# COMPUTATIONAL PEDAGOGY: FOSTERING A NEW METHOD OF TEACHING

Osman Yasar, Peter Veronesi, Jose Maliekal, Leigh J. Little,
Sounthone E. Vattana, Ibrahim Halil Yeter
The College at Brockport - SUNY

### Abstract

Teaching with technology still remains as a challenge. Making judicious choices of when, what and how specific tools and pedagogies to use in the teaching of a topic can be improved with the help of curriculum inventories, training, and practices but as new and more capable technologies arrive, such resources and experience do not often transfer to new circumstances. This article presents a case study in which computational modeling and simulation technology (CMST) is used to improve technological pedagogical content knowledge (TPACK) of teachers. We report findings of a summer training program for both preservice and in-service teachers in the Northeastern United States. CMST has shown to be effective on both teaching and learning. Results show that it helps teachers to integrate technology into their teaching in a more permanent, constructive, and tool-independent way. It has also shown to improve student learning in a constructive fashion by first enabling deductive introduction of a topic from a general simplistic framework and then guiding the learner to inductively discover underlying STEM principles through experimentation.

### General Terms

Technological Pedagogical Content Knowledge, Computational Pedagogy, K-12 Teaching

### Keywords

Deductive and Inductive Reasoning, Cognitive Processes, Memory Retrieval

### Introduction

Educators structure training and curriculum based on learning theories of how the human mind works. Recent findings from empirical research by cognitive psychologists and neuroscientists have created a critical mass to change the way we prepare teachers and support their classroom instruction. *Make it Stick*, an ostensibly groundbreaking book published recently and coauthored by several prominent cognitive scientists has turned conventional ideas of learning upside down. [9] The book offers many sound practices to help students easily retrieve content they learned in class, retain it, and apply it in different contexts to solve problems. New research suggests that repeated, delayed and interleaved retrievals make new concepts stick in memory longer if the process is effortful (pp. 47). Learning is mediated by memory, because human brain attempts to interpret new concepts in terms of previously registered knowledge and facts. However, some degree of forgetting is also good for learning because it forces the learner to use effort to cognitively engage oneself to recall or reconstruct newly acquired concepts through different neural pathways or links that exists and are retrievable. And, the more links to associated concepts, the higher the chances of recalling the newly acquired concept when needed later. Cognitive retrieval practices attempted at different times, various settings and contexts are good because every time the recall is attempted it establishes more links that will help the remembering and learning. Exposure to new concepts through links to multiple views from different fields of study is, therefore, an effective retrieval strategy recommended by cognitive psychologists (pp. 49). This is called *interleaved retrieval* practice and it forms a cognitive foundation for the interdisciplinary computational pedagogical content knowledge (CPACK) framework that has been developed recently by computational science practitioners and educators. [69] In the following sections, we will describe an in-service and a pre-service implementation of CPACK and how its findings relate to the current literature in engineering education and teacher professional development.

## Interdisciplinary CPACK Education

Interleaving retrieval practices by weaving together multi-disciplinary features around a common topic (i.e., interdisciplinary education) has great advantages for gaining deep and lasting knowledge but it is not easy for several reasons. It would require a more cognitive effort than usual and as such, it would slow down the process of learning. In college, it would delay graduation and in public schools' packed schedules it would risk compliance with local and state-mandated curriculum. Technology can be used to speed up this interdisciplinary learning but it needs training of teachers to teach content in pedagogically appropriate ways, thereby requiring a close integration of technology, pedagogy, and content as shown in Figure 1. Recently, a theoretical framework, namely technological pedagogical content knowledge (TPACK), has been developed by Mishra & Koehler [36] to address challenges of T, P, and C integration. Practicing teachers have been offered professional development (PD) to help them deploy appropriate technologies in the classroom, stay up-to-date with emerging technologies, and assess efficacies of different pedagogical approaches. [10,33] But, due to frequent changes in available tools, challenges might never go away as far as transferring curriculum inventories and PD content to new circumstances. Furthermore, teaching with technology often requires customization and the needed technologies must be both content specific and pedagogically suitable at the same time. [28] While latest technologies offer more capacity for applicability, their optimum utilization may necessitate knowledge of tools' operational underlying principles for easier transfer into new circumstances and better integration. [21,28,44,69,73]

There is an important feature of interdisciplinary education that can be best described by Aristotle's wellknown statement, "the whole is more than the sum of its parts," or the theory of Gestalt psychology, "the whole is other than the sum of its parts," which means that the whole has a reality of its own, independent of the parts. [30] Accordingly, educators have noted an emerging nature of TPACK when technology, pedagogy, and content are closely integrated. [36] When

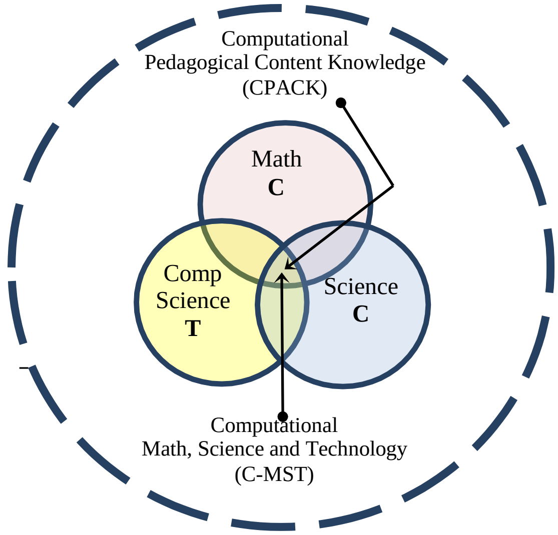


Figure 1: CPACK framework. [69]
Computational pedagogy is an inherent outcome of computing, math, science and technology integration.

mathematics, computing, and sciences are integrated, as shown in Figure 1, their integration gives birth not only to a new content domain of *computational science*, as witnessed by degree programs in the past two decades, [32,47,60,76] but also a particular computational pedagogy which was not among the constitutive domains of CMST to start with. [71-72] This multi- faceted interdisciplinary knowledge domain has been called Computational Pedagogical Content Knowledge (CPACK) domain framework, which has received a Best Paper award from the TPACK community. [69] CPACK involves the use of computational modeling and simulation tools in a pedagogical way that support both deductive and inductive [67,71] approaches to teaching and learning.

## Computational Modeling and Simulation Technology (CMST) & Relevant Pedagogies

Modeling and testing has been an important tool for scientific and engineering research for hundreds of years. Scientists often start *deductively* with a model (e.g., a hypothesis or a concept) based on the current research, facts, and information. They test the model's predictions against experimental data. If results do not match, they, then break down the model into its parts (sub models) to identify what needs to be tweaked.

They retest the revised model through *what-if* scenarios by changing relevant parameters and characteristics of the sub models. By putting together new findings and relationships *inductively* among sub models, the initial model gets revised again. This deductive/inductive cycle of modeling, testing, *what- if* scenarios, synthesis, decision-making, and re-modeling is repeated — as shown in Figure 2 — while resources permit until there is confidence in the revised model's validity. [6,47]

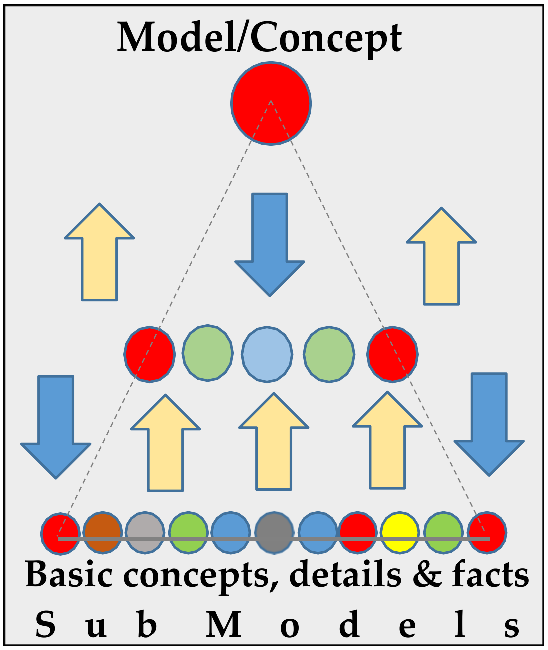


Figure 2: Scientific methodology of modeling-testing-remodeling process used in the conduct of research.

In recent years, computational modeling and simulation technology (CMST) has been very effective in conducting scientific and engineering research because computers speed up the model building and testing of different scenarios through simulations that provide quick feedback to researchers in order to improve the initial model. [45] CMST's role in scientific and industrial research was proven beyond doubt when computational predictions matched behavior of physical models in high-stake cases (e.g., safety of cars and planes, emissions from engines, and approaching storms). Its use was uniquely justified when a study was impossible to do experimentally because of its size (too big such as the universe or too small such as subatomic systems), environmental conditions (too hot or dangerous) or cost. CMST eventually demonstrated to be generating innovation and insight, just like experimental and theoretical research and this ultimately led to the recognition of computation by the scientific community as a third pillar of doing science besides theory and experiment.[6,47]

While such capacity was available only to a small group of scientists in national labs, their demand for computationally competent post-docs and doctoral students led to graduate programs in research universities. A dramatic increase in access to and power of high performance computing and the drop in its cost in the past 20 years helped spread the use of CMST tools into the manufacturing industry. Driven by market needs and trends, rather than empirical research into their effectiveness in education, funding agencies and colleges started investing in new CMST-based BS, MS and Ph.D. degree programs across the world. [31,56-57,60,76-78] It was not until friendly versions of such tools were available and considered for use in K-12 settings that a detailed and thorough empirical research was undertaken to measure their effectiveness in education (see reviews by Smetana & Bell [58], Rutten *et al.* [54], de Jong & Joolingen [14] and Yaşar *et al.* [73])

Modeling is a simplification of reality — it eliminates the details and draws attention to what is being studied. In education, it enables the learner to grasp important facts surrounding a topic before revealing the underlying details. Tools such as those in Table 1 now make it possible for instructors to offer easy experimentation in the classroom without having to expose students to underlying STEM concepts and principles. For example, Interactive Physics (IP) and AgentSheets (AS) can be used to create many fun things that could engage students into science experimentation, either by modifying an existing model or creating one from scratch. These user-friendly tools can shield students from having to know content knowledge of mathematics (e.g., differential equations), computing (e.g., algorithmic and programming) and science (e.g., physics) to conduct scientific experiments such as harmonic and planetary motion.

Simulation adds another level of benefit on top of easy modeling by providing a dynamic medium for the learner to conduct scientific experiments in a friendly, playful, predictive, eventful, and interactive way to test hypothetical scenarios. For example, in a harmonic motion of an object attached to a spring (Figure 3), Interactive Physics can provide control buttons to change physical parameters such as string constant, mass of the swinging object and its initial velocity, intensity of gravitational acceleration, among others. It also gives the user the ability to change some operational parameters, such as the run-time and accuracy desired from the simulation. Furthermore, it allows the learner to go into the initial model's details and break it into its constitutive parts in order to run various *what-if* scenarios. Based on these scenarios and their outcomes, the learner can go back to the design phase and change the model (spring and box) to his desire. This dynamic of making decisions that lead to modifications to the initial model based on what- if scenarios is an *inductive process* because it lets the learner to put pieces of the puzzle to come up with a revised model. [67]

Table 1. A typical list of CMST tools.

| |
|---|
| *Interactive Physics (IP)*: investigate concepts without prior physics background. http://www.design- simulation.com/IP. |
| *AgentSheets (AS)*: create games and simulations through agents and rules of engagement. |
| *STELLA*: model a system by a pictorial diagram of initial values and rate of change equations. http://www.iseesystems.com. |
| *Geometer's Sketchpad (GSP)*: model geometrical concepts; compute distances, angles & areas. |
| *Project Interactivate (PI):* online courseware for exploring scientific and mathematical concepts. |
| *Excel Spreadsheets:* conduct modeling & simulations using a simple algebraic equation (new = old + change) for rate of change. |
| *Texas Instruments (TI) Tools:* advance graphing tools to conduct algebra, functions, and rates of change. |

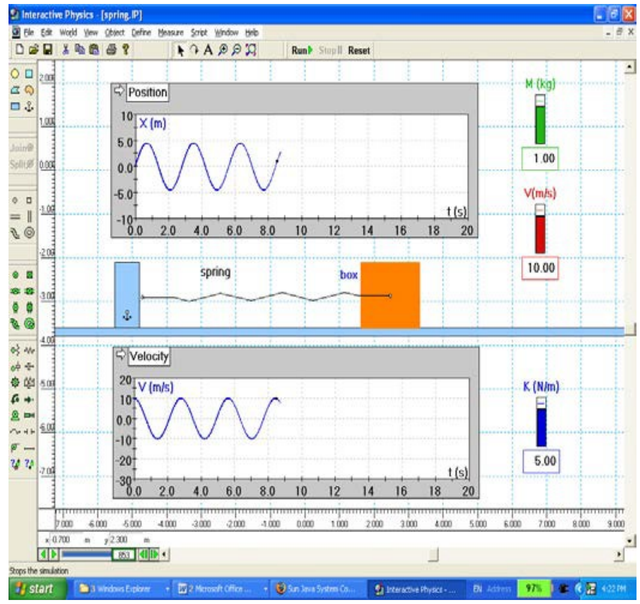


Figure 3: An example of harmonic motion by a box attached to a spring on a flat surface.

If used appropriately, CMST tools can involve students in inquiry-based, authentic science practices that are highlighted in the recent framework for K-12 science education. [42] A growing body of research [3-4,64] identifies computer simulation as an exemplar of inquiry-guided (inductive) learning through students' active and increasingly independent investigation of questions, problems and issues. Research into the use of computer simulations in science education has been reviewed periodically and quite frequently in recent years. These include early efforts by de Jong & van Joolingen [14] and by Bell & Smetana [4] as well as recent efforts by Rutten *et al.* [54] and by Smetana & Bell. [58] The article by the Rutten *et al.* [54] reviewed (quasi) experimental research in the past decade (2001-2010) and the one by Smetana & Bell [58] reviewed outcomes of 61 empirical studies since 1972. The overall findings support effectiveness of computer simulations. In many ways simulation has been found to be even more effective than traditional instructional practices. In particular, the literature shows that computer simulations can be effective in: 1) developing science content knowledge and process skills, and 2) promoting inquiry-based learning and conceptual change. Effectiveness of CMST in education is also well grounded in contemporary learning theories that recognize the role of experience, abstract thinking, and reflection in constructing knowledge and developing ideas and skills. [16,22,27,38,61] Cognitive aspects of CMST are being further detailed in a recent article by Yaşar [67] using a computational model of how the mind learns. Computational modeling and simulation is no longer an *adhoc* methodology or technology that scientists and engineers use in their narrow fields of study ─ everything in the universe, including matter and mind, is now suspected to behave computationally. [68] An awareness of computation's universality could help spread its utilization as a pedagogy in the advancement of STEM teaching and learning as briefly explained in the next section.

## CPACK Teacher Education

Supported by the National Science Foundation through various grants, and in partnership with local school districts, namely Rochester City School District (RCSD) and Brighton Central School District (BCSD), and national organizations (Shodor Foundation, Krell Institute, and Texas Instruments), we founded a CMST Institute in 2002 and have been offering CPACK training since then. While we constantly explore new CMST tools, we are currently using those in Table 1 because of a large repository of artifacts and lesson plans we have developed using these tools over the past decade. These modules have been downloaded by people around the world at a rate of 50-80 per day, totaling almost 100,000 since the database was launched. To this date, about 700 in-service and pre-service teachers from twenty local school districts have directly benefited from the CPACK training in the form of summer institutes and credit-bearing college courses. Partnering school districts, such the urban RCSD and the suburban BCSD, have reported using the training modules in their professional development days and teacher resources centers over the past decade, bringing the total of teachers affected by this initiative close to a thousand.

While the CPACK has been an initiative by practitioners of CMST in scientific research and education, the requirements by the sponsoring agency, particularly the National Science Foundation's Math and Science Partnership (MSP) program, helped evolve it and assess its impact by involving professional evaluators and educational researchers from other MSP Research, Evaluation and Technical Assistance (RETA) awardees. The RETA awardees that took an interest in the progress of CPACK work included the American Institute for Research, the Wisconsin Center for Education Research, Technical Education Research Centers (TERC), the Concord Consortium, and The Council of Chief State School Officers (CCSSO). The quantitative and qualitative evaluation methodology used by project evaluators were based on previously validated methods [11] and instrumentations from RETA studies as well as those found in TPACK [1,29,55] and PD literature. [19-20, 33] What follows are details and key findings from our implementation of in-service and pre-service CPACK programs.

### In-service Teacher Education

The preparation of in-service teachers to integrate CMST tools and pedagogy into their classrooms involved multiple approaches. This included a multi-tier in-depth instruction in the summer and yearlong activities such as weekend workshops as well as mentoring and coaching. The summer training was done in three steps by incrementally adding a new domain of knowledge at each year of training for the first three years. The first step included technology knowledge (TK) training, the second step included technological content knowledge (TCK) training, and the final step included teaching of STEM content through computational and pedagogical tools (i.e., TPACK). Participating teachers received 80 contact hours during their first summer institute and a minimum total of 200 by the end of their third year. They also received additional PD hours through TI-certification (~60 hours), Saturday sessions (~8 hours) and one-on-one training (~8 hours) via a CMST Coach and or Team Leader. Table 2 shows the number of in-service teachers who attended the summer institute during the life of the initiative from 2003 to 2008. Almost half of the teachers who attended TK training returned for additional TCK training, and half of those returned for the final TPACK training.

Table 2. Number of in-service teachers attending the summer training.

| Summer Training (2003-2008) | TK | TCK | TPACK | Total |
|---|---|---|---|---|
| Math Teachers | 110 | 44 | 22 | 176 |
| Science Teachers | 53 | 26 | 17 | 96 |
| Technology/Special Education Teachers | 25 | 8 | 4 | 37 |
| TOTAL in-service Teachers | 188 | 78 | 43 | 309 |

The Concord Consortium and The Council of Chief State School Officers (CCSSO) staff members conducted periodic on-site PD surveys in order to report back to NSF about our project's progress on the following aspects: 1) Partnership effectiveness, 2) Teacher preparation, 3) Curriculum and classroom impact, 4) Student achievement, and 5) Sustainability and institutional change. The quantitative and qualitative data collected from our teacher participants by these RETA projects was compiled along with other MSP projects for the purpose of overall accountability to the U.S. Congress. The sites and related data were not linked to each other in these reports, but based on their reporting, NSF highlighted our project in its reports to the Congress for its overall impact and as a result we were invited to testify before the U.S. Congress on behalf of NSF. [26]

The instrumentations used by our project evaluators benefited from those of the national RETA projects; all of which targeted the 5 areas listed above. Additionally, we used a commonly known Guskey model of professional development evaluation. [19-20] As seen in Table 3, Guskey's model involves examining five critical levels of evaluation, which basically correspond to the five aspects of project evaluation required by the sponsoring NSF program. The research and evaluation questions and how the responses were gathered followed the general outline in the table. Project evaluators collected quantitative data through school records, teacher journal entries, activity logs, interviews, and reflective answers to survey questions. Two independent evaluators read the text and coded the text segments to arrive at descriptions and common themes. An inductive process [11] was used to group these codes in order to form broad themes. The project employed additional independent experts to assist with content development and reviews of professional development.

In a survey of 40 participant teachers in 2010 who had at least two years of training, 94% agreed that the training made them more effective in the classroom; 87% agreed that it strengthened their pedagogical skills; 73% agreed that it strengthened their pedagogical content knowledge; 100% agreed that training strengthened their skills related to modeling and simulation; 86% reported that they continue to use the hardware, software and other materials made available through project in their classrooms; and 80% believed that their participation served to build leadership skills. Districts also reported high teacher retention – e.g., at the end of 7 years, 73% of participating teachers at RCSD were still teaching while 10% had moved to lead positions. This is better than the ~50% national retention rate. [41] Furthermore, according to district officials [12] the training helped retain veteran teachers and drew more and better teachers to an urban school with a hard time recruiting teachers because of the well-known urban problems. [35]

The percent of teachers feeling prepared to teach with computational tools and methodology after the first year TK training averaged as follows: 50% were confident about their preparedness and the remainder felt that they were "probably" prepared. After the second year of TCK training, 50% felt "definitely" prepared to use modeling with the remainder feeling "probably". The ongoing annual data suggested that after their first summer training, while knowledgeable about the CMST tools, teachers did not immediately feel fully prepared to put their training into practice. In fact, what teacher data revealed is that it was not until their third year of training that involved fully using CMST pedagogy and tools that the average

Table 3: Guskey's 5-level evaluation of Professional Development as applied to the CPACK project. [19-20]

| Evaluation Level | What Questions Are Addressed: | How Will Information Be Gathered? |
|---|---|---|
| **1.Participants' (Teacher) Reactions** | Did they like it?<br>Was their time well spent? Did the material make sense? Will it be useful?<br>Was the leader knowledgeable and helpful? | Pre- and post-activity questionnaires administered at the beginning and end of activity sessions |
| **2. Teacher Learning** | Did participants (teachers) acquire the intended knowledge and skills? | Paper-and-pencil instruments, Simulations, Demonstrations,<br>Participant reflections (oral and/or written),<br>Participant portfolios |
| **3. Organization Support & Change** | Was implementation advocated, facilitated and supported?<br>Was the support public and overt?<br>Were problems addressed quickly & efficiently? Were sufficient resources made available?<br>Were successes recognized and shared? What was the impact on the organization?<br>Did it affect the organization's climate & procedures? | District and school records<br>Minutes from follow-up meetings<br>Questionnaires<br>Structured interviews with participants and district or school administrators<br>Participant portfolios |
| **4. Teacher Use of New Knowledge and Skills** | Did participants effectively apply the new knowledge and skills?<br>To what degree are participants actually implementing new knowledge and skills? | Questionnaires<br>Structured interviews with participants and their supervisors<br>Participant reflections (oral and written) & portfolios<br>Direct observations<br>Video or audio tapes |
| **5. Student Learning Outcomes** | What was the impact on students?<br>Did it affect student performance or achievement? Did it influence students' physical or emotional wellbeing?<br>Are students more confident as learners/readers? Is student attendance improving?<br>Are dropouts decreasing? | Math 8, Science 8 exams<br>Regents exams: Math, Biology, Chem. and Physics<br>Course Enrollments, report cards & achievement scores.<br>Unit tests<br>Questionnaires<br>Structured interviews with students, parents, teachers, and/or administrators<br>Participant portfolios |

teacher felt confident and comfortable. This is consistent with the PD literature. [5]

When mastering new skills or strategies, the learner typically advances through a predictable series of learning stages. [23] At the start, the learner is usually halting and uncertain as he or she tries to use the target skill. With feedback and much practice, the learner becomes more fluent, accurate, and confident in using the skill. This process was typical of the CMST learners. To add a vital piece to findings in the literature, [5,23] our research suggested that a significant period of authentic practice in the classroom between training sessions was also critical in changing teacher's behavior and the classroom environment (in addition to the minimum of 80 contact hours to effect changes in teachers' instructional behaviors and a minimum of 160 contact hours to effect changes in the classroom environment). Our data implies that in this model the learning process for the average CMST Teacher appeared to occur in four overlapping stages as follows. [23]

1. Acquisition. The teacher has begun to learn how to complete the target skill correctly but is not yet accurate or fluent in the skill.

2. Practice. The teacher is able to complete the target skill accurately but works slowly. The goal of this phase is to increase the teacher's fluency with the tools and pedagogy.

3. Implementation The teacher is accurate and fluent in using the target skill but does not typically use it in different situations or on a regular basis. The goal of this phase is to get the teacher to use the tools and pedagogy in the widest possible learning situations.

4. Assimilation The teacher is accurate and fluent in using the skills learned. He or she also integrates the skill regularly in learning situations and is able to modify or adapt the skill to fit novel task-demands or situations.

Annual surveys of teachers showed that usage of the tools in the classroom was directly linked to the amount of training teachers received. All trained teachers reported that on a daily basis they used laptops for presentations, graphing calculators for math instruction, and electronic smart boards for interactive lessons. In post-training journals, while only 60% of the teachers reported occasional use of modeling tools in their classrooms after initial TK training, 78% of teachers reported that they used these tools regularly after going through the TPACK training.

In a 2007 survey of 65 active teachers who had received at least two years of training, many reported a significant use of modeling tools for both classroom instruction and special projects (see Table 4). It appears that the higher the grade level, the more regularly these tools were used in the classroom. In the survey, teachers who reported regular use of modeling tools agreed that using such tools in their classrooms significantly increased student engagement. As seen in Figure 4, students in higher-grade levels found computational modeling more engaging in both math classes (grades 7-8: 77% vs. grades 9-12: 90%) and science classes (grades 7-8: 75% vs. grades 9-12: 85%). Modeling was even found helpful to non-traditional (special education) learners (Figure 5); again the higher the grade level the higher the engagement: math classes (grades 7-8: %76 vs. grades 9-12: 100%) and science classes (grades 7-8: 75% vs. grades 9-12: 85%). 92% of surveyed teachers agreed that computational inquiry made math and science concepts more comprehensible to students. 72% of math, and 31% of science teachers reported observed improvement in problem solving skills. Student reaction to modeling (versus traditional techniques) was found to be 97% favorable in math and 77% in science classes. While science classes utilized technology less due to limited access and lack of science-related modeling examples, in instances where it was utilized, a deeper understanding of science topics was achieved, compared to math topics (83% vs. 76%).

| Table 4: Grade Level | Frequency | | |
|---|---|---|---|
| | Regularly | Special Project | No |
| 7-8 Math | 46% | 46% | 8% |
| 9-12 Math | 60% | 35% | 5% |
| 7-8 Science | 25% | 75% | 25% |
| 9-12 Science | 54% | 38% | 8% |

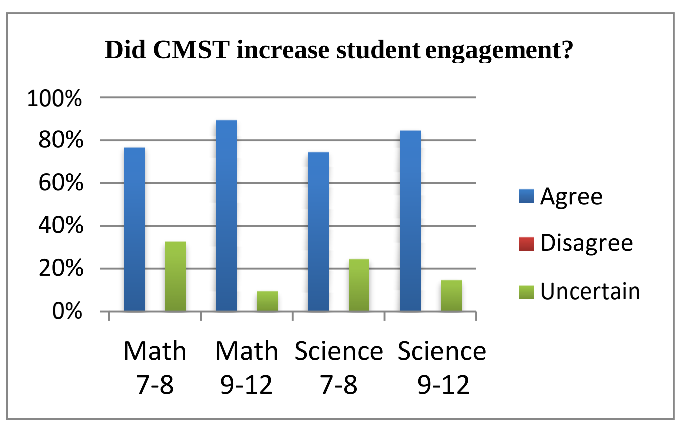


Figure 4: Student engagement per grade level.

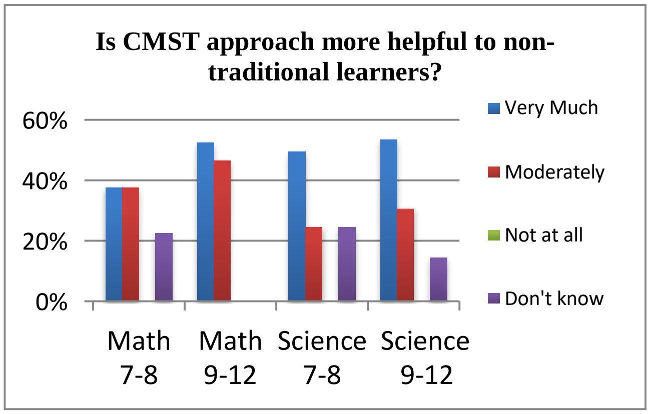


Figure 5: Impact on non-traditional learners.

To further triangulate self-reporting data by teachers, annual student achievement data were analyzed in the partnering school districts via report cards and standardized test scores. While we cannot fully isolate the impact of teacher training from other contributing factors, an upward district-wide trend was noted in both urban and suburban districts during the life of the initiative from 2003 to 2008 (see Tables 5 and 6). The percentage of students receiving a Regents diploma increased significantly from the baseline (RCSD: 21% → 59%, BCSD: 84% → 95%). The initiative exposed students from the urban district to college experiences and opportunities, and this may have led to an increased interest (78% → 83%) in both 2-year and 4-year college enrollments over the period examined. Furthermore, the passing rate (>65/100) in NY State Grade-8 Math exam increased in RCSD from 10% to 33%, while the passing rate in NY Regents Math-A exam (Grade 11-12) also increased from 13% to 67%. Passing rate in sciences also increased in areas such as Physics (3% → 22%) and Chemistry (9% → 27%). At BCSD, passing rates improved in mathematics (Math-A: 51% → 99%) and sciences (Physics: 52% → 78%). The number of students taking General Physics at Brighton increased from 50% to ~100% and the number of students taking AP Physics also doubled. Student passing rates at both districts seemed to reflect relative participation of district's math and science teachers in the initiative. All of the improvements have been found to be statically significant for sample sizes from each district.

Table 5: Student achievement at RCSD.

| School District | Passing Rate > 65 | 2002 | 2008 |
|---|---|---|---|
| RCSD: 35,000 students & 400 MST teachers | Grade 8 Math | 10% | 44% |
| | Grade 12 Math-A | 13% | 67% |
| | Grade 12 Physics | 3% | 22% |
| | HS Diploma | 20% | 56% |

Table 6: Student achievement at BCSD.

| School District | Passing Rate > 65 | 2002 | 2008 |
|---|---|---|---|
| BCSD: 3,000 students & 40 MST teachers | Grade 8 Math | 89% | 91% |
| | Grade 12 Math-A | 51% | 99% |
| | Grade 12 Physics | 52% | 78% |
| | HS Diploma | 84% | 95% |

While cohorts of 8$^{th}$ grade male and female students from both districts had a gap in their average math performance at the beginning of the initiative, not only were the gaps closed but also reversed four years later (12$^{th}$ grade) as shown in Table 7. At RCSD, while both male and female students did much better than four years earlier, the graduation rate of the same cohorts still reflected a gender-based trend in performance growth, favoring female students. To examine whether the difference is statistically significant, we calculated the two-proportion z-scores assuming a normal distribution approximation (Brase & Brase 2012). The sample sizes for male and female students were roughly the same at both districts, with about 1200 at RCSD and 150 at BCSD. The column $p$ indicates the confidence level that the difference between males and females may be due to a nonrandom effect. Normally, any confidence level below 90% is less than significant. Here, with more than 90% confidence level female cohorts outperformed male cohorts in both math performance and graduation rates. This is consistent with gender-based response to AgentSheets as reported by Repenning. [52]

The main goal of the sponsoring *No Child Left Behind* program was to train as many teachers as possible to potentially create a district wide impact on student achievement scores. As a result we trained twice as many as we had committed to (see Table 2). While the goals of the sponsoring

Table 7: Gender-based response to CMST at RCSD & BCSD.

| | | 2001-2002 | | 2005-2006 | | | |
|---|---|---|---|---|---|---|---|
| | | Male | Female | Male | Female | *z* score | p (%) |
| RCSD | Math Cohort | 13% | 10% | 41% | 49% | 3.97 | 99 |
| | Graduation Rate | | | 34% | 44% | 5.06 | 99 |
| BCSD | Math Cohort | 92% | 84% | 93% | 93% | 0 | 0 |
| | Graduation Rate | | | 85% | 90% | 1.29 | 90 |

agency were met, as witnessed by gains in the standardized test scores reported by partnering districts, no comprehensive research was done by the project to more closely link the gains in student achievement scores to the teaching and learning resulted from the initiative. By the time the goals of sponsoring NSF program shifted from ‘leaving no child behind’ outreach to ‘researching the interventions’ we had almost run out of control groups in partnering school districts’ math classrooms. The initiative invited science teachers but limited access to computer labs, skepticism about use of technology, and inadequate number of readymade curricular modules discouraged many to invest in trainings that lacked significant science content and representative lesson plans. By the end of the project while almost all secondary math teachers in RCSD and BCSD received training and yearlong PD, only 20% of science teachers took part.

In final years of the study, when focus shifted towards researching the intervention, a few treatment- control comparisons were conducted. A pair of CMST and non-CMST high school teachers from the same school taught properties of quadrilaterals in a mathematics class. The CMST teacher used GSP in a class of 24 pupils while the non-CMST teacher used conventional methods in a class of 14 pupils. Both teachers conducted the same unit test. Even though the CMST teacher taught a more crowded class, his classroom average was 82.5 versus 49.5 for the other class. The second study involved a math triathlon similar to Regents Math A and B tests involving use of TI graphing calculators. Scored by external judges, including teachers and college faculty, this study revealed that students taught by CMST teachers outperformed other students in all categories: Math-A: 60.26 vs. 49.54; Math-B: 71.9 vs. 55.6; and 7-8 Grade Math: 64.0 vs. 58.6. These findings are consistent with previously reported data on the impact of CPACK professional development on teaching and learning in Yaşar *et al*. [73] as well the pedagogical [70-71] and cognitive aspects [67] of computational modeling and simulation methodology.

**Pre-service Teacher Education**

The curriculum and lesson plans database developed by participation of in-service teachers provided content to three general education courses (*CPS 101 Introduction to CMST Principles*, *CPS 105 Games in Sciences, and CPS 302 Science, Technology, and Society*) and a pre-service methods course (*NAS 401/501 C-MST Methods and Tools*) in the college’s teacher education program. These courses have become part of the NSF Robert Noyce Scholarship program since 2012 to educate a new cadre of computational STEM teachers whose responses are being reported in this section. Table 8 shows enrollments in the pre-service methods and tools course (NAS 401/501). The content of the NAS course is similar to the first year TK training (i.e., exposure to the tools listed in Table 1) received by in-service teachers.

The major purpose of the Noyce program was to recruit pre-service science and mathematics teachers and improve their computational and pedagogical skills. So far, as shown in Table 9, the Noyce program has enrolled 16 interns (undergraduate students who have demonstrated

Table 8. Number of pre-service teachers enrolled in the NAS methods and tools course.

| Courses | 2003-2007 | 2008-2012 | 2013-2015 | Total |
|---|---|---|---|---|
| NAS 401/501 C-MST Tools for Teachers | 113 | 107 | 105 | 325 |

Table 9: Profiles of Noyce scholars and interns.

| | Cohort I | Cohort II | Cohort III | Cohort IV |
|---|---|---|---|---|
| Graduate | 5 | 8 | 5 | 2 |
| Undergraduate | 6 | 9 | 3 | 5 |
| Interns (Undergraduate) | 5 | 10 | 1 | 0 |
| | | | | |

some interest in teaching as a career) and 50 scholars (undergraduate and graduate students enrolled in the college's teaching certification program). Cumulative demographics for all cohorts (I through IV) indicate a distribution of 55% female and 45% male students. The interns have no obligations for the summer support they get other than taking a CMST course (e.g., CPS 101) afterward but the scholars are required to serve in a high-needs school district for the tuition support they receive prior to completion of teaching certification. Programmatic requirements for the scholars include academic preparation in three domains, including a) content area, b) educational methods, and c) computational pedagogy. Besides content and education courses, they are required to take *CPS 101* and *NAS 401/501* courses and attend an intensive two-week capstone experience in the summer to develop skills of integrating computational technology with content teaching in their subject areas.

One of the biggest challenges facing TPACK teacher education is to teach pre-service students how to judiciously choose the pedagogical technologies that can help them teach a topic in their content areas. Given the availability of multiple tools these days, teachers and pre-service students are faced with the burden of mastering a good many before making judicious choices. A tool-independent education could remedy the situation as mentioned earlier if the underlying mathematical and computational principles of modeling and simulation technology are learned. The plan was to have the Noyce scholars learn these principles in the CPS 101 course. Also, while the original plan included only one summer experience, based on our experience from in-service training we added 2nd summer experiences to give scholars additional time and support to further improve their CPACK skills before graduation. All in all, the CPACK pre-service training was similar to the multi-tier in-service training except that the pre-service students had an additional opportunity to learn mathematical and computational principles of modeling and simulation tools.

Since the beginning of the Noyce initiative in 2012, project evaluators have conducted focus groups interviews and pre- and post-activity surveys. These surveys were developed based on previous studies in the literature on how to measure TPACK [1,24-25,29,55] and general PD skills. [19-20,33,65] While the TPACK literature usually covers general information technology skills, our focus has been rather on interdisciplinary computational technology skills. Noyce students were given an annual satisfaction survey at the end of the summer course. An inductive approach [46] was used for analyzing the participants' responses to the open-ended questions. The inductive approach allowed for themes to emerge from the data instead of predetermined patterns. First, the data was read carefully and core categories were developed to describe the participants' perceptions about using computational modeling as an instructional method including topics, purposes, instructional approaches, and challenges. The data was then organized in a matrix to look for cross-case themes.

The quantitative Likert-scaled surveys initially attempted to measure student satisfaction with workshop content, learning new software skills and principles of computational modeling and simulation as well as benefits of group work and projects. A typical survey response from 14 students in 2014 is shown in Table 10. As the project evolved and new personnel were added, surveys attempted to measure additional responses, particularly the following values as shown in Table 11 [65]: 1) *Intrinsic Value (IV);* questions 1- 10: *How much pre-service teachers enjoyed engaging in the collaboration among STEM subjects – i.e., integration of technology, science, and mathematics*. 2) *Attainment Value (AV);* questions 11-17: *How much importance pre-service teachers place on doing well in their computational modeling coursework*. 3) *Utility Value (UV);* questions 18-26: *How likely pre-service teachers feel that being successful in the creation of the STEM model will lead to success integrating STEM content in their future classrooms*. Table 11 shows average scores (out of 5) from 2015, which we will mention later. Here, Pre-activity average scores for all 3 groups (IV, AV, and UV) are high, indicating that these pre-service students had an overall positive attitude coming into the workshop. From pre-activity intrinsic value scores, it appears that students came in with a strong interest and motivation, and they had high expectations (average of questions 1-10 is 4.12). Post-activity average scores improved for every question; on the average it went up about 5% for all groups. The consistent improvement in all categories points to a favorable trend. However, because of the sample size, we cannot make any significant statistical inferences and generalizations.

Due to changing logistics and project personnel, we have not been able to conduct a longitudinal study to see the evolution of student responses over the 4-year duration of the project, but we were able to conduct a study on the effect of training amount on a cohort of students. Preliminary results were presented in 2014 Association for Science Teacher Education [39] and EDULEARN Conferences. [40] Additional data since then indicates that participants' perception of CMST-based instructional methods has been highly positive after the training. In addition to the focus group interviews, semi-structured interviews with 6 participants (3 sciences and 3 mathematics) were conducted a semester after the initial summer training. Interpretive, qualitative analysis of open-ended questions and interview transcripts indicated that students perceive that computational modeling can be used to help them understand science concepts in various ways, including visualization of science concepts, improving critical thinking and problem solving skills, and understanding real-world application of mathematics. Pre-service teachers' perception of what technology knowledge (TK), technological content knowledge (TCK), pedagogical content knowledge (PCK), and technological pedagogical content knowledge (TPACK) means improved after their 1$^{st}$ year exposure to CMST tools, as shown in Table 12. Understanding of computational modeling increased their interest in teaching as unanimously stated by expressions such as "I am more interested in teaching than ever and I hope to create unique lesson plans and laboratory modules utilizing modeling and simulation technology."

| Table 10: 2013 survey | Very satisfied | Somehow satisfied | Not satisfied | None of these |
|---|---|---|---|---|
| 1. Workshop content | 8 | 4 | 0 | 0 |
| 2. Learning new software skills | 9 | 5 | 0 | 0 |
| 3. Learning CMST principles | 8 | 6 | 0 | 0 |
| 4. Project based learning | 14 | 0 | 0 | 0 |
| 5. Group work (collaboration) | 8 | 5 | 1 | 0 |
| 6. Instructors' knowledge and skills | 12 | 2 | 0 | 0 |

Table 11: Responses by pre-service students before and after the 2015 summer training.

| Please indicate how you feel about the following statements by circling the best representative of your perspective. SA = Strongly Agree (5); A = Agree (4); N = Neutral (3); D = Disagree (2); SD = Strongly Disagree (1) | | Average Score | |
|---|---|---|---|
| | | Pre | Post |
| 1 | I am committed to developing program skills to integrate tech into teaching cross cutting | 4.1 | 4.2 |
| 2 | I want to continue developing programming skills to teach cross cutting concepts. | 3.9 | 4.1 |
| 3 | Technology can be used to motivate learning of science and math concepts. | 4.8 | 4.9 |
| 4 | I would enjoy designing instruction by combining math and science concepts with technology. | 4.0 | 4.5 |
| 5 | I want to pursue computational modeling as a means to teach STEM content. | 3.7 | 4.0 |
| 6 | I like integrating technology into the instruction of science and math. | 4.4 | 4.5 |
| 8 | I would enjoy integrating modeling into the teaching of my content. | 4.1 | 4.2 |
| 9 | I would enjoy teaching STEM content through modeling. | 4.0 | 4.3 |
| 10 | I enjoy combining modeling with the teaching of content within my major. | 4.1 | 4.2 |
| 11 | I value modeling as a way to integrate science and mathematics content. | 3.9 | 4.3 |
| 12 | Mathematics is important for modeling real world problems. | 4.3 | 4.6 |
| 13 | Technology is important for teaching across the curriculum. | 4.5 | 4.7 |
| 14 | It is important to integrate modeling programs with instruction of science. | 4.1 | 4.3 |
| 15 | It is important to integrate modeling programs with instruction of math. | 3.9 | 4.0 |
| 16 | Realistically, modeling can be used as a means to teach mathematics. | 4.1 | 4.3 |
| 17 | Modeling is an important tool for teaching cross cutting concepts. | 4.1 | 4.1 |
| 18 | I am developing modeling skills that can be used to teach in my content. | 4.0 | 4.1 |
| 19 | I am confident I can model mathematical concepts. | 3.4 | 3.8 |
| 20 | I can use modeling to design teaching modules. | 3.7 | 4.0 |
| 21 | I am confident I can model the cross cutting concepts. | 3.4 | 3.8 |
| 22 | I am confident I can provide problem-solving opportunities using models. | 4.1 | 4.2 |
| 23 | I am confident I can model scientific concepts. | 3.9 | 4.0 |
| 24 | I am confident I can combine scientific and math content to teach the cross cutting concepts. | 3.9 | 4.3 |
| 25 | I can model mathematics and science concepts using technology. | 4.1 | 4.3 |

Participants felt after their initial exposure that they needed more training and experience to practice integrating technological content knowledge (TCK) with technological pedagogical knowledge (TPK) in order to teach topics in their areas of teaching. Based on this input, in the following years, a 2nd summer workshop was added to the program.

The 2nd summer experience included a review of CMST principles to make sure students who had not taken CPS 101 had some understanding of tool-independent operation of CMST tools. This involved replicating some of the earlier simulations ─ done with tools in Table 1 ─ using Excel and programming languages such as freely available Scratch (scratch.mit.edu). Evaluators asked open-ended questions through focus group interviews and Likert-scaled questions through surveys. Table 13 shows quantitative responses to programming tools from a class of 14 pre-service students. Interactive Physics (IP) and AgentSheets (AS) are easy to use because of their graphical user interface but their multiple features give an impression of complexity that a learner may never feel proficient enough to overcome.

The response to using Scratch and Excel has been overwhelmingly positive in comparison to the tools, such as IP and AS, that they had been using since the first training. While Scratch and Excel are simple tools, they enable students to see what computations are done and how they are done to model and problem and simulate its dynamics. For example, the harmonic motion done with

Interactive Physics (Figure 3) is replicable in Excel or Scratch by using a simple algebraic formula, *new= old + change*, that can be applied to position ($xnew = xold + dx$) and velocity ($vnew= vold + dv$) of a spring-driven object at times ($tnew = told + dt$) separated by an interval of choice *dt*. Here, change in *x* and *v* can be computed via $dx= v \cdot dt$ and $dv= a \cdot dt$ if acceleration *(a = Force/mass)* is known. Since the force applied by a spring with a stiffness *k* to an attached object of mass *m* is $F= - k \cdot x$, then $a= - (k/m) \cdot x$. A simple iterative computation, as shown in Table 14, can then be used to generate position and velocity profiles as predicted by the IP in Figure 3.

Table 12: Pre-service teachers' perceptions of using CMST for teaching based on TPACK framework.

| TPACK Category | Before the program | After the program |
|---|---|---|
| **TK** | MS office/Excel etc. | Knowledge of using CMST tools |
| **TCK** | Visualization of small scale and unobservable phenomena or complex system. | Recognition of the differences between CMST tools to represent and teach certain science and math concepts. |
| **TPK** | Motivation/Interest | Recognition of the benefits of using CMST tools for improving teaching efficiency, student engagement, motivation, and classroom management. |
| **PCK** | | Feeling more comfortable integrating science and math concepts. Recognition that CMST reinforces the connections between STEM fields. |
| **TPACK** | | Concrete ideas of how CMST tools can be used for improving student understanding of science and math concepts, inquiry skills, and problem solving. Recognition that CMST tools can help teach difficult concepts such as those involving abstract ideas and extremely small-scale or global phenomena. |

Table 13: Responses to question of "How helpful are these tools for learning computational modeling."

| | Very helpful | Somewhat helpful | Not helpful | I did not understand it well |
|---|---|---|---|---|
| Interactive Physics (IP) | 3 | 4 | 4 | 3 |
| AgentSheets (AS) | 11 | 3 | 1 | 0 |
| Scratch (programming-based) | 14 | 0 | 0 | 0 |
| Excel (computation-based) | 10 | 4 | 0 | 0 |

Table 14: Steps to simulate the harmonic motion.

```
Input initial position (x), velocity (v), and time (t) Input
time step (dt), maximum time (T), mass (m) While t <=T:
        Print position (x), velocity (v), and time (t)
        Compute force: F = - k     · x
        Compute acceleration: a = F/m
        Compute velocity:     v = v + a × dt
        Compute position: x = x + v × dt
        Update the time: t = t + dt
End of While Loop
```

One of the most important benefits of learning fundamentals of computational modeling is to understand that a computation is only an approximation of the reality and that its accuracy increases if we use smaller time steps ($dt$) ─ the smaller the step, the more data points to compute. There is cost for accuracy. Another important benefit is that a strong link gets established between computing and natural sciences through the computation of *change* because computation of change in position and in velocity requires computation of acceleration, which itself requires knowledge of the Force that is causing the motion. Learning principles of modeling and simulation can interest computer science majors into learning laws of natural sciences.

While computational modeling and simulation is as an effective pedagogy [71] to expose non-science majors to STEM concepts in an incremental fashion by using tools that hide the underlying mathematics and science involved in the simulations, it can also motivate STEM majors to learn computer programming. By using multiple tools (IP, Excel, and Scratch) to solve the same problem, learners get a chance to weigh advantages of each tool and conclude first-hand that more accurate and faster computation of *new =old + change* for a large number of data points will require computer programming. The responses by pre- service math and science students in our program are consistent with such expectation as they indicated a strong desire to learn and teach programming and computational modeling to young students (see Table 15). So, learning fundamental operation of computational and simulation methodology and being able to generate the same simulations with multiple tools seem to be an effective way of giving pre-service teachers the high confidence and the choice that they need to judiciously and comfortably choose what tools to use with the teaching of a specific topic.

## Conclusion

Effectiveness of computational modeling and simulation technology in teaching and learning has been reported extensively in this manuscript. What our work adds to the literature is more complete and user- friendly understanding of the cognitive and pedagogical aspects of CMST for engineering educators, along the lines of what other studies have done for science educators. [67-78] Our previous studies have generally reported in-service and pre-service education programs separately, and this is the first attempt to put them together within a single framework, the CPACK. As stated before, CPACK is a special case of TPACK in which the technology employed is computational modeling and simulation technology. [69] Results from our in-service and pre-service TPACK experience show that fundamental knowledge of how a particular technology works could help teachers to integrate it into their teaching in a more permanent, constructive, and tool-independent way. This TPACK knowledge is often of interdisciplinary nature and it might require a substantial amount of training. Logistically, it is easier to include such preparation of teachers in a pre- service program as it can be spread into several courses and capstone experiences. In our program, while only a quarter of in-service participants reached a mastery level of CMST principles within a 5-year timeframe, all of the pre-service participants accomplished it in just two academic years.

| Table 15: Common themes from interview transcripts |
| --- |
| • Scratch is really useful to look at parts of different models and see the math and the physics behind it. It was really a good exposure to those things and it kind of connects everything together.<br>• Scratch allows students to see what's going on a little bit better; plus you can see what other people have done. You can look at their code and see what goes on. It's so simple that even a nine-year-old can do it.<br>• I went home and showed my daughter Scratch. Within 5 minutes she created a program. And, that really showed me, you know that my students can do it too. And, the fact that Scratch allows you to share helps when having trouble getting your program to work.<br>• I could use Scratch with calculus, trigonometry, geometry, and definitely with integrals and derivatives.<br>• I would like to have a little more text based interface exposure to programming. If students are able to replicate what is shown or taught, then true learning will take place. |

While we have not had a chance to study the impact of pre-service teacher preparation on student learning, the evidence from partnering school districts where the in-service teachers taught support what other researchers have reported about the effectiveness of CMST-enhanced teaching. When used together, computational modeling and simulations affords the learner the opportunity to cycle iteratively back and forth between the deductive and inductive approaches to learning. [49-50,67-72] CMST has also shown to improve student learning in a constructive fashion [17] by first enabling deductive introduction of a topic from a general simplistic framework and then guiding the learner to inductively discover underlying STEM principles through experimentation. If used appropriately in the context of real world applications, CMST tools can involve students in inquiry-based, authentic science and engineering practices that are highlighted in the recent framework for K-12 science education. [42-43] For instance, the K-12 Framework suggests that performance expectations combine relevant science and engineering practices with core disciplinary ideas and crosscutting concepts that are appropriate for students at each grade level. It is the crossroads of performance expectations, relevant practices, core disciplinary ideas and crosscutting concepts that the deductive aspect of computational modeling could help with in order to adjust the level of exposure to scientific and engineering principles. Further, deeper understandings of science and engineering practices could emerge based on the grade level these tenets would be designed for.

High levels of student engagement reported by our participating teachers strongly support the effectiveness of computational modeling as a deductive pedagogical tool. It shielded students from having to know detailed content knowledge of mathematics (e.g., differential equations), computing (e.g., algorithmic and programming) and science (e.g., physics) to conduct experiments of linear, harmonic, and planetary motion. Once immersed into an authentic experimentation through computer-based simulations, students can naturally engage in the eight practices of science and engineering as identified by the Appendix F of the K-12 Framework, including inductive analysis and interpretation of data which could lead to a constructive experience, conceptual change as well as modification to the initial design model. The inductive process resulting from experimentation through simulations helps learners to rediscover principles of computing and sciences, therefore leading to deeper content learning. Project-based experiences reported in the NSF's MSPNET.org digital library by a group of 9$^{th}$ and 10$^{th}$ grade high school students from BCSD High School (NY) offers a testimony of how students could gain a deeper understanding of STEM concepts. [79-80] Improved student achievement scores in both local and statewide exams at partnering school districts also point to a lasting impact of the dual nature of computational pedagogy on learning. Computational thinking (CT) is heavily emphasized by the K-12 Framework and the NGSS standards as an element of recommended science and engineering practices. It is with the newly emerging NGSS themes and frameworks that highlight issues espoused in this paper that we believe that problem decomposition and abstract thinking aspects of CT skills [2,66] can be naturally fostered through the deductive and inductive reasoning cycle of computational pedagogy that has been articulated in Yaşar. [67] The top-down and bottom-up arrows in Figure 2 can help illustrate the parallels between distributive nature of deductive reasoning and decomposition as well as between associative nature of inductive reasoning and abstract thinking.

While our initial focus on pedagogical aspects of CMST was to develop a tool-independent TPACK training for our teacher education program in order to maximize transfer of curriculum inventories to new conditions when newer technologies become available, we have actually stumbled upon much more. Information revolution has taken electronic computing devices to every corner of the globe but still very few would be familiar with and relate to computational modeling and simulation. In fact, even some researchers and educators might consider CMST as an *ad hoc* technology. Furthermore, computing is usually not considered as a branch of science [15] because it deals with artificial phenomena, not natural phenomena. However**,** as artificial and imitational as electronic computation has been since 1936 by its inventor (Alan Turing [62]) we believe that it might eventually help us discover how the biological computation (i.e., the mind) generates complex mental states. [37] We actually think that it might even do more than that because a full understanding of how pervasive computational behavior is in the universe could change how we relate to ourselves and everything else in the universe. [68] That, indeed, would be a noble service to what other sciences try to accomplish.

Effectiveness of computational modeling and simulation processes resonates well with how the mind itself works because it, too, uses a similar dual methodology (distributive and associative) in its information storage and processing. [9,34,37,53,67]. A scientist's mind is a good example of how a mind learns best because it utilizes the scientific methodology. [6] Computational modeling and simulation process is nothing but the scientific methodology itself, except that it is put on turbo because computers speed up the modeling and testing process which was illustrated earlier in Figure 2. So, since the latest learning theories [7] as well as the new K-12 Framework for next generation science standards [42-43] suggest that students learn better if they are engaged in activities closely resembling the way scientists think and work, then this suggests, at least theoretically, that computational pedagogy have the potential to foster a new way of teaching and learning, as documented here in this article. The remaining challenge is to scale this up [13] to a national level by creating programs, curriculum modules, tools, and databases to help prepare a greater number of teachers to implement the science and engineering practices recommended by the national standards.

## Acknowledgement

This work was supported by the National Science Foundation (NSF) funds from the Math and Science Partnership (MSP), Transforming Undergraduate Education in STEM, and Robert Noyce Scholarship programs via Grants EHR-0226962, SCI-0520036, DUE-0942569, and DUE-1136332. We would like to thank the faculty and teachers whose efforts contributed to the development, teaching, and assessment of the reported courses and materials.

## References

1. Archambault, L., & Crippen, K. (2009). "Examining TPACK among K-12 online distance educators in the United States." *Contemporary Issues in Technology and Teacher Education*, *9*(1). Retrieved from http://www.citejournal.org/vol9/iss1/general/article2.cfm.

2. Armoni, M. (2013) "On Teaching Abstraction to Computer Science Novices." *J. Comp in Math & Science Teaching*, 32 (3); 265-284.

3. Bell, L. R., Gess-Newsome, J. and Luft, J. (2008) *Technology in the Secondary Science Classroom*. National Science Teachers Association (NSTA).

4. Bell, L. R and Smetana, L. K. (2008). "Using Computer Simulations to Enhance Science Teaching and Learning. In Technology in the Secondary Science Classroom," (Eds. Bell *et al*.). Washington, DC: NSTA Press.

5. Beatriz, C., *et al.* (2005). *Review of Evaluation Studies of Mathematics and Science Curricula and Professional Development Models*. Urban Institute, Supported by GE Foundation (2005). http://www.urban.org/.

6. Bradford, A. "Deductive Reasoning vs. Inductive Reasoning." *LiveScience*, March 23, 2015. http://www.livescience.com/21569-deduction-vs-induction.html

7. Bransford, J., Brown, A. and Cocking, R. (2000). *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, D.C.

8. Brase, C. H. and Brase, C. P. (2012). *Understandable Statistics.* 10th Edition. ISBN: 0840048386.

9. Brown, P. C., Roediger, H. L. and McDaniel, M. A. (2014) *Make it Stick*. The Belknap Press of Harvard University.

10. Bybee, R. W., and Loucks-Horsley, S. (2000). “Advancing Technology Education: The Role of Professional Development.” *The Technology Teacher*, Oct. 2000, 31-34.

11. Creswell, J. W. (2012) *Educational Research: Planning, Conducting and Evaluating Quantitative and Qualitative Research*. 4th Edition. Pearson Education, Inc.

12. Crowley, M. (2007). Rochester City School District. Mathematics Curriculum Director. Private communication. Email: Margaret. Crowley@rcsdk12.org.

13. Dede, C., Honan, J., and Peters, L. (2005). *Scaling Up Success: Lessons Learned from Technology-Based Educational Improvement.* John Wiley and Sons.

14. De Jong, T., & Van Joolingen, W. R. (1998). “Scientific Discovery Learning with Computer Simulations of Conceptual Domains.” *Review of Educational Research*, 68(2), 179-201.

15. Denning, P. (2009). “Beyond Computational Thinking.” *Communications of the ACM*, Vol. 52 No. 6, Pages 28-30.

16. Donovan, S. and Bransford, J. D. (2005). *How Students Learn.* The National Academies Press, Washington, D.C.

17. Grabinger, R. S. and Dunlap, J. C. (1995). “Rich environments for active learning: a definition,” *Association for Learning Technology,* **3** (2); 5-34.

18. Geometer’s Sketch Pad (GSP). http://www.dynamicgeometry.com.

19. Guskey, T.R. (2000) *Evaluating professional development,* Thousand Oaks, CA: Corwin.

20. Guskey, T. R. (2002). “Does It Make a Difference? Evaluating Professional Development.” *Redesigning Professional Development*, 59 (6); pp. 45-51.

21. Flick, L. and Bell, R. L. (2000). “Preparing tomorrow’s science teachers to use technology: guidelines for Science educators.” *Contemp. Issues Technol. Teach Educ.* 1: 39-60.

22. Hammond, L-D., Austin, K., Orcutt, S. and Rosso, J. (2001). “How People Learn: Introduction to Learning Theories. Stanford University. ”http://web.stanford.edu/class/ ed269/hplintrochapter.pdf.

23. Haring, N. G., Lovitt, T. C., Eaton, M. D. and Hansen, C. L. (1978). *The fourth R: Research in the classroom.* Charles E. Merrill Publishing Co.: Columbus, OH.

24. Harris, J., Grandgenett, N. and Hofer, M. (2010). “Testing a TPACK-based Technology Integration Assessment Rubric,” in D. Gibson & B. Dodge (Eds.), *Proceedings of Society for Information Technology & Teacher Education Int. Conference 2010* (pp. 3841-3848). Chesapeake, VA: AACE.

25. Hofer, M., Grandgenett, N., Harris, J., and Swan, K. (2011). “Testing a TPACK-Based Technology Integration Observation Instrument,” in M. Koehler and P. Mishra (Eds.). *Proceedings of Society for Information Tech. & Teacher Education Int. Conf. 2011* (pp. 4352-4359). Chesapeake, VA: AACE.

26. House Hearing (2003). *Implementation of the Math and Science Partnership Program: Views from the Field.* 108th Congress. https://www.gpo.gov/fdsys/pkg/CHRG-108hhrg90162/pdf/CHRG108hhrg 90162.pdf. Oct. 30, 2003.

27. Illeris, K. (2009). *Contemporary Theories of Learning.* Taylor & Francis Group: New York.

28. Koehler, M. J., and Mishra, P. (2008). "Introducing TPCK," in *Handbook of Technological Pedagogical Content Knowledge (TPCK) for Educators*, Routledge Press, New York & London.

29. Koehler, M. J., Shin, T.S., & Mishra, P. (2011). "How do we measure TPACK? Let me count the ways," in R. N. Ronau, C.R. Rakes, & M. L. Niess (Eds.). *Educational Technology, Teacher Knowledge, and Classroom Impact: A Research Handbook on Frameworks and Approaches*. Information Science Reference, Hershey PA.

30. Koffka, K. (1935). *Principles of Gestalt psychology*. New York: Harcourt, Brace & World.

31. Landau, R. (2006). "Computational Physics: A Better Model for Physics Education?" *IEEE Comp. in Sci & Eng.*, 8 (5), 22-30.

32. Little, L. (2003). "The computational science major at SUNY Brockport." *FGCS*, 19, pp. 1285-1292.

33. Loucks-Horsley, S., Stiles, K. E., Mundry, S., Love, N., Hewson. (2010). *Designing professional development for teachers of science & mathematics*. 3rd Edition, Thousand Oaks, CA: Corwin Press.

34. MacDonald, M. (2008) *Your Brain: The Missing Manual*. O'Reilly Media: Canada.

35. Margolis, J., Estrella, R., Goode, J., Holme, J. and Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge, MA: MIT Press.

36. Mishra, P., Koehler, M. J. (2006). "Technological pedagogical content knowledge: A framework for integrating technology in teacher knowledge." *Teachers College Record*, 108 (6), 1017-1054.

37. Montague, R. (2006). *How We Make Decisions.* Plume Books: New York.

38. Mooney, C. G. (2013). *An Introduction to Dewey, Montessori, Erikson, Piaget, and Vygotsky*. Redleaf Press: St. Paul, MN, 2013.

39. Nam, Y., Veronesi, P. and Yaşar, O. (2014a). "Preservice Science and Mathematics Teachers' Perceptions about Using Computational Modeling as an Instructional Method," *Proceedings of the Association for Science Teacher Education (ASTE) International Conference*, January 16-18, 2014, San Antonio, TX.

40. Nam, Y., Veronesi, P. and Yaşar, O. (2014b). "Computational Pedagogy for Preservice Science and Mathematics Teachers: Summer Professional Development Program," *EDULEARN14 6th International Conference on Education and New Learning Technologies*, July 7-9 2014, Barcelona, Spain.

41. NCES (2014). "Teacher Attrition and Mobility: Results From the 2012-13 Teacher Follow-up Survey." National Center for Education Statistics, Sept. 2014. Report No. NCES 2014077.

42. NRC (2012). *A framework for K-12 science education: practices, crosscutting concepts, and core ideas*. National Research Council Report. National Academies Press, Washington.

43. NGSS. Next Generation Science Standards. http://www.nextgenscience.org.

44. Niess, M. (2005). "Preparing teachers to teach science and mathematics with technology: Developing a technology

pedagogical content knowledge." *Teaching and Teacher Education*, 21, 509-523.

45. NSF Blue Ribbon Report (2006). *Simulation-Based Engineering Science: Revolutionizing Engineering through Simulation.* http://www.nsf.gov/pubs/reports/sbes_final_report.pdf.

46. Patton, M. Q. (2002). Qualitative research and evaluation methods. Thousand Oaks: Sage.

47. PITAC Report (2005). "Computational Science: Ensuring America's Competitiveness," http://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf.

48. Project Interactivate. http://www.shodor.org/interactivate/. Supported by NSF CISE funds.

49. Prince, M. J. and Felder, R. M. 2006. "Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases." *J. Engr. Education,* 95 (2); 123-138.

50. Prince, M. J. and Felder, R. M. 2007. "The Many Faces of Inductive Teaching and Learning." *Journal of College Science Teaching,* 36 (5); 14-20.

51. Reed, D. & Dongarra, J. (2015). "Exascale Computing and Big Data." *Comm. of the ACM,* 58 (7), pp. 56-68.

52. Repenning, A. 2012. "Programming Goes Back to School." *Comm. of the ACM*, 55 (5), 35-37.

53. Restak, R. (2001) *The Secret Life of the Brain.* The Dana Press: New York.

54. Rutten, N., van Joolingen, R., and van der Veen. (2012). "The Learning Effects of Computer Simulations in Science Education." *Computer & Education*, 58; 136-153.

55. Schmidt, D. A., Baran, E., Thompson, A., Mishra, P., Koehler, M., and Shin, T. (2009). Technological pedagogical content knowledge (TPACK): The development and validation of assessment instrument for pre-service teachers. *Journal of Research on Technology in Education*, 42, 123-149.

56. SIAM Report on Graduate CSE Education (2001), *SIAM Review*, 43, pp. 163-177.

57. SIAM Report on Undergraduate CSE Education (2007), SIAM Conference on CSE, www.siam.org/conferences, February 19-23, 2007, CA.

58. Smetana, L. K. and Bell, R. L. (2012). "Computer Simulations to Support Science Instruction and Learning: A critical review of the literature." *Int. J. Science Education*, 34 (9); 1337-1370.

59. STELLA. http://www.iseesystems.com.

60. Swanson, Charles. (2002). "A Survey of Computational Science Education." *http://cssvc.ecsu.edu/krell/Computational%20Science%20Education%20Survey%20Paper.htm.*

61. Tenenbaum, J. B., Kemp, C., Griffiths, T. L. & Goodman, N. D. (2011). "How to Grow a Mind: Statistics, Structure, and Abstraction." *Science*, 331, 1279-1285.

62. Turing, A.M. (1936). "On Computable Numbers, with an Application to the Entscheidungs problem." *Proceedings of the London Mathematical Society*. 2 (1937) 42: 230–265. doi:10.1112/plms/s2-42.1.230.

63. Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

64. Wieman, C. E., Adams, W. K. and Perkins, K. K. (2008). “PhET: Simulations That Enhance Learning.” *Science*, 332, 682-83.

65. Wigfield, A., Eccles, J. S., Schiefele, U., Roeser, R. W., Davis-Kean, P. (2006). “The development of achievement motivation.” in N. Eisenberg (Ed.), *Handbook of child psychology* (Vol. 3, 6th ed.). New York: Wiley.

66. Wing, J. M. 2006. “Computational Thinking,” *Comm. of the ACM*, Vol. 49, No. 3, 33-35.

67. Yaşar, O. (2016). “Cognitive Aspects of Computational Modeling and Simulation in Teaching and Learning.” *J. Computational Science Education,* Vol. 7, No. 1. pp. 2-14.

68. Yaşar, O. (2015). “A Universal Process: How Mind and Matter Seem to Work.” *Science Discovery,* 3 (6), 79-87.

69. Yaşar, O., Veronesi, P., Maliekal, J. and Little, L. (2015) “Computational Pedagogical Content Knowledge (CPACK),” in D. Slykhuis & G. Marks (Eds.), *Proceedings of Society for Information Technology & Teacher Education Conference 2015* (pp. 3514-3521).

70. Yaşar, O. (2014). “A Pedagogical Approach to Teaching Computing Principles in the Context of Modeling and Simulations.” *J. Computing Teachers*, Winter Issue. Vol. 11, pp. 15-26.

71. Yaşar, O. and Maliekal, J. (2014a). “Computational Pedagogy: A Modeling and Simulation Approach.” *IEEE Comp. in Sci & Eng*, 16 (3), 78-88.

72. Yaşar, O. and Maliekal, J. (2014b). “Computational Pedagogy Approach to STEM Teaching and Learning,” in M. Searson & M. Ochoa (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2014* (pp. 131-139). Chesapeake, VA: AACE.

73. Yaşar, O., J. Maliekal, L. little, and P. Veronesi. (2014). “An Interdisciplinary Approach to Professional Development for Math, Science, and Technology Teachers.” *Journal of Comp. in Mathematics and Science Teaching*, 33 (3), pp. 349-374.

74. Yaşar, O. (2013a). “Computational Math, Science, and Technology (C-MST) Approach to General Education Courses.” *J. Computational Science Education*, 4 (1), pp. 2-10.

75. Yaşar, O. (2013b). “Teaching Science through Computation.” *Science, Tech and Society*, Vol. 1 (1), pp. 9-18.

76. Yaşar, O. and R. Landau. (2003). “Elements of Computational Science and Engineering Education.” *SIAM Review*, 45, pp. 787-805.

77. Yaşar, O. (2001). “Computational Science Education: Standards, Learning Outcomes and Assessment.” *Lecture Notes in Computer Science*, 2073, 1159-1169.

78. Yaşar, O., Rajasethupathy, K., Tuzun, R., McCoy, A. and Harkin, J. (2000). “A New Perspective on Computational Science Education,” *IEEE Comp. in Sci. & Eng.*, **5** (2), 74-79.

79. Yaşar, P., Kashyap, S., and Roxanne, R. (2005). “Mathematical and Computational Tools to Observe Kepler’s Laws of Motion.” NSF MSPNET Library, http://hub.mspnet .og/index.cfm/14566.

80. Yaşar, P., Kashyap, S., and Taylor, C. (2006). “Limitations of the Accuracy of Numerical Integration and Simulation Technology.” NSF MSPNET Library. http://hub.mspnet.org/ index.cfm/14568.

## Biographical Information

Osman Yasar is an endowed professor and director of the CMST Institute at The College at Brockport, SUNY. He established the first undergraduate degree program in computational science in the United States and developed a computational pedagogical content knowledge (CPACK) framework for teacher professional development. His research interests include engineering and science education, computational pedagogy, computational theory of mind, fluid and particle dynamics, engine ignition modeling, and parallel computing. He has a Ph.D. in engineering physics and MS degrees in computer science and nuclear engineering from the University of Wisconsin–Madison. He also has BS and MS degrees in physics from Hacettepe University-Ankara. He co-founded a national supercomputer center and a doctoral program in computational science and engineering at Istanbul Technical University. In 2005, he was honored as one of the Top 25 national icons in his native homeland.

Peter Veronesi is program coordinator and lead faculty for the Adolescence Inclusive Science Education programs at The College at Brockport-SUNY.

Jose Maliekal is the Dean of the School of Science and Mathematics at SUNY College at Brockport.

Leigh Little is currently a member of the Earth Sciences Department at SUNY Brockport.

Sounthone Vattana obtained his bachelor's degree in Mathematics from SUNY College at Brockport, 1997, and master's degrees in Adolescent Education, Roberts Wesleyan College, 2003, and Computational Science, SUNY College at Brockport, 2005. He has worked as a 7-12 math and computer programming teacher and is currently an adjunct professor in the School of Science and Mathematics at the College at Brockport.

Ibrahim H. Yeter is currently a Ph.D. candidate in the Curriculum and Instruction program at the College of Education, and at the same time, he is pursuing his Master's degree in Petroleum Engineering at Texas Tech University. He is highly interested in conducting research within the Engineering Education frame- work.