following recommendations:

- **Recommendation T1:** *Determine the scope of pre- and in-service science teacher and science classroom computational needs at all levels, and determine the quantity and duration of professional development necessary for teacher expertise to develop and to be maintained.* This will entail ongoing assessment as teachers, classroom resources, school administrations, students and standards evolve. In addition, as new strategies and tools find their way into classrooms, we must evaluate their impact and effectiveness.

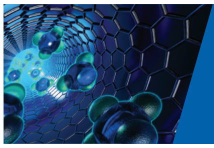
- **Recommendation T2:** *Identify educational leaders, including K-12 classroom teachers who can lead workshops for their peers and teacher educators, who, in partnership, can create sustained support systems that will enable teachers to develop both competence and confidence in integrated computing and science.* The integration of computing into science teaching will necessitate new skills and competencies that will in turn lead to pedagogical change. As we collaborate with teachers to identify effective practices and pedagogies, we will need to propagate these practices and pedagogies via professional development for in-service teachers and updated courses for pre-service teachers.

- **Recommendation T3:** *Articulate and leverage theoretical frameworks for integrated computing education to guide the development of classroom tools, software, content standards, assessments, and expectations for teacher preparation and ongoing teacher certification*. As we transition to integrated computing in science for K-12, new theoretical frameworks will emerge that illuminate factors affecting both teacher and student learning, their computational fluency, and the use of computational tools. As new affordances emerge and teachers' computing skills grow, they will uncover new approaches to deepen integration.

- **Recommendation T4:** *Understand whether and how teachers come to value computing as a component in science and mathematics education.* Teachers will not adopt materials that they do not value in their disciplinary and pedagogic contexts. As most teachers are themselves new to using computing in science, we expect teachers to evolve (positively and negatively!) in their appreciation and conceptions of integrated computing over time. Understanding these dynamics will be essential for designing effective learning materials for both teachers and students.

Equity Considerations

In preparing teachers to work with students, professional development must also help teachers design learning experiences that pay attention to equity. Students in teachers' classrooms come from a wide variety of backgrounds and identity groups including many from marginalized groups. Teachers will need to learn methods to best engage students who don't hold computing or science identities or who have low self-efficacy around computing or science. Specifically, teachers will need professional development that helps them to attend to the participation of students from marginalized groups in their classrooms, to reflect on their teaching practice that engages these students, and to develop activities that best support learning and engagement of these students. Professional development should provide materials or guidance for teachers in bringing cultural responsiveness (Ladson-Billings, 1994 & Gay, 2000) into integrated materials.



IV. Student Learning and Curriculum

How can we help students put content knowledge, practices, and thinking tools to use? Recent advances in establishing a research-based vision for science and mathematics through the *Framework for K-12 Science Education* and its associated *Next Generation Science Standards (NGSS),* and *Common Core State Standards* anchor efforts to support the development of student proficiencies in science fields.

However, setting standards does not ensure that fruitful learning will follow. The 21st century technology-enhanced learning environment is filled with tools and affordances that students must learn to use proficiently, creatively, and proactively to construct and apply the fundamental conceptual models in the various disciplines. While we have a large body of research on science learning with traditional laboratory materials and equipment, we still know very little about students' thinking and learning as it unfolds with the use of computational tools. At the very least, new tools for thinking and making sense of data call for curriculum resources that take into account students' developing computational literacy. With the introduction of this new competency, novel effects may emerge concerning student engagement, motivation, and identity in computationally enhanced classrooms.

When and in what sequence can and should various computing skills be learned? What skills are most useful in each of the respective scientific disciplines? What useful disciplinary knowledge can students learn (or learn more deeply and coherently) with computational support? What digital skill set does a student need for the long term? These questions and more are yet to be answered, and their answers will point the way to new learning experiences, new curricula and new opportunities for students. We offer the following recommendations:

- **Recommendation S1**: *Develop robust, developmentally appropriate models of computational integration that reflect the authentic nature of the use of computing in support of scientific inquiry.* Modern scientific work relies on a variety of computational concepts: systems decomposition and characterization for simulation, coding for model development, modeling data, and designing and debugging algorithms are several examples. The specific computational skills needed for different disciplines and tasks need to be articulated and expanded into learning progressions that make sense in the context of science courses. Existing standards for K-12 Computer Science (k12cs.org, 2012) have been defined for stand-alone computer science courses, but not for computing content integrated in other disciplinary contexts. The design of integrated learning progressions might deviate from the recommendations of stand-alone CS standards both in the order in which concepts are introduced and in the range of computing contexts included. These differences should be gathered, disseminated, examined and discussed in both the science and computing education communities.

- **Recommendation S2:** *Examine the development of students' identity, agency, positioning and motivation in relation to their engagement in computational tasks.* Computational skills and competencies co-evolve with technologies. With the development of learning tasks that engage students in design and give them greater control over the pathways they take to achieve desired outcomes, students' interests and views of themselves may shift. Studying student work on design projects, for example, could yield models of how students leverage computation to advance their own interests. These models can inform the development of curricular and computing resources, (e.g., languages, development environments, design activities) and instructional strategies that foster robust and flexible practices that enhance STEM identity and motivate an interest in developing STEM competencies.

- **Recommendation S3***:* *Look for ways in which equipping students with computers changes their approaches to problem-solving in the different science disciplines, and ways to monitor the evolution and development of these computationally supported approaches as students become more sophisticated users of computational tools.* Closed-form problems have been the standard in K-12 science education for centuries. Computing will empower students to tackle questions that were formerly considered "beyond the scope" of their course of study. No longer will they have to "ignore the effects of friction" in studying the motion of an object, or be content with modeling just one factor's effect on population or climate change. This enriched problem space will lead not only to new ways of asking questions and formulating approaches to answering them, but also to new notions of what the science curriculum for each discipline should be.

- **Recommendation S4:** *Identify and characterize the range of computation tools and affordances that are effective in science learning environments and characterize the optimal and equitable uses of each*. In astronomy, direct measurement of most celestial phenomena is impossible, but sophisticated computational tools are available that allow students to extract data from images. In chemistry, we do not directly see molecules, but simulations that encode the properties and behavior of molecules enable students to predict their behavior. In biology, populations of organisms are affected by a multiplicity of factors, but students can determine the relative effects of each of these factors by encoding the rules that govern each and running their computer model to see if the changes in population in the model match those that we observe in the real world. Every discipline is going to have a particular set of computational tools and approaches that best suit its learning goals. Identifying these and determining when and how they are best introduced and developed will provide useful support in developing curricular resources in each discipline.

Applications of information technology depend on the context and nature of the problem and the system of study. Some disciplines focus on remote or hard-to-define systems such as the global climate, an ecosystem, the integration of a biosystem with technology, or extraterrestrial life, while other professionals explore direct application of domain knowledge to optimize systems like the lifelines of our civil infrastructure or autonomous transportation systems. The choice of tools depends on whether we are seeking innovative changes to the system or just monitoring current performance to ensure the system is working as intended. In one case, the professional is looking for opportunities to refine and enhance the system; in other cases, the professional is performing well known, but complex, processes associated with maintaining the health of the system. Depending on the choice of objective, the type of computational tool employed and the level of user modification will fall somewhere along a continuum of computational practices and proficiencies. How can these manifold variations be organized in a framework? A related question: Are the problem-solving tools emergent from a specific implementation of computational thinking in any way generalizable?

Equity Considerations

Since some sciences and computing fields struggle to achieve diverse participation, student computational learning experiences must be designed with equity in mind. The Universal Design for Learning (UDL) (Rose and Meyer, 2006) framework can be a guide to ensure the accommodation of individual learning needs in computational teaching and learning environments. UDL pushes us to consider critically for whom our classrooms and instruction are designed and whom those environments and instruction might continue to marginalize. Informed by UDL's focus on reducing barriers to learning, this approach will require that we develop and implement:

1. Methods to engage students who do not hold computing/STEM identities or who have low computing/STEM self-efficacy, and
2. Practices that focus on equitable participation and assimilation of knowledge and understanding.

Throughout the entire developmental process, the following questions should be considered to determine how equity has been impacted. These questions should be continually referred back to during and after implementation to inform the revision process:

1. What are the factors that influence students' persistence in a science major or in their career choices?
2. What roles do self-efficacy, identity, interest, and anxiety play in persistence in a computationally integrated science program?
3. What does the integration of computation in science courses mean for retention in or attrition from STEM programs?
4. What are the implications for equity issues and career opportunities?

This work requires that research on issues of equity in computing education be conducted and communicated to the broader teaching and learning community. In carrying out this work, we must ensure this research is conducted with the participation of all students, including those from marginalized groups. It is critical that this work address attitudinal and affective aspects of computational learning as those aspects are central to participation and learning (National Academies of Sciences, Engineering, and Medicine, 2018). In particular, we should consider:

1. Who values computing education?
2. Who feels they can do computing?
3. What influences attitudes/self-efficacy of students (e.g., the school environment, the teacher, and school policies)?
4. What motivates adoption by teachers? What motivates participation by students?
5. What shapes a teacher's identity/attitudes/self-efficacy around computing?

The results of this research need to be communicated to all stakeholders including students, teachers, policy makers, families, and the communities and institutions in which these stakeholders are embedded.

V. Assessment

Ideally, integrating computational thinking into the sciences will improve educational outcomes in these disciplines, as well as in computing. In order to check for these hoped-for improvements, and to confirm that integration has not harmed science learning, there is an urgent need for reliable and valid assessments. Assessments will be needed for cognitive factors, affective factors, and career orientation. Cognitive assessments must consider both performance in the original integrated context and transfer of computational skills to additional science contexts (noting that transfer is difficult to achieve in practice). Beyond the conduct of assessments, research must consider how to report results and make them actionable for students, teachers, and curriculum developers. Consequently, a variety of robust assessment instruments, comprising both formative and summative tasks, must be developed in a generalizable fashion such that they can easily be adapted to different contexts and tasks in a variety of domains. With this in mind, we recommend the following areas of research and development:

- **Recommendation A1:** *Systematically identify discipline-specific learning outcomes for integrated computing programs that focus on the aspects of disciplinary learning made possible by computing that would not have been possible in the absence of computing and computational power.* Identification of these learning goals is best done by working groups that include disciplinary education researchers, computer science education researchers, and active classroom teachers of the discipline at the target levels.
- **Recommendation A2:** *Develop a national repository of classroom assessment tasks* organized by grade levels and STEM disciplines (e.g. life science, physical science) that includes common student challenges and suggested remediation and extension approaches. These assessment tasks need to be developed and made freely accessible to teachers and researchers, where teachers can easily contribute both new tasks and feedback on existing tasks. (There have been a variety of attempts to consolidate such resources in physics education, from the *Pretty Good Physics Site* (2018) to *PhysPort* (2019), but sustainability has been an issue for these sites and similar sites in other STEM disciplines. Despite the promise of sites such as the American Association for the Advancement of Science's *Science Assessment* website (2019), most of these sites do not allow teachers to submit new items.
- **Recommendation A3:** *Develop a national repository of tasks that measure transfer of understanding,* to which researchers can contribute assessment tasks designed to compare performance among programs and to which teachers can contribute (properly anonymized) data. For example, the *PhysPort Data Explorer* (2019) provides a repository of assessment tasks and a user interface for instructors to analyze results across levels of physics instruction. When possible, tasks should strive to be cast in a universal computer language (perhaps pseudo-code) that is not tied to a particular programming environment. However, many in the computing education community (Krishnamurthi and Fisler, 2019) recognize that truly computer-language-independent assessments are not feasible in general due to significant differences in how problems might be expressed across languages.
- **Recommendation A4:** *Undertake long-term and longitudinal studies* to understand more completely the effects of integrated computing in science on student attitudes and self-efficacy, course selection in secondary education, course/major/program selection in higher education, and impacts on broadly conceived student outcomes. These studies necessarily should follow students' career pathways over multiple years.

Equity Considerations

Assessments can implicitly encode assumptions about students, and those assumptions in turn may create inequitable learning environments. To accommodate a rich variety of students, assessments should feature approaches and design that are free(er) from bias, support multiple ways of knowing and demonstrating knowledge, and learning goals that align with students' backgrounds. Assessment procedures should give more weight to formative assessment, and include assessments that are appropriate for all learners. In addition, specific research studies should address these inequities and work to design assessments that allow all students to demonstrate their understanding.

VI. Policy, Implementation, and Sustainability

Educational initiatives are often accompanied by broad-swath policies that are difficult to implement and fail to account for the key actors in any education reform—the teachers. For example, in recent years, the state of Texas has mandated that computer science be taught in all high schools, despite the fact that in 2015, only 14 pre-service teachers completed certifications in computer science education (Magorrian, n.d.) and there is a persistent shortage of qualified teachers overall in Texas (WeTeachCS, 2018).

In considering the integration of computing into the sciences, it is important to be attentive to the real needs of teachers to provide students with a quality education. It is essential to consider that the computational preparation of science teachers can be lacking and that science teachers are often under-supported even in the core science disciplines that they teach, an issue that is particularly acute for the physical sciences. Another critical policy consideration is that the responsibility to integrate computing into other disciplines does not rest solely with the sciences, but should happen in tandem with mathematics.

The following recommendations do not propose "solutions" to the challenge of integrating computing into the sciences, but are meant to serve as guidelines for careful consideration by policymakers who can fund and support the development and dissemination of educational innovations with integrated computing in science.

- **Recommendation P1:** *Promote the research on and adoption of learning standards that foster the authentic integration of computing in science and math courses.* In recent years many states have approved K-12 computer science standards (Jacobson, 2018). Some of these standards are inspired by the *K-12 Computer Science Framework* (K12cs.org, 2016), which has explicit alignment with the *Next Generation Science Standards.* The *NGSS* lists mathematical and computational thinking as a core practice of science and engineering. However, the responsibility to align computational skills to existing disciplinary core practices should not rest with the natural sciences alone. The *Common Core* Mathematical Practices describe a number of skills associated with computational thinking, such as modeling with mathematics, using appropriate tools strategically, making use of structure, and expressing regularity with repeated reasoning. Given this coordination of concepts and practices, a state's adoption of the *NGSS* and *Common Core* themselves, or *NGSS*- or *Common Core*-aligned computer science standards can promote the integration of computing in science and mathematics courses. Compared to science education, there is less agreement that computational thinking should be an explicit learning objective of mathematics instruction. This lack of agreement appears in the 2006 opinion piece by Jeanette Wing that contains the aphorism "think computationally, not mathematically." Moving forward, computational thinking should be seen as a set of skills and competencies that all teachers, including mathematics teachers, should explicitly contribute to developing in students. Science and computer science teachers cannot be expected to shoulder the entire burden of integrating computational thinking into schools.
- **Recommendation P2:** *Encourage schools to offer science and math courses with integrated computing in addition to taking stand-alone computer science courses.* As of 2017, nine states mandated that pure computer science courses be permitted to substitute for required science courses for high school graduation (Education Commission of the States, 2017), and, in many other states, Code.org has successfully advocated for computer science to count in place of Algebra 2 as a "core" graduation requirement (2018). As states and prominent computer science educators (such as Mark Guzdial, 2014) publicly consider what kinds of courses can substitute for one another, it is important to acknowledge that the replacement of courses, such as physics, with computer science could further limit the access of young women and underrepresented groups to fundamental natural science and engineering (Orban, 2019). To acknowledge the value of the intellectual traditions that both science and mathematics bring to education, courses that replace (rather than augment) much of the natural science and math content with computer science should be discouraged. In states where the decision to allow computer science to count as core math or science requirement has already been made, those courses should integrate at least some math and science disciplinary content.
- **Recommendation P3:** *Prioritize educational programs that permit teachers to innovate and integrate computing into core sciences.* While Advanced Placement (AP) and International Baccalaureate (IB) science programs can helpfully promote rigor in their disciplinary fields, they might also present a barrier for explicitly integrating computing. In intense courses such as these, teachers are typically constrained by an externally imposed timeline to focus only on topics that they know will be on the exam. These exams are designed to mimic college courses that do not typically integrate computing. We encourage schools to weigh the benefits of the AP and IB programs against the tendency for these programs to implic-

itly or explicitly discourage the integration of computation into the curriculum. Further, we encourage the developers of AP and IB programs, as curricular and instructional leaders who carry significant influence in the community, to consider explicitly promoting the integration of computation into their course offerings. While some AP teachers do integrate computation—such as a sub-group of AP Physics teachers who use the *Matter & Interactions* (Chabay & Sherwood, 2015) text to teach calculus-based physics through coding with Python—explicitly including computing in core science and math courses could re-shape public perceptions about the field and how it should be taught. However, we do offer a serious caveat with respect to the assessment of computational skills in physics. As discussed in the section on *Assessment*, there is still much to be learned about assessing student knowledge of integrated computing in science. Therefore, it is premature to propose *immediate* changes either to existing coursework or end-of-year exams.

- **Recommendation P4:** *Support a community of K-12 educators as learners, teachers, content developers, and researchers through long-term, longitudinal funding opportunities.* In order to make a lasting impact on educational practices, we need professional development that supports K-12 teachers to integrate computing into their areas of science expertise. The typical three-year funding cycle of many research and development projects encourages the initial development of programs, but leaves classroom teachers at the cusp of mastery without the support necessary to make gains permanent and spread best practices deeper into districts and across disciplines. The education community needs to be provided with capacity-building mechanisms for teacher professional development (pre-service and in-service) to help teachers build the necessary skills and community to begin and sustain implementation.

For example, the isolation of innovative teachers who are already integrating computing into their science teaching is exacerbated by long-standing isolation of disciplinary science teachers in general. Jennifer Broekman, a conference attendee and high school teacher, who is deeply involved with an NSF-funded computational modeling in physics grant, shared that the only workshop being offered in her state that claims to integrate computing into physics is spreadsheet-based, which is a very specific and limited form of computation. She commented that she is the only physics teacher in several nearby districts who is including coding in non-AP physics. Although she is among the few teachers who are truly innovating in this area, the expectation remains that educational technology or vocational teachers are responsible for teaching computation. When a local university math and science teacher educator directed inquiries about integrating computational thinking into science classes to Jennifer's district, he contacted the school's educational technology teacher. Despite the shared value and interest in integrated computing, the community of cutting-edge educators is largely fragmented.

In some ways, the situation surrounding integrated computing education is a microcosm of the struggle that teacher networks face in the United States where there may be National Science Foundation, U.S. Department of Education, or state funding for a few years followed by a dry spell. Without sustained funding, many of the responsibilities for these networks fall to a small number of teachers who become overextended. No one wins if experienced teachers are expected to fill too many roles, or if teacher networks disintegrate and teachers are isolated from each other, unable to share what is working in their classrooms. These dynamics will affect any effort to improve instruction and they affect efforts to integrate computing into the sciences.

- **Recommendation P5:** *Integrate computation into undergraduate core science courses and require all pre-service STEM teachers to be trained in computational thinking relevant to their disciplines.* Undergraduate and graduate education for pre-service science teachers must integrate computation into classes, and should require that pre-service teachers integrate developmentally appropriate computational thinking into their teaching. Encouraging and funding more collaboration between resource-sharing communities like PICUP (2019) and teacher education-focused initiatives like PhysTEC (2019) would help to meet this need. Fields outside of physics, such as chemistry or the life sciences, could be encouraged to emulate the kind of computational resource sharing that PICUP facilitates.

Equity Considerations

Policies play a significant role in equity because they influence the extent to which inequalities will persist or be diminished at institutional levels. As we work to integrate computing into science teaching and learning environments, we must develop policies that ensure an equitable implementation of computing. Reflecting the equity considerations from all perspectives (professional development and teacher learning, student learning and curriculum, and assessment), policies must ensure equity of opportunity and distribution of resources across systems.

VII. Conclusion

As science becomes increasingly dependent upon computing, it is important that we attend to how it transforms the way we support teachers and students. As our vision for science instruction evolves to include computing, it is likewise important that we attend to how we manage the educational systems. All our efforts to promote integrated computing in the sciences must address equity at the core. We are reliant upon the work of educational stakeholders, including practitioners, researchers, public policymakers, and leaders in business and industry, to achieve the vision for quality and equitable integrated computational education for all.

References

American Association for the Advancement of Sciences. (2019). Science Assessment. Retrieved from http://assessment.aaas.org/topics.

Code.org. (2018). State of Computer Science Education: Policy and Implementation. Retrieved from https://code.org/files/2018_state_of_cs.pdf.

Chabay, R. & Sherwood, B. (2015). *Matter & Interactions,* 4th Ed. New York, NY: Wiley.

Education Commission of the States. (2017). State of the States Landscape Report: State-Level Policies Supporting Equitable K-12 Computer Science Education. Retrieved from https://www.ecs.org/ec-content/uploads/MassCAN-Full-Report-v10.pdf.

Ericson, B. (2019). *AP CS Report for 2018*. Retrieved from https://cs4all.home.blog/2019/02/28/ap-cs-report-for-2018/.

Gay, G. (2000). *Culturally-Responsive Teaching: Theory, Practice & Research.* New York: Teachers College Press.

Gonczi, A. L., Maeng, J. L., & Bell, R. L. (2017). "Elementary teachers' simulation adoption and inquiry-based use following professional development," *J. of Technology and Teach. Edu.*, **25** (2), 5-34.

Grant, M. M. & Hill, J. R. (2006). Weighing the rewards with the risks? Implementing student-centered pedagogy within high-stakes testing, in R. Lambert & C. McCarthy (Eds.), *Understanding Teacher Stress in the Age of Accountability*, 19-4, Greenwich, CT: Information Age Publishing.

Guzdial, M. (2014). *Why Counting CS as Science or Math Is Not Considered Harmful.* Retrieved from https://cacm.acm.org/blogs/blog-cacm/175498-why-counting-cs-as-science-or-math-is-not-considered-harmful/fulltext.

Haag, S. & Megowan, C. (2015). "Next generation science standards: A national mixed-methods study on teacher readiness," *School Science and Mathematics,* **115** (8), 416-426.

ISTE and CSTA, (2011). "Computational Thinking Operational Definition for K-12 Education. https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf.

Jacobson, L. (2018). *Report: 44 States Have Implemented at Least One K-12 Computer Science Policy*. Retrieved from https://www.educationdive.com/news/report-44-states-have-implemented-at-least-one-k-12-computer-science-polic/533226/.

K12cs.org. (2016). *K-12 Computer Science Framework*. Retrieved from https://k12cs.org/.

Krishnamurthi, S. & Fisler, K. (2019) *Programming Paradigms and Beyond. Chapter 13 in The Cambridge Handbook of Computing Education Research*, pages 377-413. Eds: Sally A. Fincher and Anthony V. Robins. Cambridge: Cambridge University Press.

Ladson-Billings, G. (1994). *The Dreamkeepers.* San Francisco: Jossey-Bass Publishing Co.

Lortie, D. (2002). *Schoolteacher*. Chicago: University of Chicago Press.

Magorrian, D. (n.d.) Expanding Computer Science Education in Texas. Retrieved from https://txedmag.com/expanding-computer-science-education-in-texas/.

Margolis, J., Estrella, E., Goode, G., Holme, J. J., & Nao, K. (2017). *Stuck in the shallow end: Race, education, and computing.* Updated Edition. Cambridge: MIT Press.

Meltzer, D. & Otero, V. (2015). "A brief history of physics education in the U.S.," *Am. J. of Phys.* **83**(5), 447-458.

National Academies of Sciences, Engineering, and Medicine. (2018). *How people learn II: Learners, contexts, and cultures*. Washington, DC: National Academies Press.

National Center for Education Statistics. (2018). *Student Access to Digital Learning Resources Outside of the Classroom*. NCES 2017-098. Retrieved from https://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2017098.

National Equity Project. (2017). Intro to Equity Webinar [Video File]. Retrieved from https://www.youtube.com/watch?v=mIWOtQzsoZg&feature=youtu.be.

National Governors Association Center for Best Practices, Council of Chief State School Officers. (2010). *Common Core State Standards.* Retrieved from http://www.corestandards.org/.

National Research Council (2012), *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. Retrieved from https://www.nap.edu/catalog/13165/a-framework-for-k-12-science-education-practices-crosscutting-concepts.

NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States.* Washington, DC: The National Academies Press.

Orban, C. (2019). Computer Science Now Counts as a Math Credit in Most States – Is This a Good Idea? Retrieved from https://theconversation.com/computer-science-now-counts-as-math-credit-in-most-states-is-this-a-good-idea-123424.

PhysPort (2019) Retrieved from https://www.physport.org.

PhysPort Data Explorer. (2019). Retrieved from https://www.physport.org/DataExplorer/Preview.cfm.

PhysTEC: Physics Teacher Education Coalition. (2019). PhysTEC Home. Retrieved from https://www.phystec.org/.

PICUP: Partnership for Integration of Computation into Undergraduate Physics. (2019). Welcome to PICUP! Retrieved from https://www.compadre.org/picup/.

Pretty Good Physics Site. (2018). *Welcome.* Retrieved from https://sites.google.com/prettygoodphysics.org/prettygoodphysics/home?authuser=0/.

Rose, D. H. & Meyer, A. eds. (2006) *A Practical Reader in Universal Design for Learning*. Cambridge: Harvard Education Press.

Schanzer, E., Krishnamurthi, S. & Fisler, K. (2019) What does it mean for a computing curriculum to succeed? *Communications of the ACM* **62**, 5 (April 2019), 30-32. DOI: https://doi.org/10.1145/3319081.

WeTeachCS. (26 July 2018). Computer Science Included in 2018-2019 Teacher Shortage Areas and Loan Forgiveness Programs. Retrieved from https://sites.utexas.edu/weteachcs/computer-science-included-in-2018-2019-teacher-shortage-areas-and-loan-forgiveness-programs/.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016) "Defining Computational Thinking for Mathematics and Science Classrooms," *J. Sci. Educ. Technol.* **25**, 127-147.

Wing, J. (March 2006). "Computational thinking," *Communications of the ACM* **49**(3), 33–35.

Appendix A

Advancing the Integration of Interdisciplinary Computational Thinking in the Physical and Life Sciences

May 2-5, 2019

Pre-Conference Work	
Short readings about Integrating Computational Thinking and STEM education, with directed questions to think about.	
Thursday, May 2 at Cambria Hotel, College Park, MD	
Late afternoon arrival	
4:00 – 4:30	***Introductions, Overview, and Purpose, Ice Breaker*** Goals and objectives overview
4:30 – 5:30	***Plenary # 1: David Weintrop, University of Maryland*** "Computational Thinking Taxonomies: An Update"
5:30 – 6:30	Dinner at hotel
6:30 – 7:30	***Table Groups: Theme – Integrating CT and STEM Education*** Gather information about how participants think about integrating CT. What do they think requires further investigation? How do lists of CT "practices" and research studies influence the implementation of the integration of CT and STEM education?
Friday, May 3 at American Center for Physics	
	Breakfast at hotel
8:00 a.m.	Bus leaves for ACP
8:30-9:00	***Leaders report out summary from previous night's group work***
9:00-10:00	***Plenary #2: Midge Cozzens, Rutgers University and Kristi Adams, Cottey College*** "Computational Thinking in High School Classrooms - Surprise Findings from an Initial On-line Professional Development Course"
10:00-10:30	Break

10:30-11:15	***Case Study Session #1:*** Shriram Krishnamurthi, Brown University
11:15-12:00	***Discussion of case studies and plenary in light of the Integrating CT and STEM Education theme.*** Guiding questions: – What do we already know about CT integration? – What does integrating computational thinking and practices add to STEM teaching and learning? – What does STEM add to learning about computation?
12:00-1:00	Lunch
1:00-2:30	***Case Studies Session #2*** Midori Kitagawa, University of Texas at Dallas Ruth Chabay, University of North Texas CMBF-B (Rebecca Vieyra, University of Maryland; Colleen Megowan-Romanowicz, American Modeling Teachers Association; Kathi Fisler, Brown University)
2:30-3:00	Break
3:00-4:00	***Plenary #3: Shuchi Grover, Stanford University*** "Designing for Learning & Transfer in STEM+CT Integration: Promising Ideas, Fertile Mappings, & Lingering Tensions"
4:00-5:00	***Case Study Session #3:*** Tor Odden, University of Oslo Steven Temple, San Rafael High School
5:00-5:45	***Small Group Discussions: Framing and re-framing research questions*** What metrics should be used to assess the quality of integrated CT materials and instruction? What research questions must be answered to help develop appropriate metrics?
5:45	Bus to hotel
6:45-7:45	Dinner
7:45–8:45	Poster Session (with dessert and refreshments)
Saturday, May 4 at American Center for Physics	
	Breakfast at hotel
8:00	Bus to ACP
8:30-8:45	***Organizers report out. Introduce today's work.***
8:45-9:30	***Groups from yesterday report out: CT Integration themes, research gap analysis***

9:30-10:30	***Plenary # 4: Satabdi Basu, SRI International*** "Designing Assessments That Integrate Concepts and Practices in Science and Computational Thinking: A Principled Approach Using Evidence-Centered Design"
10:30-11:00	Break
11:00-12:00	***Case Study Session #4:*** Devin Silvia, Michigan State University Chris Orban, Ohio State University and Richelle Teeling-Smith, University of Mount Union
12:00-1:00	Working Lunch
1:00-2:00	***Breakout groups. Guiding questions:*** What needs to happen to move from theory to practice with CT integration? Who are the key stakeholders? What are their needs?
2:00-3:00	***Working Groups: Establish groups. Groups refine goals and agenda. Begin work.***
3:00-3:30	Break
3:30-4:15	***Working groups continue tasks***
4:15-5:15	***Report outs from Working Groups, discussion***
Sunday, May 5 at hotel	
8:30-9:15	***Guiding Questions:*** Where are we now? What needs to be done this morning? More share out from Working Groups
9:15-10:15	***Working Groups continue with their tasks***
10:15-10:30	Break, hotel check out
10:30 – 11:30	***Next Steps: overview, writing assignments, writing schedule, writing teams make plans and start working.*** Can we articulate 3-5 major recommendations (or goals, outcomes, etc.) for further work?
11:30-11:45	***Closing, post-workshop survey, etc.***

Appendix B

Conference Participants

Kristi Adams**
Cottey College

Satabdi Basu**
SRI International

Ernest Behringer
Eastern Michigan University

Jennifer Broekman
Emerson Public Schools

Sean Brophy
Purdue University

Danny Caballero
Michigan State University

Ruth Chabay
University of North Texas

Hunter Close
Texas State University

Margaret Cozzens**
Rutgers University

Thomas Finzell
University of Michigan

Kathi Fisler
Brown University and Bootstrap

Michael Gallagher
Oakland Schools

Melissa Girmscheid
Peoria Unified School District

Amanda Gonczi
Michigan Technological University

Shuchi Grover**
Stanford University

Roxana Hadad
UCLA

Lisa Hardy
The Concord Consortium

Brandon Helding
University of Nebraska-Lincoln

Michael Hicks
University of Maryland

Robert Hilborn
AAPT

Paul Irving
Michigan State University

Midori Kitagawa
University of Texas at Dallas

Shriram Krishnamurthi
Brown University and Bootstrap

Joyce Mayln-Smith
Education Development Center

Colleen Megowan-Romanowicz
American Modeling Teachers Association

Tony Musumba
Bismarck State College

Tor Ole Odden
University of Oslo

Chris Orban
Ohio State U. / STEMcoding

Kelly Roos
Bradley University

Joshua Samani
UCLA

Bruce Sherwood
University of North Texas

Devin Silvia
Michigan State University

Richelle Teeling-Smith
The University of Mount Union

Steven Temple
San Rafael High School

Israel Touitou
Michigan State University

Rebecca Vieyra
University of Maryland

Lucas Walker
Weston Public Schools

Charles Wallace
Michigan Technological University

David Weintrop**
University of Maryland

Aman Yadav
Michigan State University

** Plenary Speaker

Advancing Interdisciplinary Integration of Computational Thinking in Science

May 2–5, 2019, College Park, MD

Conference Report
January 2020

Published by:

The American Association of Physics Teachers
1 Physics Ellipse
College Park, MD 20740
www.aapt.org

ISBN: 978-1-931024-35-8