Computational Thinking Integration in Elementary Teachers' Science Lesson Plans

Merijke Coenraad, Lautaro Cabrera, Heather Killen, Dr. Jan Plane, and Dr. Diane Jass Ketelhut,
University of Maryland - College Park

Corresponding Author: Merijke Coenraad, *mcoenraa@umd.edu*

Abstract

Due to the increasingly computational nature of professions, computational thinking (CT) is of growing importance to authentic science learning and the education of future scientists. To meet this need, CT integration is expanding within classrooms. We provided professional development (PD) for pre-service and in-service teachers focused on integrating CT into elementary science. At the PD culmination, 36 teachers wrote and enacted 22 unique CT-integrated science lessons, individually or with teaching partners. Waterman et al. (2020) suggested three levels of integrating CT within lesson plans: *exist*, labeling already present CT; *enhance*, adding CT components; and *extend*, adding activities supporting science learning with CT. Using this framework, we examined these lesson plans, their alignment to CT practices, and the level of CT integration. Our results indicated 83.3% of teachers successfully integrated CT within their lessons, focusing on Using Data, Computational Simulations, and Programming practices. Further, we found the level of integration differed by CT practice. Data practices generally led to *exist* level integration, Computational Simulation practices to *enhance* level integration, and Programming practices tended to *extend* science lessons or exhibit the science topic with Programming. Our data demonstrated teachers can write CT-integrated lesson plans, but all levels of integration are not equal opportunities for authentic scientific learning. As the field seeks to offer equitable and quality CT experiences for all students integrated within disciplinary subjects, we must understand the level of CT integration and consider how different levels of integration could affect opportunities for students.

Introduction

As society becomes increasingly computational, computational thinking (CT) instruction has taken on a growing role in schools (Bocconi et al., 2016; National Research Council, 2010). Beyond focusing on CT within computer science (CS) courses, researchers and educational standards encourage the integration of CT into disciplinary subjects (Lee et al., 2020), particularly science (NGSS Lead States, 2013). This shift towards CT integration has two goals: to provide opportunities for all students to access computing opportunities—a first step to broadening participation in computing courses—and to enhance science learning by making content more authentic to modern professional science. However, to provide CT learning opportunities that meet these two goals, it is imperative we equip teachers with the necessary knowledge and skills to integrate CT in ways that are both authentic to computing and prepare students to engage in computing as a way to learn science. It is important to prepare teachers for this task at the elementary level, where children are beginning to explore their academic and vocational identities and are impacted by positive experiences in science and computing (Tran, 2019).

Prior research has focused on preparing both pre-service and in-service teachers to integrate CT. Teacher educators have integrated CT learning into pre-service technology courses (e.g., Chang & Peterson, 2018; Mouza et al., 2017) and science methods courses (e.g., Jaipal-Jamani & Angeli, 2017; McGinnis et al., 2020). Further, researchers have encouraged focusing on both technology and disciplinary teaching (Blikstein, 2018; Yadav et al., 2017). Studies have built pre-service and in-service teacher knowledge using CT tools such as robotics (e.g., Jaipal-Jamani & Angeli, 2017), block-based programming environments (e.g., Bean et al., 2015; Bort & Brylow, 2013; Dodero et al., 2017), and simulations (e.g., Ahamed et al., 2010).

Taken together, this research has had mixed results. While both pre-service and in-service teachers showed an improved understanding of CT following professional development (PD) (Jaipal-Jamani & Angeli, 2017; Curzon et al., 2014; Yadav et al., 2014) and increased self-efficacy and attitudes about the importance of CT (Bower et al., 2017; Simmonds et al., 2019), some still had misconceptions about CT (Chang & Peterson, 2018; Lamprou & Repenning, 2018) and had difficulty writing lesson plans integrating

CT concepts and tools into disciplinary contexts (Bort & Brylow, 2013; Mouza et al., 2017). Therefore, it is important to investigate how best to support both pre-service and in-service teachers in enacting their knowledge of CT when designing integrated lessons and implementing them in their classrooms.

Our work examined how pre-service and in-service elementary teachers (herein referred to as "teachers") learned to integrate CT into elementary science teaching through PD. We focused on supporting teachers to write and enact CT-integrated lesson plans. To support our instruction, we developed the Framework for Teachers' Integration of Computational Thinking into Elementary Science specifically focused on supporting elementary teachers within a CT for science perspective (Ketelhut et al., 2019). Based on Weintrop et al. (2016) and informed by our prior work with teachers, the framework was designed with three main considerations: eliminating computer science jargon teachers found inaccessible, selecting practices elementary-aged children could engage in, and differentiating CT from scientific inquiry—a distinction we found was blurry for teachers. The framework divides CT into four sets of practices: Using Data, Programming, Computational Simulations, and Systems Thinking from a CT Perspective (Figure 1; for detailed definitions of each practice, see Cabrera et al., 2021). We introduced the framework to teachers early in their PD and used it throughout the program as a definition of CT and a set of concrete practices students should engage in during science learning. Within this paper, we also used this framework when analyzing teachers' enactments of CT in their classrooms.

Within the PD, we focused on supporting pre-service and in-service teachers to write and enact CT-integrated lesson plans within their elementary science classes. In this paper, we will present the lesson plans teachers developed and examine the CT practices integrated and the level of CT integration achieved according to a framework designed by Waterman et al. (2020). We aim to determine the level of CT integration in the lessons and identify patterns of integration to answer the research question: *To what extent do pre-service and in-service teachers integrate computational thinking into elementary science lesson plans?*

METHODS

We designed and implemented the *Science Teaching Computational Thinking Inquiry Group* (STIG^{CT}) to collaborate with teachers around integrating CT in elementary science lessons within a community of practice (Coenraad et al., 2021; PD guide and activities at https://education.umd.edu/stigct). STIG^{CT} was a semester-long PD for both pre-service and in-service teachers developed through Design-Based Research (Brown, 1992; Barab, 2006). Teachers and researcher facilitators met for four 165-minute sessions (February - May 2019; 11 hours total). Prior to participating in STIG^{CT}, all teachers were introduced to CT through their pre-service science methods course or a workshop for in-service teachers. The course and the workshop were designed to cover the same information to ensure that both groups received equal grounding in CT. This included an introduction to CT and the Next Generation Science Standards, presentation of our CT framework, and completing plugged and unplugged CT activities.

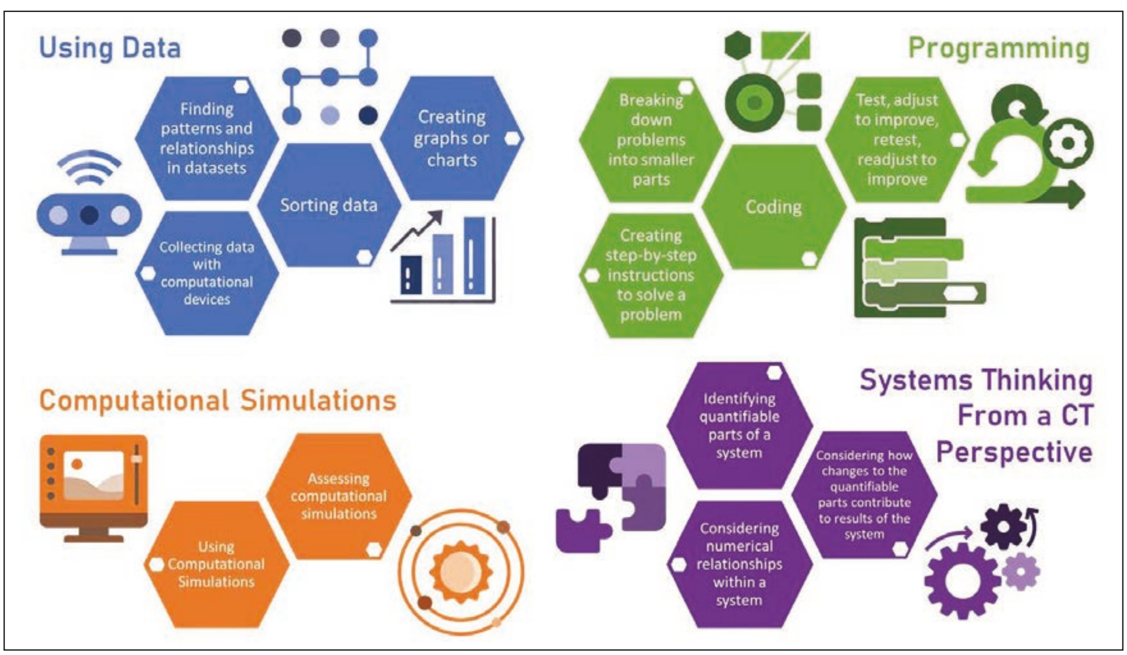

Figure 1. Framework for Teachers' Integration of Computational Thinking into Elementary Science

Each STIG^CT session focused on one of the four CT practices and included three sections: presentation of the CT practice, CT-integrated science activities from a student lens, and development of a lesson seed (the beginning of a lesson plan) with grade-similar peers and a facilitator. Each teacher selected one lesson seed and developed it into a full lesson plan they taught to their class. In the final session, teachers shared and reflected upon their lesson plan and teaching experience.

In total, 36 teachers participated in STIG^CT and submitted a lesson plan (20 pre-service, 16 in-service). They taught in elementary schools in the Mid-Atlantic region of the United States. Because some participating pre-service and in-service teachers worked together as a mentor/mentee pair or on a grade-level team at the same school, the teachers developed 22 unique lesson plans. In this paper, we analyze these lesson plans using Waterman et al.'s framework (2020) to determine the level of CT integration.

To categorize the level of CT integration within lessons, we modified Waterman et al's. (2020) three-part framework: *exist*, *enhance*, *extend* (Figure 2). In their framework, lessons are considered to be at the exist level if the "CT concepts, skills, and practices already exist in the lesson and can simply be called out or elaborated upon" (Waterman et al., 2020, p. 54). As seen in Figure 2, this is an instance of CT within a science lesson, but additional science learning is not supported by the CT. This level identifies ways CT is already in the curriculum and can act as a base for deeper integration. In their second level, *enhance*, CT is integrated based on the "creation of additional tasks or lessons to *enhance* the disciplinary concept and provide clear connection to computing concepts that are present" (Waterman et al., 2020, p. 55). In these lessons, students go beyond what is already in the curriculum, utilizing CT skills in service of their disciplinary learning. In Figure 2, this is represented by multiple instances of CT expanding the science lesson. In their final level, *extend*, teachers add CT activities, typically programming (Waterman et al., 2020). In our interpretation, we looked for teachers using CT to promote science learning through computational activities, *extending* students' learning of a disciplinary concept. As shown in Figure 2, the science lesson is expanded by the CT focus within the lesson. In addition to the levels of integration presented by Waterman et al. (2020), we included a fourth category, *exhibit*, which was identified inductively during our coding process. Lessons in this category used a CT activity, typically programming, to exhibit science knowledge students gained through other means. For example, creating a Scratch animation about an animal in its habitat based on book or online research. This can be seen in Figure 2 where CT and science are both present, but do not overlap.

During our analysis we first identified the CT practices integrated into each lesson plan. Two researchers read 20% of the data and reached 85.7% interrater reliability. The two researchers then discussed all disagreements to reach 100% agreement. One researcher then coded the remaining lesson plans. In a second round of coding, we used Waterman et al.'s (2020) framework to label each lesson plan as *exist*, *enhance*, or *extend*. Two researchers coded a subset of the lessons, discussed discrepancies in the coding, and completed the coding after reaching agreement. Following this coding, a portion of the lesson plans were identified as not aligning to any of Waterman et al.'s categories. We therefore added a fourth category for these lessons (exhibit) and re-coded the lesson plans. The initial agreement between the researchers was 90.9% (20 of 22 lessons), which was elevated to 100% after discussing discrepancies.

RESULTS

Overall, teachers were able to effectively incorporate CT into an elementary science lesson plan following their participation in STIG^CT. Thirty of the 36 teachers (83.33%) submitted a lesson plan containing at least one CT practice (16 of the 22 unique lesson plans; 73.73%). Within the 22 unique lesson plans, researchers identified the use of three of the four CT practices (Figure 3): Using Data (9 lessons; 40.91%), Computational Simulations (8 lessons; 36.36%), and Programming (7 lessons; 31.82%). No lesson plans contained Systems Thinking from a CT Perspective. Six lesson plans (27.27%) included no CT, despite teachers self-identifying practices in the lesson. Some lesson plans contained practices from multiple categories of CT practices (Figure 3). This overlap was most common

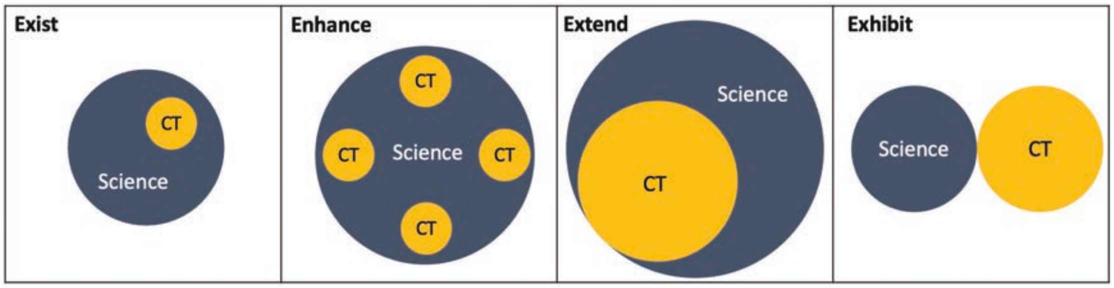

Figure 2. Levels of CT Integration

between Using Data and Computational Simulations. In these lessons, students typically collected data using a computational simulation and analyzed that data for patterns and trends to make conclusions (see Table 1 Example B below).

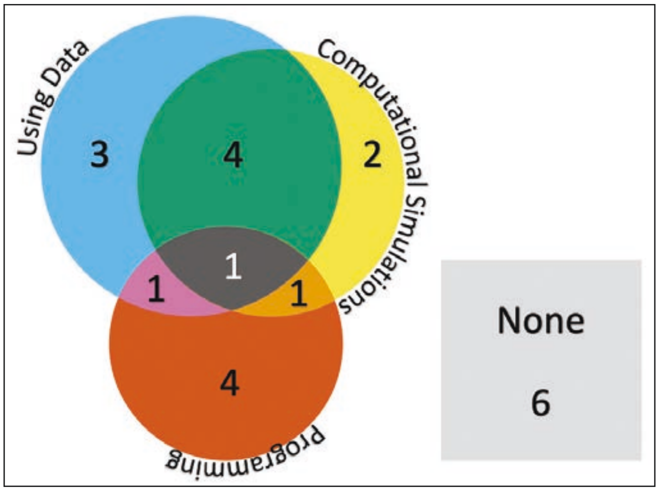

Figure 3. CT practices identified within teacher lesson plans

While teachers who participated in STIG[CT] were able to integrate CT practices within their lesson plans generally, prior research has demonstrated not all CT integration within lesson plans provides students with equal opportunities to deeply engage with CT (Bort & Brylow, 2013; Mouza et al., 2017). Of the 16 unique teacher lesson plans containing at least one CT practice, three lessons (18.75%) integrated CT at an *exist* level, identifying CT already present within a typical science lesson plan. The greatest number of lessons, eight (50%), integrated CT at an *enhance* level, using CT to support science learning by adding CT experiences with computing tools or practices (Figure 4). Two lessons (12.5%) *extended* science learning through the integration of CT tools and practices. Finally, three lessons (18.75%) integrated science and CT on an *exhibit* level, using a relevant science concept as the topic exhibited through the CT learning experience, but doing so in a way that does not explicitly increase science learning and could be replaced with a different disciplinary topic without changing the activity.

We also found patterns between the CT practices within lessons and the level of CT integration (Figure 4). Lessons identified to include *only* Using Data practices were all at an *exist* level of integration (3 of 3 lessons; Table 1 Example A). Further, those that included Computational Simulation practices integrated CT at the *extend* level (8 of 8 lessons; Table 1 Example B). Lessons that integrated Programming practices tended to reach the *extend* (2 of 2 lessons; Table 1 Example C) or *exhibit* (3 of 3 lessons; Table 1 Example D) level of integration. These trends point to a relationship between the CT practices enacted within a lesson and the level of integration the lesson reached.

Discussion

Overall, teachers successfully integrated CT into their elementary science lesson plans across most CT practices. But integrations varied in level of integration and coverage of CT practices. Our results show that no teachers integrated Systems Thinking from a CT Perspective into their lesson plans. This raises questions about whether systems thinking around quantitative relationships is developmentally appropriate for students at the elementary level and whether further support is needed for teachers to feel confident engaging their students in discussions of systems thinking. It is common for elementary classes to examine systems such as the food web or the water

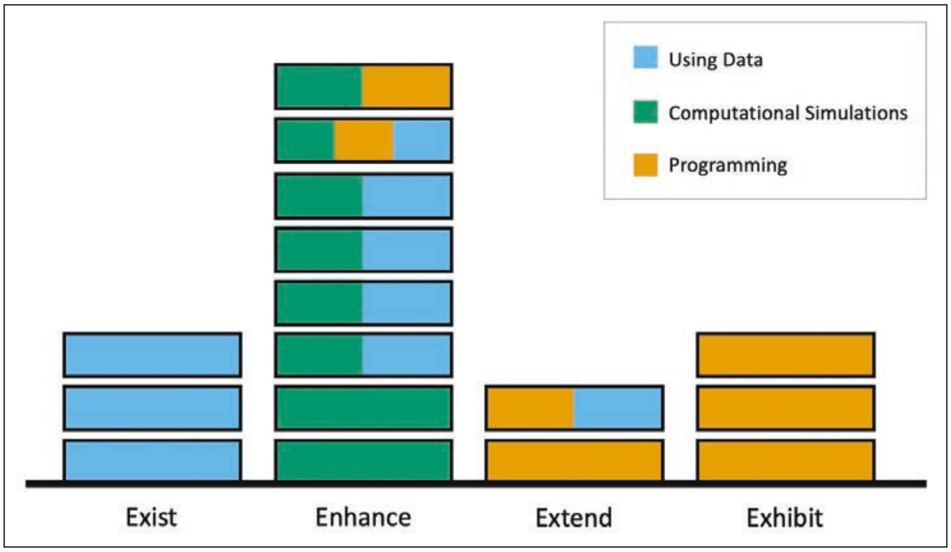

Figure 4. CT Integration levels in elementary science lesson plans

Table 1. Example lesson plans at each level of integration

Example	Level of Integration	Grade and Topic	CT Practices Integrated	Lesson Activities
A	Exist	1st grade Growing Lima Beans	**Using Data:** • Creating graphs or charts • Finding patterns and relationships in datasets	Students dissect a lima bean while making qualitative observations. Then, students plant two lima beans and place one in a lit environment (i.e., the window sill) and one in a dark environment (i.e., a dark room). Every two days, students measure and graph the height of their plants. As a class, students draw conclusions about light and plant growth based on their data.
B	Enhance	4th grade Energy Transfer	**Using Data:** • Collecting data with computational devices **Computational Simulations:** • Using computational simulations	The class reviews the vocabulary term *collision* and discusses the types of energy involved in collisions. Then, students use the PhET online simulation *Collision Lab* to collect data about two objects colliding. Students use the simulation to manipulate variables like the mass of the objects and collect data about each collision for analysis.
C	Extend	5th grade Water Pollution	**Programming:** • Coding **Using Data:** • Collecting data with computational devices	Students code a micro:bit to detect light levels. Then, using their own water samples, students measure how much light passes through water from a flashlight. Students record their data and analyze it using guided questions to make conclusions about pollution levels within the body of water from which they took their sample.
D	Exhibit	3rd grade Weather Animation	**All Programming Practices**	Students are introduced to extreme weather with an introductory video and sharing their own experiences. Then, students explore an extreme weather event by conducting guided research. To present their research, students create a Scratch animation “movie” telling about their weather event.

cycle, but these investigations rarely reach the point of interrogating the quantitative relationships within the system or representing the system using a computational tool. Future research could investigate whether elementary students can engage in systems thinking practices and develop strategies for teachers to integrate Systems Thinking from a CT Perspective.

The relationships between integration levels and CT practices we found provide insight into current gaps in knowledge and possibilities for future research around supporting teachers to write lessons with deeper levels of integration. We found that teachers who included only *Using Data* practices integrated CT and science at the *exist* level (3 of 3 lessons). This finding can partially be explained by the likeness between CT data practices and scientific inquiry, where students collect and analyze data. Further research could explore how PD can support teachers in leveraging computational aspects of data collection and analysis to move beyond *exist* level integration and into the *enhance* and *extend* levels. This effort is particularly important given that simply naming existing activities aligned with CT practices is unlikely to lead to instructional change and, therefore, new computational learning opportunities.

All lessons at the *enhance* level of integration utilized Computational Simulations (8 of 8 lessons). As the most popular tool used by teachers within their lesson plans, simulations appear to be a comfortable computational tool for integration. Yet, while teachers seemed comfortable integrating pre-made online simulations, they did not create their own simulations. Our findings suggest that integrating simulations can be an important starting point for teachers to integrate CT that can enhance science learning and inquiry. These integrations are particularly productive when studying scientific phenomena that are temporally too far away or spatially too small or large to see. However, future research could examine how teachers who feel comfortable integrating pre-made simulations could be supported to integrate CT more deeply by assessing and creating simulations with their students. While there is some important work on how students can engage with these practices (Basu et al., 2016; diSessa, 2000; Wilensky & Rand, 2015), the support that teachers need to venture into the *extend* level with simulations is less clear.

Regarding Programming practices, our findings show teachers need support to differentiate between programming activities that *extend* science learning (2 lessons) and those that only integrate science on an *exhibit* level (3 lessons). Although not included in the original Waterman et al. (2020) framework, we found exhibit to be a unique level of integration, representing the integration of science as a thematic topic in CT activities without learning-supportive integration. While the creation of a Scratch animation about a science topic is a valuable exercise to learn programming skills, the activity does not support further science learning—it can only serve as an *assessment* of content understanding. The emergence of the *exhibit* category raises questions about the integration of Scratch within CT lessons. As a tool utilized during our PD, teachers had some familiarity with Scratch. Because it is a programming environment, demonstrating a clear connection to coding and CS, the addition of Scratch was a clear-cut way for teachers to ensure they were integrating CT practices. Yet, the propensity to do so at a topic level rather than using more advanced computing such as conditionals or programming a simulation points to teachers potentially lacking confidence or knowledge with either programming tools, science, or both. To support teachers in making the differentiation between *extend* and *exhibit*, PD opportunities could include demonstrating examples of each and the differences in CT and science integrated learning they promote and providing further examples of programming that supports scientific learning.

The varied levels of CT integration highlight a need to examine the implications of different integration levels for equal and inclusive CT opportunities for students. As our findings demonstrate, even with PD focused on integrating CT in science, teachers have varied success writing CT-integrated science lesson plans. This has implications for the students in their classrooms, particularly because focusing on providing greater access to CT within classrooms is not enough to ensure equitable CT experiences for all students (Coenraad et al., 2020). If some teachers integrate at the *exist* level and others at the *enhance* or *extend* level, students are getting different levels of integration and thus different levels of preparation for the use of computing in jobs both within and outside of CS. Future research should consider the connections between school environment and teachers' level of CT integration to further understand the inequalities that could be perpetuated by different levels of CT integration. Integrating CT into science provides opportunities for *more* students to experience CT than if students only received instruction in elective or after school programs. However, as the field works toward providing equitable computing opportunities for all students, paying attention to the level of integration and the practices teachers are integrating will ensure quality opportunities for students.

Implications

Due to the nature of our study and the data we examine, implications of our work are particularly relevant to teacher educators as practitioners responsible for supporting teachers as they learn to integrate CT into disciplinary lessons. When planning and implementing PD, teacher educators should:

- Focus on supporting teachers to integrate CT in service of science learning rather than only building CT or CS skills.
- Provide explicit discussions of the levels of CT integration and moving beyond finding CT within the existing science curriculum.
- Present examples of Programming activities integrating CT at the *exhibit* and *extend* levels to demonstrate the differences in CT and science integrated learning they promote.
- Build teacher efficacy and confidence with programming environments to build their own simulations and lead students to use programming to increase understanding of science phenomena.
- Examine the barriers to integration teachers are facing and support them integrating CT at the *enhance* and *extend* levels despite the barriers they might face to provide more equitable learning experiences across CT practices for all students.

References

Barab, S. (2006). Design-Based Research: A methodological toolkit for the learning scientist. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (Issue 10, pp. 153–169). Cambridge University Press.

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning, 11*(1), 13. https://doi.org/10.1186/s41039-016-0036-2

Bean, N., Weese, J., Feldhausen, R., & Bell, R. S. (2015). Starting from scratch: Developing a pre-service teacher training program in computational thinking. *Proceedings of 2015 IEEE Frontiers in Education Conference (FIE)*. https://doi.org/10.1109/FIE.2015.7344237

Blikstein, P. (2018). *Pre-college computer science education: A survey of the field*. https://goo.gl/gmS1Vm.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education: Implications for policy and practice. In *JRC Science for Policy Report*. https://doi.org/10.2791/792158

Bort, H., & Brylow, D. (2013). CS4Impact: Measuring computational thinking concepts present inCS4HS participant lesson plans. *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)*, 427–432.

Bower, M., Wood, L., Lai, J., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education, 42*(3), 53–72.

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *Journal of the Learning Sciences, 2*(2), 141–178. https://doi.org/10.1207/s15327809jls0202_2

Cabrera, L., Ketelhut, D. J., Mills, K., Coenraad, M., Killen, H., & Plane, J. (2021). *Designing a Framework for Teachers' Integration of Computational Thinking into Elementary Science*. [Manuscript submitted for publication].

Chang, Y., & Peterson, L. (2018). Pre-service Teachers ' Perceptions of Computational Thinking. *Journal of Technology and Teacher Education, 26*(3), 353–374.

Coenraad, M., Cabrera, L., Byrne, V., Killen, H., Ketelhut, D. J., Mills, K. M., & Plane, J. (2021). *STIGCT: The Design of a Science Teaching Computational Thinking Inquiry Group to Promote CT Integration in Elementary Science*. [Manuscript submitted for publication].

Coenraad, M., Mills, K., Byrne, V. L., & Ketelhut, D. J. (2020). Supporting teachers to integrate computational thinking equitably. *Proceedings of 2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology, RESPECT 2020*. https://doi.org/https://doi.org/10.1109/RESPECT49803.2020.9272488

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014). Introducing teachers to computational thinking using unlugged storytelling. *Proceedings of the 9th Workshop in Primary and Secondary Computing Education (WIPSCE 2014),* 82–92. https://doi.org/https://doi.org/10.1145/2670757.2670767

diSessa, A. A. (2000). Changing Minds: Computers, Learning, and Literacy. MIT Press.

Dodero, J. M., Mota, J. M., & Ruiz-Rube, I. (2017). Bringing computatonal thinking to teachers' training: A workshop review. P*roceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*, 1–6. https://doi.org/10.1145/3144826.3145352

Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology, 26*, 175–192. https://doi.org/10.1007/s10956-016-9663-z

Ketelhut, D. J., Cabrera, L., McGinnis, R. J., Plane, J., Coenraad, M., Killen, H., & Mills, K. M. (2019). Exploring the Integration of computational Thinking into Preservice Elementary Science Teacher Education. *National Science Foundation STEM+C PI Meeting*. http://stemcsummit.edc.org/slides/DianeJass.pdf

Lamprou, A., & Repenning, A. (2018). Teaching how to teach computational thinking. *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE 2018*, 69–74. https://doi.org/10.1145/3197091.3197120

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology, 29*(1), 1–8. https://doi.org/10.1007/s10956-019-09803-w

McGinnis, J. R., Hestness, E., Mills, K., Ketelhut, D., Cabrera, L., & Jeong, H. (2020). Preservice science teachers' beliefs about computational thinking following a curricular module within an elementary science methods course. *Contemporary Issues in Technology and Teacher Education, 20*(1), 85-107.

Mouza, C., Yang, H., Pan, Y. C., Yilmaz Ozden, S., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology, 33*(3), 61–76. https://doi.org/10.14742/ajet.3521

National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. http://www.nap.edu/catalog/12840

NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States*. http://www.nextgenscience.org

Simmonds, J., Gutierrez, F. J., Casanova, C., Sotomayor, C., & Hitschfeld, N. (2019). A teacher workshop for introducing computational thinking in rural and vulnerable environments. *SIGCSE '19 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1143–1149. https://doi.org/10.1145/3287324.3287456

Tran, Y. (2019). Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. *Journal of Educational Computing Research*, 073563311774391. https://doi.org/10.1177/0735633117743918

Waterman, K. P., Goldsmith, L., & Pasquale, M. (2020). Integrating computational thinking into elementary science curriculum: an examination of activities that support students' computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology, 29*(1), 53–64. https://doi.org/10.1007/s10956-019-09801-y

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147.

Wilensky, U., & Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. Cambridge, Massachusetts; London, England: The MIT Press. doi:10.2307/j.ctt17kk851

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education, 14*(1), 1–16. https://doi.org/10.1145/2576872

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 60*(4), 55–62. https://doi.org/10.1145/2994591