Journal of Science Education and Technology (2020) 29:35–52
https://doi.org/10.1007/s10956-019-09795-7

Sociomathematical Norms for Integrating Coding and Modeling with Elementary Science: A Dialogical Approach

Amanda Catherine Dickes[1] · Amy Voss Farris[2] · Pratim Sengupta[3]

Published online: 28 January 2020

Abstract

In recent years, the field of education has challenged researchers and practitioners to incorporate computing as an essential focus of K-12 STEM education. Integrating computing within K-12 STEM supports learners of all ages in codeveloping and using computational thinking in existing curricular contexts alongside practices essential for developing mathematical and scientific expertise. In this paper, we present findings from a design-based, microgenetic study in which an agent-based programming and computational modeling platform—ViMAP—was integrated with existing elementary science and math curricula through lessons co-designed and taught by the classroom teacher across a period of seven months. We present a *dialogical* re-positioning of coding, where disciplinarily grounded meanings of code emerge through the construction of computational utterances—i.e., computer models as well as complementary conversations and physical models that serve as mathematical and scientific explanations—through the use of *socio-mathematical norms*.

Keywords Coding · Modeling · Elementary science · Computational thinking · Sociomathematical norms · Science education

Introduction

Conceptual development in science is inseparably intertwined with the development of epistemic and representational practices such as modeling (Nersessian 2008; Pickering 1995). The implication for science education is that science educators should focus not only on conceptual ideas but also on supporting the development of modeling as the practice through which these ideas emerge (Lehrer and Schauble 2006). Recent years have also seen an emphasis on integrating *computational thinking* (Wing 2006)—an analytic problem solving and design approach fundamental to computing—with K-12 STEM classrooms (Grover and Pea 2013; Ilic et al. 2018; NGSS 2013; Sengupta et al. 2013; Weintrop et al. 2016).

✉ Amanda Catherine Dickes
adickes@gmri.org

[1] Gulf of Maine Research Institute, Portland, ME, USA

[2] The Pennsylvania State University, University Park, Pennsylvania, USA

[3] University of Calgary - Werklund School of Education, Calgary, AB, Canada

Although the term *computational thinking* was only coined in 2006, the push to integrate computer programming and modeling in K-12 science is certainly not new. Early research on Logo (Harel and Papert 1991; Papert 1980), Boxer (diSessa et al. 1991; Sherin et al. 1993), and the Envisioning Machine (Roschelle and Teasley 1995) are all examples of efforts to accomplish objectives largely similar to Wing's call for democratizing opportunities for thinking like computer scientists (Wing 2006), albeit in domains such as mathematics and physics. An important finding that emerged from this body of work, however, is that curricular integration of computing within K-12 STEM contexts is a complex and challenging endeavor for both teachers and students that involves the introduction and adoption of new literacies (e.g., programming) alongside disciplinary ideas that students already find challenging to understand (Guzdial 2006; Sherin et al. 1993). This challenge is echoed in more recent studies that have identified the importance of paying attention to, and scaffolding for students' conceptual difficulties in understanding computational abstractions (e.g., initialization, algorithms) as well as scientific domains (e.g., biology and physics) in middle grades (Basu et al. 2016).

One response to these challenges has been the design and development of low-threshold, high ceiling programming languages which utilize highly supported, simple block interfaces which lower the threshold for novice programmers in learning introductory programming. These include languages such as Scratch (Resnick et al. 2009), StarLogo NOVA (2014), ViMAP (Sengupta et al. 2015; Sengupta and Wright 2010), and CTSiM (Basu et al. 2013). Research investigating how these tools are taken up by learners both in and out of school has demonstrated the importance of situating computing within familiar and accessible disciplinary contexts (Sengupta et al. 2013; Wilkerson-Jerde et al. 2015); designing activity systems which integrate computing with complementary forms of activity (Dickes et al. 2016; Peppler et al. 2010); supporting a community of practice around computing (Brennan and Resnick 2012; Kafai and Burke 2013); providing rich computational environments for students to express agency in using, creating, and modifying with computational artifacts (Dickes and Farris 2019; Farris and Sengupta 2016; Lee et al. 2011, 2014); and paying close attention to curricular content and practices alongside student knowledge and interests (Wilkerson-Jerde et al. 2015).

Grounded in a phenomenological approach (Sengupta et al. 2018), our work aims to extend this growing body of research by investigating how computing can be integrated within existing elementary science curriculum over an extended period of time (7 months) through lessons *co-designed* and *taught* by the classroom teacher, who herself had no prior experience with programming. While phenomenological approaches to integrating coding in K-12 science highlight the heterogeneity inherent in K-12 classrooms in terms of the importance of different forms of representations and modeling (Sengupta et al. 2018), it is also necessary to focus on integration across these forms, similar to Pickering's notion of an interactive stabilization (Pickering 1995) of machinic performance and representational heterogeneity. As we explain in the following section, this in turn is a fundamentally Bakhtinian reframing of coding, where the emphasis shifts from cognitivist visions of mastery over computational abstractions to the dialogical co-construction of computational utterances (Sengupta et al. in press). Specifically, in this paper, we advance an argument that the classroom teacher's emphasis on *mathematizing* and *measurement* as key forms of learning activities helped to meaningfully integrate programming as an integral component of STEM work in the classroom in a manner that also helped her connect two new literacies—computational thinking and modeling—with her regular mathematics and science curricular needs. We illustrate how the classroom teacher, Emma (pseudonym), in partnership with researchers, found and created opportunities to integrate agent-based programming with her regular science and mathematics curricula by iteratively developing *sociomathematical norms* (Cobb et al. 1992; McClain and Cobb 2001; Yackel and Cobb 1996) for modeling motion.

Framing Computational Modeling Dialogically in the Science Classroom: Model-Eliciting Activities and Sociomathematical Norms

The overarching perspective that we present in this paper frames computational thinking and modeling as *model-eliciting activities* (Lesh and Doerr 2003) in the science classroom. As Lesh and Doerr (2003) point out, model-eliciting activities usually involve mathematizing—"by quantifying, dimensionalizing, coordinatizing, categorizing, algebratizing, and systematizing relevant objects, relationships, actions, patterns, and regularities" (*ibid* p. 5). In such activities, the "products" of student work include descriptions, explanations, justifications, or constructions, which also usually need to be sharable, transportable, or reusable (Lesh and Doerr 2003). One could argue that through developing such normative modeling practices, students can also develop meta-modeling knowledge (Schwarz and White 2005)—i.e., knowledge about the nature and purpose of scientific work—because, they may learn to identify abstract representations as models and demonstrate how models can be used to predict and explain scientific phenomena.

Underlying this framing is the deep synergy between computational thinking and modeling and the science as modeling perspective (Sengupta et al. 2013; Weintrop et al. 2016). Phenomenologically, as Sengupta et al. (2018) have argued, computational thinking involves both representational and epistemic work that are also grounded disciplinarily, materially, and socially. While Wing (2006) has argued that the notion of computational abstractions is central to understanding and developing computational thinking, adopting a "practice" lens implies that abstractions are inseparable from practices, in the same way that epistemic growth in the sciences has been shown to be deeply intertwined with representational growth (Pickering 1995). To this end, Sengupta et al. (2018) have argued that computational thinking and modeling could be understood as practices that are fundamental to computing and computer science, as well as in the sciences. Some examples are problem representation, abstraction, decomposition, simulation, verification, and prediction. These practices are central to modeling, reasoning, and problem-solving in a large number of scientific, engineering, and mathematical disciplines (NRC 2007; NGSS 2013).

The particular form of model-eliciting activities that we focus on in this paper involves the iterative development and refinement of collective (i.e., classroom-level), *normative* modeling practices (Lehrer et al. 2008; McClain and Cobb 2001). Science educators have shown that the question of what counts as a "good" model also needs to be normatively established in classroom instruction in order to deepen students' engagement with scientific modeling in elementary grades, and that these norms follow similar shifts toward deeper disciplinary warrants over time (Ford and Forman

2006; Lehrer and Schauble 2006; Lehrer et al. 2008). Lehrer and colleagues (2000, 2001, 2006, 2008) demonstrated that authentic epistemic work in the science classroom must develop and deepen through the social construction of scientific knowledge, and highlight mathematics as a meaning-making lens through which the natural world can be systematized and described (Lehrer et al. 2001). Furthermore, an emphasis on measurement, including aspects of measurement such as precision and error, and normatively guided model refinement help students move beyond a focus on superficial features of the target phenomena to modeling "unseen" relationships between variables and underlying mechanisms (Lehrer and Schauble 2000, 2006; Lehrer et al. 2008).

The specific genre of norms we focus on in this paper have been termed *sociomathematical norms* (Cobb et al. 1992; McClain and Cobb 2001; Yackel and Cobb 1996). Sociomathematical norms differ from general social norms that constitute classroom participation structure in that they concern the normative aspects of classroom actions and interactions that are specifically mathematical, emerge through interaction with a mathematical object (such as a programming environment) and are given social value by the practicing community. These norms regulate classroom discourse and influence the learning opportunities that arise for both the students and the teacher. In the work of Cobb and his colleagues, teachers initiate and guide the development of social norms in mathematics classrooms that sustain classroom microcultures characterized by explanation, justification, and argumentation (Cobb et al. 1989; Yackel et al. 1991). In our context, we believe that focusing on sociomathematical norms can lead to a fundamentally *dialogical* (Bakhtin 1983) re-positioning of coding (Sengupta et al. in press), where the meaning of code emerges through the construction of *computational utterances*. These utterances include computer models as well as complementary conversations and physical models that serve as mathematical and scientific explanations. In a deeply Bakhtinian sense, these *utterances* (Bakhtin, 1983; Todorov 1984) are at once individual voicings, as well as negotiated culturally through sociomathematical norms. Furthermore, similar to Cobb and colleagues, our focus is on the perspective of the teacher, who initiated these norms on her own accord, without any prompting by the researchers.

An important and fundamental sociomathematical norm is concerned with what counts as an *acceptable* mathematical solution within a particular community. This norm typically originates as a *socially* defined norm, and shifts over time to a *sociomathematically* defined norm (Yackel and Cobb 1996). However, given that these norms are often teacher-initiated, it is also important to look at how these ideas and opportunities are taken up by students in their work (Cobb et al. 2009). In such contexts, the "value" of different forms of models—inscriptions, verbal explanations, computer models, etc.—is socially established in terms of disciplinarily valued practices.

Researcher-Practitioner Partnerships: an Integrative Stance

Our focus in this paper is to study how computational modeling and programming were integrated as part of typical STEM work *from the perspective of the classroom teacher*. Overall, research on integrating computational modeling and programing with K-12 science curriculum has been largely interventionist in nature (diSessa et al. 1991; Sengupta et al. 2013; Wilkerson-Jerde et al. 2015). In contrast, our work here takes an *integrative* stance, where our role as researchers was largely limited to designing activities in partnership with the teacher, and guided by what the teacher wanted to accomplish on a day to day basis. Rather than focusing solely on the outcomes of such a partnership, we trace the impact of the partnership itself (Coburn and Penuel 2016), by presenting how a new literacy—computation—was successfully integrated and taken up within typical instruction. Schools are complex organizational structures (Coburn et al. 2009; Spillane 1998), and our teacher's local knowledge (Geertz 1983)—in the form of institutional pressure and her students' progress in their regular curriculum—shaped her framing of programming within the classroom as mathematization, with a particular emphasis on multiplicative reasoning, geometry, and graphing. We believe that such forms of researcher-teacher partnership (Coburn et al. 2013), where teachers exercise significant agency in the direction and co-design of the curricular activities and lead the classroom teaching and implementation, are methodologically crucial for addressing the issue of effectively managing the tradeoff between teaching programming and teaching science.

Research Questions

We propose that emphasizing mathematizing and measurement as key forms of learning activities can help teachers meaningfully integrate programming as a part of STEM instruction, and further, that teachers can accomplish this by supporting the development of sociomathematical norms for assessing the "goodness" of computational models. Specifically, we are interested in how the classroom teacher's actions and interactions with an agent-based programming tool impacted how computing was taken up and integrated within existing classroom practice. To this end, we investigate the following research questions

1. How did the teacher's emphasis on mathematizing and measurement support the development of sociomathematical norms around model "goodness," and how were those norms taken up by the students?
2. Did these norms shape in any way the development of students' computational models and computational thinking? If so, how?

Method

Setting and Participants

School and Students This study was conducted over the course of 7 months in a 3rd grade classroom in a 99% African-American public charter school located in a large metropolitan school district in the southeastern United States. The authors had worked with this school for 3 years prior to the start of the study. The school serves a population of approximately 245 students (K-4); greater than 95% of the student population is eligible for free or reduced lunch. Fifteen third grade students—14 African-American and one Latino—participated in this study, of which eight were female and seven were male. According to achievement levels as determined by the state achievment test in the previous year (student level is characterized as advanced, proficient, basic, or below basic) 86% of students in the class were within the "basic" or "below basic" ranges in mathematics and 64% of students were within "basic" or "below basic" ranges in science. Informed consent, including permission to video and audio record, was obtained from legal guardians prior to beginning the study, and verbal assent was obtained from each student.

Teacher The classroom teacher, Emma, was a first-year teacher in the school district. Emma had previously worked as an elementary school teacher in a metropolitan school district in a neighboring state and had a master's degree in Education. In early Fall of 2013, the authors met with the school principal to discuss establishing a year-long research partnership with one 3rd grade and one 4th grade teacher in the school. The principal shared the research goals, timeframe, and participation requirements with the 3rd and 4th grade teachers, and Emma expressed an interest in participating. An in-person meeting was held in September of 2013 between the authors and Emma to formalize the partnership.

Researcher-Teacher Partnership The role of the teacher, as both co-designer and instructor of content, was an emergent focus on this work. Prior to data collection and implementation, the authors and Emma met several times in September and October of 2013 to introduce Emma to the ViMAP programming and modeling language and demonstrate its capability with regard to modeling geometric shapes, motion, and inheritance of traits. A 7-day sequence of shape-drawing tasks (designed by the authors) was shared with Emma, who subsequently co-taught the sequence with the authors during the first week of the study. Emma made two critical pedagogical moves during these early shape drawing tasks which altered the trajectory of the research in a significant way. We discuss these pedagogical moves below.

First, during the first week of the research Emma approached the lead author and requested that curricular activities incorporate relevant third grade standards in mathematics including measurement, reasoning with data, and math processes. Emma had observed her students using mathematical reasoning, particularly multiplicative reasoning, to program shapes in ViMAP and she wanted to expand ViMAP's pedagogical impact by making more explicit connections to required 3rd grade mathematics concepts. She then shared a lesson on regular polygons she had planned to teach the following day and discussed with the lead author ways that ViMAP could support this lesson. Out of this conversation, a sequence of activities investigating and modeling mathematical relationships with regular polygons, congruent shapes, and perimeter were co-designed between the lead author and Emma (Phase I). Emma provided the topic for each lesson, and brainstormed with the lead author ways students could model the mathematical content of each lesson using ViMAP. Second, and equally important, Emma took over the role of lead (and often sole) instructor in the classroom. She continued to serve in this role for the remainder of the school year.

Emma met with the authors again in January of 2014 to discuss possible learning activities for the Spring semester. During this meeting, a series of activities in Kinematics (Phase II) and Ecology (Phase III) were brainstormed, co-designed, and sequenced. Emma was an equal co-designer with the authors and, as before, guided the design of activities which supported content standards in mathematics and science, particularly learning goals related to measurement, data, interdependence, and motion. Emma's designs often emphasized opportunities for students to directly experience a phenomenon through embodiment (e.g., leaving footprints on banner paper) and the development of classroom wide conventions. As a novice programmer herself with no prior experience in programming, Emma saw physical enactment of computational commands as a valuable form of sensemaking and encouraged activity design which scaffolded student thinking in similar ways.

Any lesson modifications were discussed during weekly conferences between the lead author and Emma. These changes were made based on Emma's formal and informal assessments of student understanding of the material or in-the-moment responses to student ideas. These adjustments often took the form of extending instructional time on a topic and modifying the designed classroom materials to better meet the mandated instructional goals.

Materials and Measures

The Learning Activities The learning activities are divided into three phases: Geometry (Phase I), Kinematics (Phase II), and Ecology (Phase III). The present paper reports on Phase II, Kinematics, and traces the development of normative mathematical criteria for what counts as "good" ViMAP models of motion. Instruction during Phase II blended complementary forms of modeling, including embodied, physical and computational, which discretized motion into "steps" to scaffold understanding of motion as a process of continuous change (e.g., aggregating "steps" to produce constant motion). The learning activities were also designed to support the invention and interpretation of mathematical measures of distance and provide opportunities for Emma to reframe programming as a mathematization of motion. Table 1 summarizes the learning activities during Phase II.

The Programming Environment We used ViMAP (Sengupta et al. 2015), an agent-based, visual programming language that uses the NetLogo modeling platform as its simulation engine (Wilensky 1999). In ViMAP, users construct programs using a drag-and-drop interface to control the behaviors of one or more computational agents. The ViMAP command library (Table 2) includes domain-specific and domain-general commands. In addition to assigning agent-level attributes (e.g., step size, heading, color), ViMAP also supports modeling increasing or decreasing functions and writing procedures. A measure command block allows users to design mathematical measures based on periodic measurements of agent-specific and aggregate-level variables (e.g., speed or number of agents) which are then displayed in a separate graphing window.

For the present paper, students interacted with a single-agent and two-agent version of ViMAP. In terms of computational thinking, goals for Phase II included distinguishing between setup and go procedures, debugging programs, and use of the repeat control block and the place measure point command block to generate accurate graphs of distance over time. As shown in Fig. 1, placing measure commands required students to think carefully about the sequencing of command blocks. Variables such as color and pen width can also be used to visually discretize motion and communicate selected actions of the turtle agent.

Data Sources Data for this work comes from informal interviews with the participants, video recordings of class activities and discussion, student artifacts (e.g., student representations, activity sheets, ViMAP models, and pre-, mid, and post-assessments), and daily field notes. The lead author and Emma conducted informal interviews with six focal students during opportune moments while the students were engaged in single, pair, or small group work around modeling and representational activities. Focal students were selected to offer insight into typical student experience and were representative of the class based on race, gender, and performance. In addition, focal students had high attendance rates and were present for a majority of the implementation.

Informal student interviews typically took two forms: First, interviews were conducted after the student had called upon the teacher or lead author to help him or her with a difficulty. Second, interviews were conducted to ask students to explain their reasoning regarding their programming decisions and their interpretations of model outcomes. These interviews were not structured and were variable in length up to approximately ten minutes. Focal students were typically interviewed two or three times during an activity.

Student ViMAP models were saved and a screen capture script was utilized to record screen captures at 30-s intervals on all student computers. A pre-assessment was administered on November 5, 2013, and a post-assessment was administered on May 22, 2014. A midterm assessment was administered at the conclusion of Phase II on April 10, 2014.

Data Analysis

Using the methodology of grounded theory (Glaser and Strauss 1967), the raw data for Phase II (all video data, screen captures, and student work) was closely examined by the lead author to uncover any broad themes and categories, and descriptive codes were generated. Activity during Phase II was categorized into three episodes of activity: (1) inventing measures, (2) defining approximations, and (3) generating predictions. Significant teaching moments during each episode were transcribed by the lead author. The lead author and third author then conducted a secondary analysis to uncover representative themes across the three episodes of activity. The lead author and the third author carefully re-watched the interview

Table 1 Summary of learning activities during Phase II

Activity	Description
Leaving footprints	Students leave ink footprints on banner paper.
Generating measures	Students iteratively develop, apply, test and refine a measurement of distance termed a "step size."
Collecting step-size data	Students use the "step size" measurement convention to measure their personal step sizes.
Modeling step-sizes in ViMAP	Students model and refine their "step-sizes" in ViMAP, generate and discuss "total-distance" graphs and make predictions using ViMAP's grapher.
Modeling motion as a process of continuous change	Students model motion scenarios in ViMAP and check the validity of those models using ViMAP's grapher and the total distance equation.

Table 2 ViMAP's command library

Movement Blocks	Measure Blocks	Drawing Blocks	Control Blocks
Set step size	Place measure point	Set [variable] equal to [number]	If /Then/Otherwise
Set heading	Clear measure points	Pen down	
Right	Start over measuring	Pen up	Repeat [number]
Left	Label [variable]	Stamp	
Go forward Go backward		Go invisible Go visible	If [variable] less than/greater than [variable]
Set xy		Change shape to	
Set random heading [0] to [360] Set [variable] plus/minus [number]		Set [variable] plus/minus [number] Set [variable] equal to [variable]	If [variable] less than/greater than [number]
Set [variable] equal to [variable]			

videos and re-read the transcripts to generate pattern codes (Huberman and Miles 1994) to characterize the data into more meaningful units. This analysis revealed the emergence of sociomathematical norms around measuring, describing (approximation), and extending (prediction) data and the take up of those norms in student work.

Student ViMAP models were also coded along several dimensions to characterize growth in computational thinking. These codes include (1) evidence of design-based thinking, (2) distinguishing between setup and go procedures, (3) use of variables, (4) use of the place-measure-point command, and (5) use of the repeat control block. These codes were identified by the lead and third author who coded all student Phase II models along each dimension. The second author was then asked to code a subset of the data using the codes identified by the lead and third author. The second author's analysis of student ViMAP models was above 85% agreement with the analysis performed by the lead and third authors. The rubric used to measure change in student's computational thinking is provided in Table 3.

We present the analysis in the form of explanatory case studies (Yin 1994), which are well suited as a methodology to answer *how* and *why* questions. We find this to be good fit because our goal here is to illustrate the *process* through which the classroom developed sociomathematical norms, which includes answering *how* the development of these norms shaped the students' interactions with ViMAP and other modeling experiences, and *why* these norms were deemed useful by the teacher. Following previous studies (Dickes et al. 2016), our selection and analysis of cases were guided by the following two criteria: representativeness and typicality.

Representativeness implies that the selected cases should aptly represent key aspects of learning experienced by the students. These key aspects or themes, in turn, are defined based on the research questions. For our purposes, representativeness implies that each case should highlight an important aspect of the process through which the relevant sociomathematical norm emerged. In our paper, this includes chronicling the emergence of each sociomathematical norm through the eyes of Emma and her students. *Typicality* implies that the selected case(s) should potentially offer insights that are likely to have wider relevance for the remainder of the participants in the study. In other words, the cases selected should represent aspects of the process of learning experienced by a majority of the student population. For the emergence of each norm, we also present evidence of take-up of the norm in the work of one of our focal students, whose participation was identified as typical, and therefore generalizable, to that of the majority of the class. Therefore, in our analysis below, we present (a) excerpts from classroom discussion showing the emergence of each norm and (b) take-up of those norms in focal students' work. Finally, to answer our second research question, we also present a classroom-level analysis

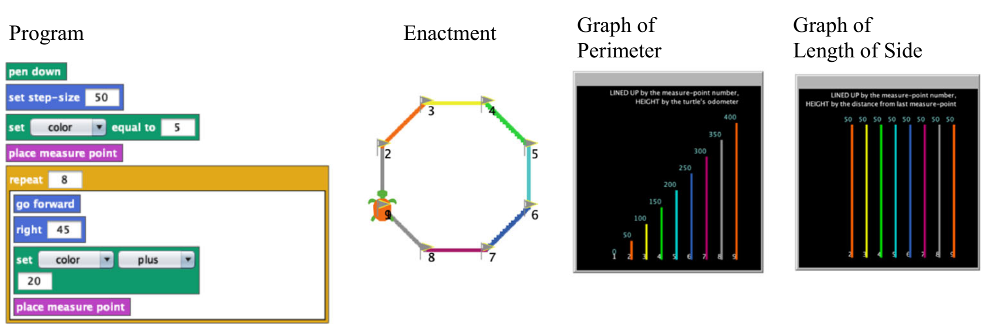


Fig. 1 Example of a program for generating a regular octagon, the enactment by the turtle agent and resulting graph of line segment and perimeter

Table 3 Rubric used to assess growth in computational thinking

Design-based thinking • Identifies bugs and attempts to fix them • Iteratively refines models (not just completes them)	0 Never identifies/fixes bugs or iteratively refining their models.	1 Rarely *[1–2 debugging events or major refinements (shifts)]*	2 Sometimes *[3–5 debugging events or major refinements (shifts)]*	3 Always or almost always *[6 or more debugging events or major refinements (shifts)]*
Use of loops is systematic/productive		Rarely	Sometimes	Always or almost always
• Number of repeats reflects either time or other pertinent aspects of the data	Number of repeats reflects either time or other pertinent aspects of the data	1	2	3
• Identifies when it is useful to use a repeat	Identifies when it is useful to use a repeat	1	2	3
• Avoids use of repeat when it is not parsimonious/or does not reflect data	Avoids use of repeat when it is not parsimonious/or does not reflect data	1	2	3
Manipulates step-size and measure commands appropriately in order to control attributes of the enactment and/or graphs		Rarely	Sometimes	Always or almost always
• Manipulates step-size in ways that reflect constant acceleration or energy	Step-sizes reflect speed (e.g., pattern of constant acceleration) or energy	1	2	3
• Uses measure-points at appropriate places to show change in graphs	Uses measure-points at appropriate places to show change in graphs	1	3	3
Setup vs. go [distinct, appropriate]		Rarely	Sometimes	Always or almost always
• Attends to initial speed (or initial energy) of the ball (the step-size in the setup procedure), by making it 0. (this may not show up until 02.13.13)	Attends to initial conditions	1	2	3
• Distinct use of set-up and go (If there is no repeat loop, it is okay to have set-step-size and pen-down in go, as long as it is at the beginning of the code)	Distinct and appropriate code in setup and go.	1	2	3
Exploration of Variable Space		Rarely	Sometimes	Always or almost always
• Controls pen-width, pen-up, or pen-down to show something relevant to the model	Color	1	2	3
• Controls color to show something relevant to the model	Pen width, pen up, pen down	1	2	3
• Controls direction and turns to show something relevant to the model	Turns	1	2	3

of students' ViMAP models in terms of the quality of their code.

Findings

Inventing Measures: Movement from Social to Sociomathematical

Instruction during Phase II began with an investigation of animal tracks. Using the richly illustrated children's book *Wild Tracks! A Guide to Nature's Footprints* (Arnovsky 2008), Emma and her students discuss how animal footprints are data-laden. Among the ideas offered by students include animal tracks as histories of "where [the animal] started [moving] and where [the animal] stopped" and whether or not the animal was "running or walking." Each of these ideas emphasize footprints as *measurable objects*. In particular, they leverage footprints as sources of data on the rate and distance traveled by an agent, ideas that will frame all student investigations during Phase II.

Students then generated an embodied artifact, their own footprints inked onto a strip of banner paper and, guided by Emma, problematized the idea of a *step size*. What is a *step size* and if we were to measure one, where would we begin measuring and where would we end? Students offer three options for measuring a step size: (1) heel-to-toe, (2) heel-to-heel, and (3) toe-to-toe (Fig. 2). At this stage in their motion investigations, selection of the "best" (Emma's word) measurement convention was primarily a social endeavor. Students defined the "best" step-size measure based on a majority (> 50%) class vote, ultimately selecting the heel-to-toe measurement convention because, as shared by students, it was "the biggest" or because their "friend voted for it." This is evidence of what Cobb and colleagues have termed a socially defined mathematical norm—that is, the normatively accepted rationale for deciding which mathematical measure is the most appropriate results from a social decision without mathematical warrants (Yackel and Cobb 1996).

This was troubling but not unfamiliar to Emma, as she noted in a weekly debrief with the authors. Students would often take decisions based on social interactions rather than disciplinary warrants in her classroom. In class, however, Emma took a different approach; rather than challenging the students' decisions, she encouraged the social negotiation that the students engaged in, telling her students that ultimately "[they] had to make the decision" and that there was no "right or wrong" convention provided they used the measure consistently. Emma's shift away from the social was toward the individual—that is, she instructed the students to bring their attention back to their own work, with the idea that it might be easier for students to develop a deeper mathematical grounding of their own measures, rather than trying to think about how others may have designed their measures.

Emma instructs her students to return to their own footprints and measure their unique step-sizes using their popularly selected heel-to-toe measurement convention. She then asks them to add up their recorded step-sizes to generate a total distance traveled. Finally, Emma instructs her students to measure, with yardsticks or measuring tape, the total distances they actually walked on their footprint artifacts (i.e., the straight-line distance between the first and the final step). In a conversation with the authors while students were engaged in the activity of measuring their unique step-sizes, Emma explained why she wanted students to generate two

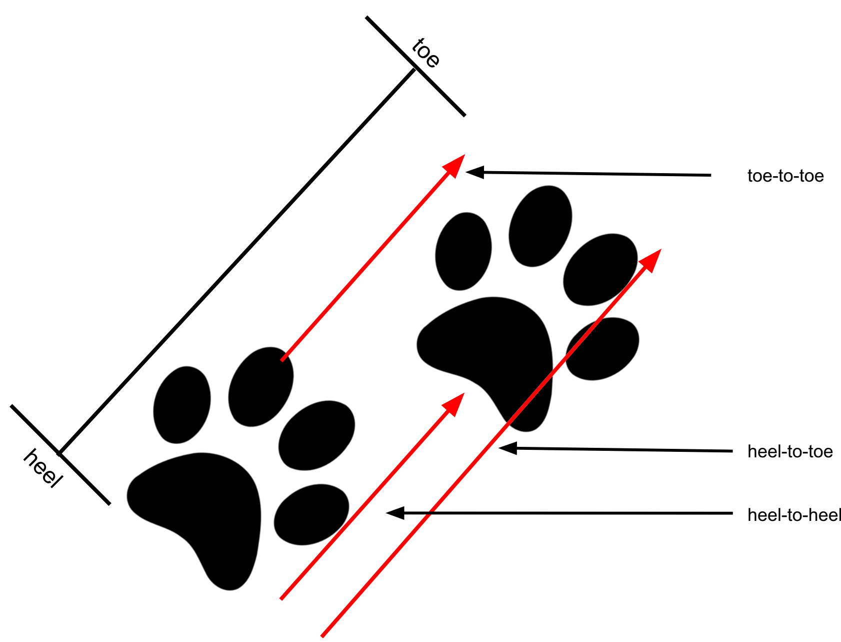


Fig. 2 Reconstruction of a whiteboard diagram representing student ideas on how to measure a "step-size"

measures of total distance. Emma's focus during these initial lessons was to break down continuous motion into a series of discrete steps, which could then be re-aggregated through either embodied (walking) or computational modeling (ViMAP). Emma recognized that the convention her students had selected was not, to use her words, an "accurate" measure of total distance traveled because it produced an overlap (Fig. 3), effectively measuring portions of the distance twice. For Emma, an "accurate" measure communicated exactly how much distance was traveled. Upon completing the activity, students publicly reported their findings and found that the measured (ruler on footprint artifact) and the calculated distance (adding individual step sizes using the heel-to-toe measurement convention) didn't match when they had predicted that they would (Fig. 3).

Students were surprised by the mismatch between the two measures. Emma suggested to the class that "maybe [they] need[ed] to find a measure that [was] more mathematically accurate." It is important to remind ourselves here that Emma's use of the word "accurate" is her own, and not grounded in the literature in math or statistics education. In conversations with Emma, and during observations in class, we (the authors) noted that accurate was a term Emma used inconsistently. In the context of her teaching, accurate could indicate precision (measures and approximations), fidelity (models and graphs), or simply "correct." Emma emphasized "accuracy" throughout her instruction, to the extent that students identified "accuracy" as *the* criteria for model quality near the end of Phase II and into Phase III (we discuss this development in more detail in the Discussion). As researchers, we elected not to intervene at this moment to suggest a more appropriate term; rather, our intention was to see where Emma took the conversation next. Furthermore, our decision to not intervene was also based on the interventionist bias in the field of educational computing we pointed out earlier, where teachers' voices have traditionally been ignored in favor of researcher-driven implementations.

Emma's next move was to resolve the tension students had uncovered between their measured and calculated total distances. Turning to the class, she asks her students to explain to her why knowing their own step size was "useful." Students suggested that step-sizes help them "know how far you went", indicating that they had begun to recognize step-sizes as measures of distance traveled. However, students continued to disagree on where a step-size should begin. Two students offered competing definitions: (1) measurement begins at the heel of the first foot and (2) measurement begins at the toe of the first foot. Emma reminds the class of the question they are trying to answer ("how far someone went") and invites two students to the front of the classroom to model forward movement. Emma explains that these students are competing in a race, and draws a starting line and a finish line on the floor using a dry-erase marker. The two students line up at the starting line and race (by taking steps), until one has crossed the finish line. This demonstration is repeated once more, after which Emma faces the class and asks which of the two proposed measurement conventions they think accurately measures total distance traveled, and *why*. One student, Marvin, explains to the class that he observed forward movement of the racing students beginning at the toe, and therefore suggests

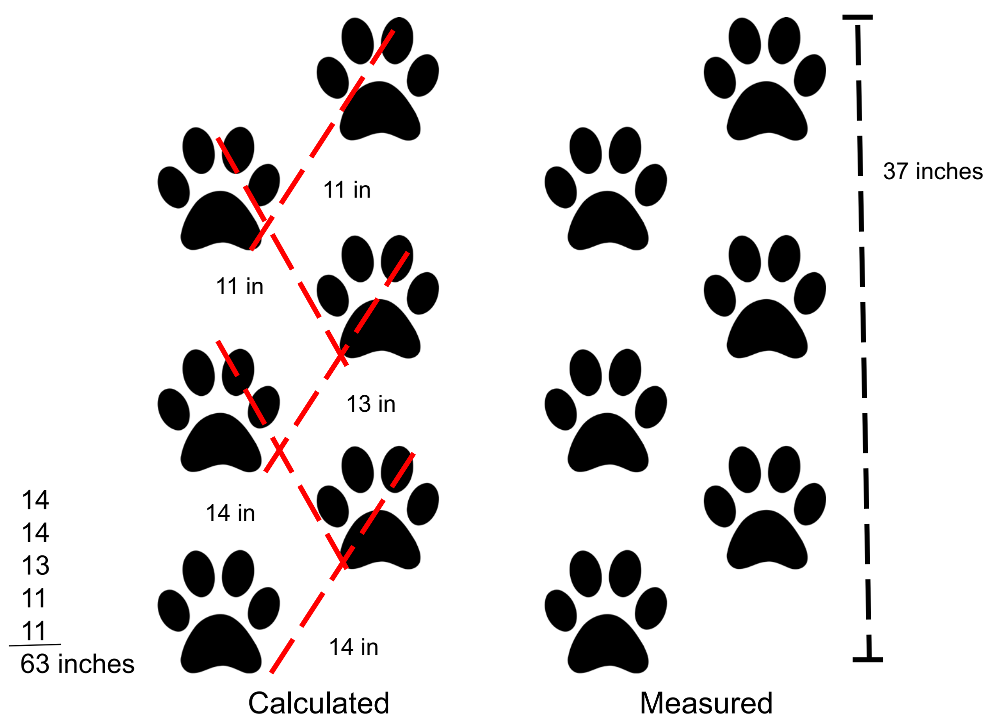


Fig. 3 Reconstruction of a whiteboard diagram representing the mismatch in distance traveled caused by the popular-vote heel-to-toe *step size* measurement convention

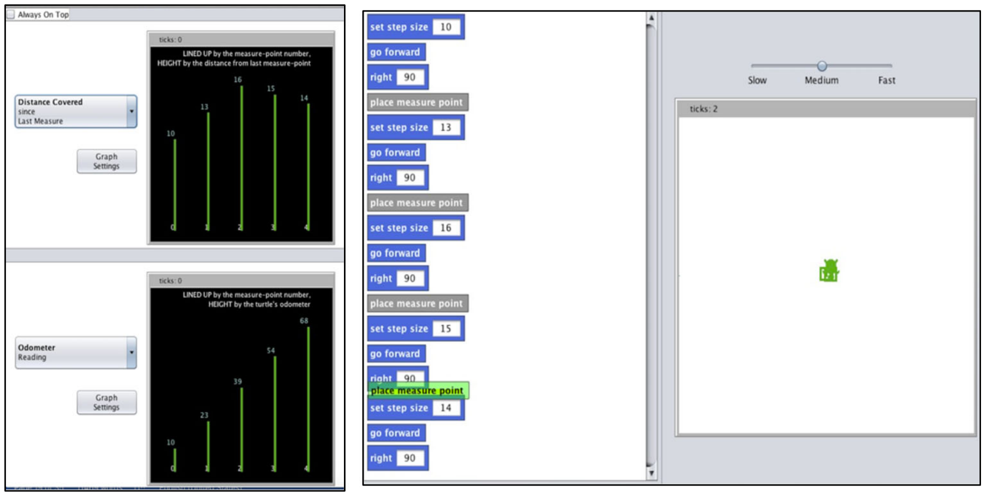


Fig. 4 Example student ViMAP model using the revised toe-to-toe measurement convention. The student is inserting the "place measure point" command to produce graphs of individual step size (top left) and total distance (bottom left).

that measurement should begin at the toe because "at the toe you *begin to go forward*." Marvin cited as evidence the fact that the racing students had aligned their toes—and not their heels—with the starting line. Emma invites the class to respond to Marvin's claim. Marvin's classmates agree with him, and, adding on, share that the race was won when the toe of the winning student crossed the finish line. At the conclusion of this discussion, students had taken up what Emma called the "mathematically accurate" toe-to-toe measurement convention shown in Fig. 2. Grounded in their own noticings and a new sociomathematical norm for qualifying "good" measurements, the students re-measure their step-sizes using the toe-to-toe convention. These revised measures were then modeled in ViMAP, an example of which is shown in Fig. 4.

What is notable in this episode is that the development of criteria for what counts as a "good" measure largely followed the characteristics outlined by Yackel and Cobb (1996). First, similar to Yackel and Cobb, it is important to note that the norm is teacher-initiated. That is, at the heart of this work was Emma's suggestion of emphasizing the *accuracy* of measures and her push to represent continuous movement in discrete steps. Second, similar to Yackel and Cobb (1996), initially, what counted as "good" measures were socially defined, and shifted toward deeper mathematical grounding over time. The initial choice of the heel-to-toe convention was based solely on popular vote, with students deciding how to measure step-sizes for reasons entirely unrelated to the purpose of the measure. Following attempts to use the heel-to-toe convention to measure total distance, the criteria for what counted as a "good" measure shifted towards greater mathematical warrants. Emma played an important role in this shift by orchestrating opportunities for normative social definitions of measures to be placed at odds with student's own measurements. In her instruction, Emma made explicit attempts to highlight the epistemic value of measures, and to make those measures personally meaningful to students by connecting those measurements to embodied experiences grounded explicitly in mathematical work. The experience of walking, viewed through the lens of measurement, was now a mathematical problem of measuring the total distance traveled during walking.

One might then ask: what is the relevance of this work in light of students' computational modeling and programming? One connection that is worth pointing out here is Emma's emphasis on representing continuity in terms of discrete measures, and her overall emphasis on measurement as a learning activity. Furthermore, the analysis we presented illustrates how a teacher-initiated socio-mathematical norm of what counts as a good measure can support students in doing this work, and serves as an essential precursor for grounding the students' computational utterances in disciplinary contexts. In the next section, we will see how Emma weaves student work with ViMAP with physical modeling and measurement activities through the introduction of two new sociomathematical norms.

Approximation and Prediction: Norms for Model Refinement

Approximation In conversations with the authors, Emma noted that a "step-size" was different for each student, and - wanting to make student work with ViMAP "concrete" (Emma's term) and *useful* - Emma designed a series of activities exploring "approximation" and "prediction" (Emma's terms) as ways to "quickly" model distances that "could not be walked" (Emma's phrase). Emma introduced approximations as *representative* values—i.e., values that are, to use Emma's words, "kind of real" and "helpful" because they can be used to make predictions about the total distance traveled without having to measure each step. Emma shared with the authors earlier in the year she had attempted to teach *averages*, but observed her students experiencing difficulty with ideas of typicality and representativeness. Emma noted to the authors that she saw students' step-size work as an opportunity to reintroduce the idea in a more concrete way. To do so, Emma asks her students to consider how they might describe their step-size data with a single number, as opposed to reporting the size of each step as a different number. She frames this thought experiement as a measurement problem, asking the class to imagine how they might "know how far [someone] traveled after fifty or one hundred steps" without having to "actually walk and add up all those different step -sizes."

For the next half hour, students in the class struggled with the idea of representative values. Students were heavily grounded in their embodied movement of counting and measuring each step, and were unable to think about the problem outside of this context. Turning to the lead author, Emma said that she needed to think of a way to frame the question "in words [the students] will understand." Addressing the entire class, she asks students to report what they notice while she physically enacts the step-size data collected by a student. One student, Damien, points out that each step is "changing." Emma asks Damien if each step is changing "by a lot" or "by a little" and encourages students to re-examine their own data ("look at [your own] data sheets"). Damien responds that his steps "mostly change by a little." Another student, Jayla, agrees with Damien, stating that she "walked mostly the same" when she measured her individual step sizes. Emma then mimes walking with a dramatically different step-size per step, and, laughing, students agree with Damien and Jayla's noticing that their steps are "about the same size." Emma asks students to think about what their next step-size might be if they had continued walking. One student, Keenan, suggests that unknown steps would be "close to" the same size as known steps—but not necessarily the "actual" step size.

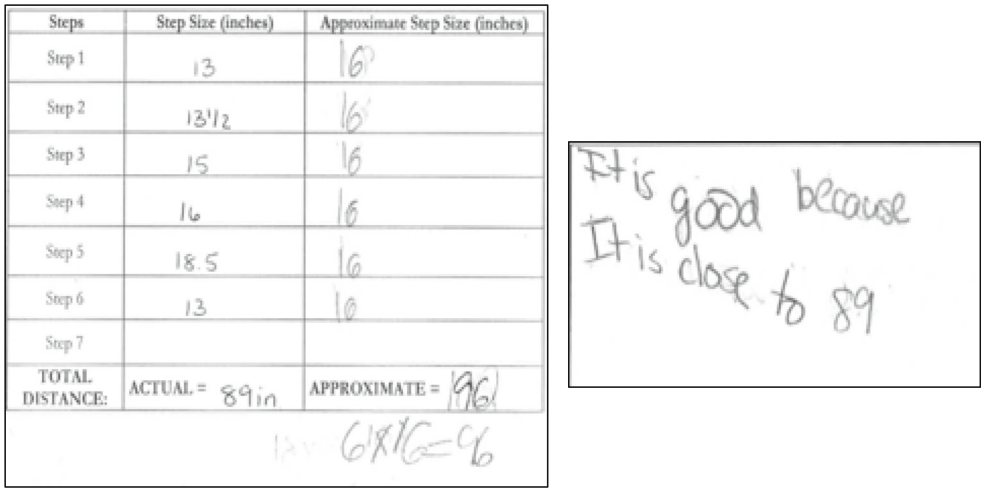
The class is now at the brink of converging on a new understanding of an *approximate measure* as a measure that is "close to the actual, but not exact" (class definition of approximate values). Emma's next move involved engaging students in a modeling activity where they put this definition to use. She provides students with a hypothetical data set (step sizes of 11, 9, 11, and 12), and asks them to build a ViMAP model of the total distance traveled based on the general pattern of the step-sizes. To facilitate this, she asks students to reason about the following: If the hypothetical student continued walking what would their next step be? In a flurry of discussion, students contributed their ideas. Fourteen out of fifteen students (93%) agreed that a good "future step," as referred to by the students, was any value already within the range of the set of empirical data, i.e., a value of 9, 10, 11, or 12. One student in particular offered that *11* was the best choice because it appeared "the most times" and was "in the middle" of the data set. Only one student deviated from the other students, suggesting that *13* was the logical next step since it "continued the pattern" established by the final two steps of 11 and 12. This deviance from the emerging norm of a "good" approximation was addressed by Emma by referring back to the shared classroom definition of approximation: *close to actual, but not exact.* She models in ViMAP an approximate step size of *13* and asks the class to consider the total distance traveled in each model: 43 using actual step size values and 53 using an approximate value of *13*. One student notices that the approximate total distance was "too far away from the number [the person] actually walked." The class agrees that the two distances are not "close" and comes to a consensus that "good" approximate values are close to the actual value in terms of both individual step-sizes *and* total distance traveled.

Take-up in Student work The work of one of our focal students, Marvin, demonstrates the take-up of the norm that approximate values are "close to the actual, but not exact." In the example below, Marvin originally selected an approximate step size value of eighteen inches.

When the lead author asked Marvin why he selected an approximate value of 18, Marvin responded that he chose that value because it was the upper limit of his data range ("because it's my biggest number"). Following the discussion depicted in the classroom level analysis above, Marvin changes his approximate value from 18 to 16 to make his approximate total distance closer to his actual total distance of 89 inches.

Marvin's work in Fig. 5 illustrates his change in thinking. Marvin changed his approximate step value from 18 to 16 (note the erasure marks in the table on the left), reducing his approximate total distance traveled from 108 inches to 96 inches. Marvin explains in writing why he changed his approximate value from 18 to 16, commenting that his total distance of 96 inches using an approximate step-size of 16 is close to his actual distance of 89 inches. In an informal interview with the lead author, Marvin explained that he made this change because he could "get closer to his actual distance" with a value of 16 rather than 18 inches. In other words, his approximate value was more *accurate* in terms of how well it represented his larger data set.

Prediction Emma saw student work with approximate values as an opportunity to deeply engage her class with mandated mathematics curriculum such as multiplcative reasoning and math processes. In attempting to do so, she next led a class discussion on calculating approximate total distances. She presented a hypothetical data set where a student moved forward by a step-size of 8 units over 6 steps, and asks her students to use their lessons from math class on repeated addition and multiplication to calculate the approximate total distance. Several students explain that you can use repeated addition ($8 + 8 + 8 + 8 + 8 + 8$) or multiplication (8×6) to quickly solve for the total



Steps	Step Size (inches)	Approximate Step Size (inches)
Step 1	13	16
Step 2	13 1/2	16
Step 3	15	16
Step 4	16	16
Step 5	18.5	16
Step 6	13	16
Step 7		
TOTAL DISTANCE:	ACTUAL = 89in	APPROXIMATE = 96

6×16=96

It is good because It is close to 89

Fig. 5 Marvin's refined approximate step-size value and mathematical rationale

distance traveled. Emma then asks her students how they might find the total distance traveled without "using numbers." Antione responds that, to find the total distance, you multiply the number of steps by the approximate step size. Moving to the whiteboard, Emma writes down: *Total Distance* = *Number of Steps* × *Approximate Step Size*. Emma explains that what the students had just discovered was a *formula* which not only calculates known total distances, but could also *predict* unknown future distances.

Take-up in Student Work In the excerpt shown below, Angelo, a focal student, interprets the formula as a means to both "win a bet" as well as mathematically verify the accuracy of his ViMAP model of distance (Table 4).

Angelo comments in lines 4 and 5 that if someone bet him that he could only travel less than or equal to 100 units of distance, he would *know* that statement to be false. The researcher affirms Angelo's observation, asking him if he could prove an acquaintance wrong if he knew his approximate step size (lines 6, 7, 8, and 9). Angelo responds in line 10 that he could. When asked by the researcher how he could prove them wrong (line 11), he offers two possible solutions: the graphs he had generated in his ViMAP model (shown in Fig. 6) and the class formula (line 14). Epistemologically, this is a significant move. As Angelo put it, using approximate values allows him to "know" (line 4). We believe that Angelo's explanation of "betting" and "knowing" here is his intuitive way of explaining what prediction is. Furthermore, this demonstrates that Angelo is able to mathematically summarize discrete values to model continuous patterns of change.

Table 4 Angelo's prediction

	Utterance	Line
Researcher	How far did you walk after taking 15 steps?	1
Angelo	300 distance	2
Researcher	That's exactly right.	3
Angelo	So, if somebody bet that I won't make it farther than 100 *I know* that I will	4
	make it.	5
Researcher	That's right. That's how a formula for approximate distance can help you.	6
	If someone said "I bet Angelo would only walk 150 inches in 15 steps",	7
	But you knew what your approximate step size was, could you prove them	8
	wrong?	9
Angelo	Yes	10
Researcher	How?	11
Angelo	I could look at my graph.	12
Researcher	Or you could do what?	13
Angelo	I could use a calculator. Fifteen times 20 equals 300.	14

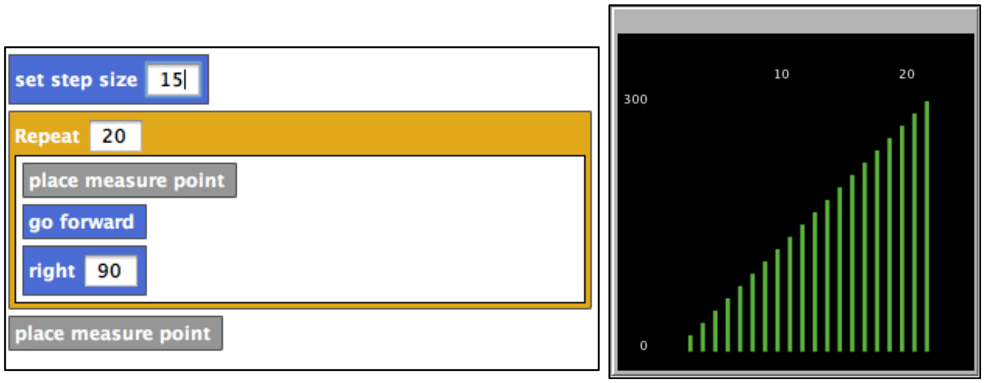


Fig. 6 Angelo's ViMAP model

Further into Prediction: Generalizing Motion Using a Multiplicative Scheme in ViMAP

Near the end of Phase II, Emma and the researchers wanted to extend the thinking students had done on developing predictive models of motion into more generalizable mathematical forms. Emma recognized that the formula for finding total distance derived by the class (*Number of Steps* × *Approximate Step Size*) was a specialized form of a multiplicative scheme that also serves as a rate equation: *Distance* = *Speed* × *Time*. She told the researchers that she considered students ViMAP work to be a rich context for engaging her students in multiplicative reasoning, and turning to the class, she explains that the formula they had discovered was "powerful", and could be used to analyze many real-world situations. She then introduces a "real world" problem, in which students were asked to determine which of two cars, Car 1 or Car 2, traveled further in a four-hour period: Car 1 which traveled at a speed of 45 mph for 3 hours, or Car 2 which traveled at a speed of 35 mph for 4 hours. A sample student solution to the two-car problem is shown in Fig. 7. As students share their ViMAP models at the front of the class, we can observe that all of the students are able to produce ViMAP models using appropriate and non-redundant variables. Additionally, multiplicative reasoning is evident in students' use of *repeat* and *step-size*, as shown in Fig. 7, where car 1 travels 3 (repeat) × 45 (step-size) units, and car 2 travels 4 (repeat) × 35 (step-size) units.

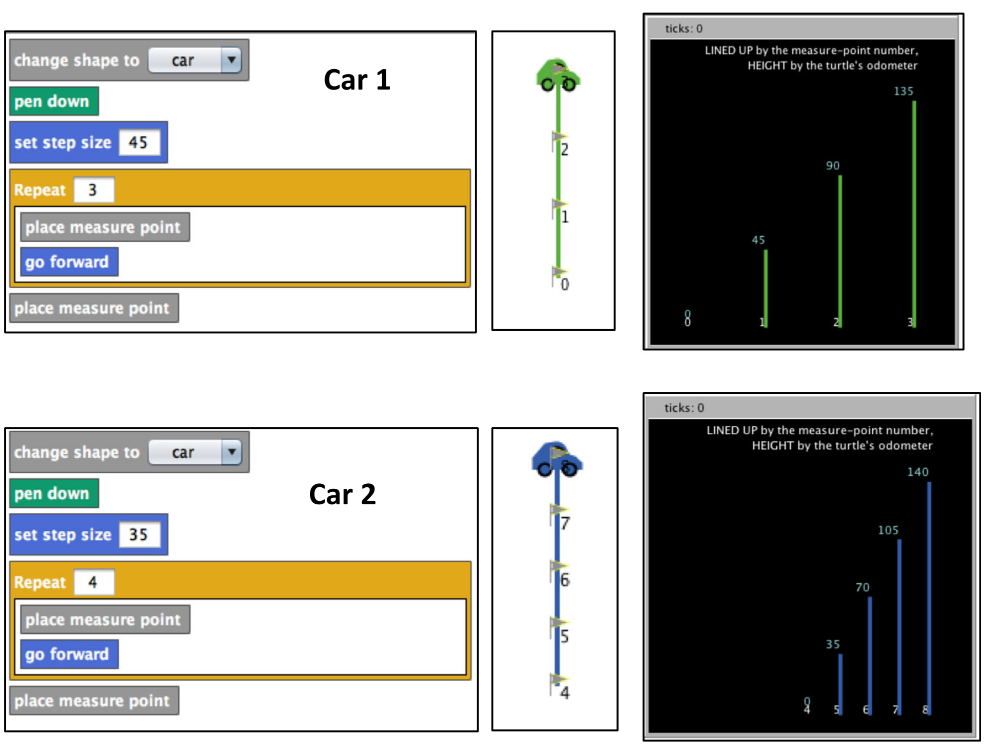


Fig. 7 A student's solution to the two-car problem using ViMAP

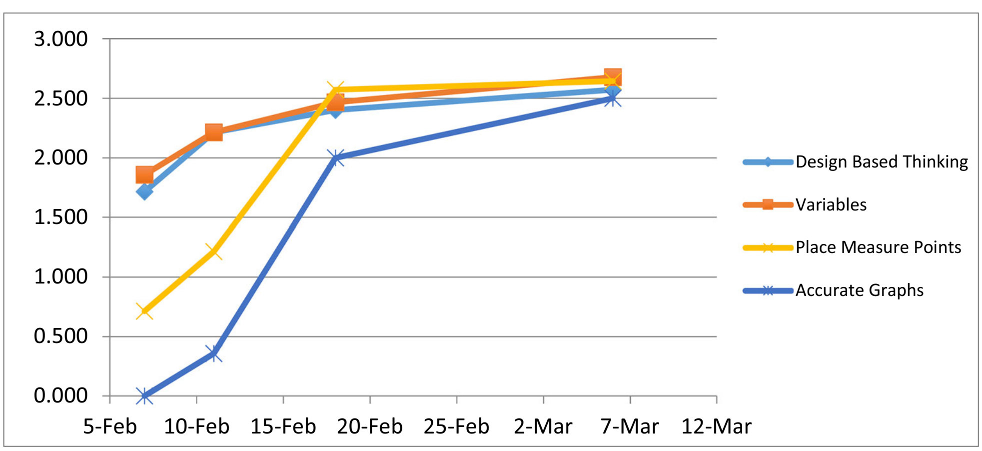
Co-development of Sociomathematical Norms and Computational Thinking

Our analysis shows that across the class, there was an increase in students' ability to compose ViMAP models that represented their data with greater fidelity. This included appropriately initializing their ViMAP models and using computational abstractions such as variables and loops with progressively fewer errors. Student models early in Phase II were characterized by missing commands (e.g. "set step size"), duplicate and/or additive commands (e.g. two "go forwards" placed in sequence), missing *setup* procedures, and - at times - haphazard placement of command blocks (e.g. unecessary repeat blocks). But most importantly, these early programs were rarely representative a student's lived data. In contrast, as Phase II progressed students' ViMAP models grew in complexity and representativeness following classwide investigations of *approximation* and *prediction*. Specifically, our analysis demonstrates that the accuracy of graphs in students' later models represented skillful use of the "repeat" and "place measure point" command, which in turn relied on a conceptual understanding of when to initialize the measurement, and how often the measurement had to be repeated in order to generate the desired graph. For example, consider the following two programs composed by the same student within the span of a single lesson: Repeat 15 [place measure point, set step size 20, go forward, right 90] and Repeat 15 [set step size 20, go forward, right 90, place measure point]. In the first program, the student has placed the measure command first within the repeat block. As the program executes, the last step of the turtle is not measured resulting in a total distance graph of 240. The student notices this mistake when they check the accuracy of their ViMAP model by comparing the total distance represented in their ViMAP graph (240) with the total distance calculated using their specialized rate equation ($15 \times 20 = 300$). They subsequently correct their programming error by relocating the place measure command to the bottom of the repeat loop. While this change may look small, it is indicative that the student understood sequencing within ViMAP, the order of execution of commands placed within a repeat loop and when to initialize measurement to generate accurate graphs of the target phenomenon.

The growth in students' overall computational fluency is evident in Fig. 8, which shows how students' use of the ViMAP programming commands became increasingly sophisticated as they held their models accountable to the sociomathematical norms throughout the duration of the activities reported here (Phase II). This is an apt example of how students' computational utterances became progressively more mathematically grounded, through the ongoing negotiation with sociomathematical norms. We scored each students' ViMAP model across Phase II using the rubric we designed for this research, paying particular attention to students' use of appropriate variables in their ViMAP code, whether their graphs represented appropriate element(s) of the phenomenon being simulated, and the degree to which students iteratively refined their models and successfully debugged them. Overall, student fluency with ViMAP increased dramatically following the series of lessons on *approximation*. Our further analysis indicates that design-based thinking was positively correlated with variable exploration (r-squared, 0.603), and that the use of the place measure command was positively correlated with exploration of the variable space (r-squared, 0.6454). This finding is particularly intriguing because it suggests that computational practices (variables) were co-developing alongside scientific and mathematical practices (place measure point and design-based thinking), further validating the argument that computing *reflexively* develops (Kafai and Harel 1991) alongside STEM practices.

Finally, our analysis has also found that our sociomathematical norms are linked to changes in students' computational practices such that with the genesis of each norm, new computational modeling practices are added (Fig 9). In general, students use of the ViMAP graphing tool, their use of repeat as a measure for both number of steps and time, and their use of variables

Fig. 8 Growth in students' computational fluency



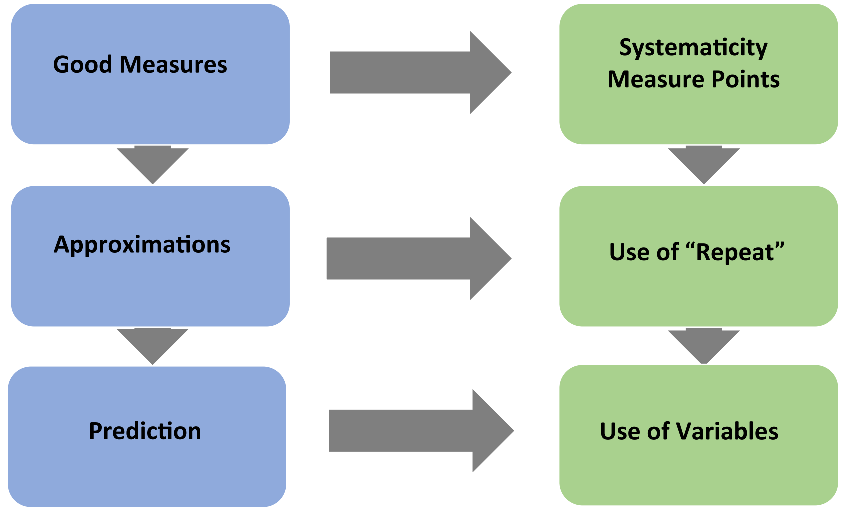


Fig. 9 Co-development of sociomathematical norms and computational practices

increase in sophistication as they iteratively generate predictive models of motion in ViMAP.

Why did this improvement happen? We believe that the illustrative cases we presented show that the development, deployment and refinement of sociomathematical norms led to iterative improvement in the quality of students' models as progressively more *authentic* representations of the phenomena they were modeling. Emma was an integral participant in this refinement, evident in her push for *accuracy* which was often taken up by students in their modeling work, and became a disposition that was "taken as shared" (Cobb et al. 1992) by the classroom community. The push for designing "approximate" measures, alongside an emphasis on multiplicative reasoning led students to develop predictive models in ViMAP. Students' multiplicative reasoning was evident in their use of ViMAP programming blocks, i.e., in their use of loops and agent-level variables (No. of Repeats × Step size), as well as a more careful attention to the design of graphs.

We also believe that Emma's emphasis on physical and embodied modeling as a way to complement computational modeling and thinking played an important role in the take-up of each norm by the students. Cobb and colleagues have argued that sociomathematical norms pertaining to what counts as an acceptable mathematical explanation and justification typically are interpretable in terms of actions on mathematical objects that are *experientially real* to the students, rather than in terms of procedural instructions (Cobb et al. 1992). In our study, by emphasizing embodied modeling as a way to mathematize motion, the teacher facilitated the students' take-up of norms pertaining to what counts as a "good" model of motion by making ViMAP commands such as "step-size" experientially real to the students. The construction and refinement of students' ViMAP models and explanations - their computational utterances - is thus inextricably intertwined with the *experientially real* forms of modeling using step-sizes. Furthermore, the teacher's focus on the measure command in ViMAP as a way to "see" individual steps enacted by the ViMAP turtle helped students to discretely represent the motion of the turtle agent and correctly interpret the resultant graphs. At the end of Phase II, the communicative nature of graphs was privileged, with students remarking that novices unfamiliar with ViMAP as a modeling tool would be unable to interpret turtle enactment as it relates to the phenomena, remarking that if novices "look at the [enactment] it wouldn't be understandable, but if they look at the graph, they will *know*." This is the form of representational and disciplinary heterogeneity that is fundamental to a *dialogical* (Bakhtin 1983) reframing of coding, particularly in K-12 STEM classrooms (Sengupta et al. in press).

Discussion

Sociomathematical Norms Can Integrate Computational Thinking and Science

Our study highlights the *reflexive* relationship between computational thinking, scientific modeling, and mathematical thinking when agent-based programming is the computational medium. While this has been noted previously in researcher-led studies (Kafai and Harel 1991; Papert 1980; Sengupta et al. 2013), our work here shows that teachers with no background in computing can integrate programming with their existing science curricula by reframing programming as mathematization—in particular, designing measures of change. Furthermore, our study also shows that using agent-based programming as the means to develop these models of change can be supported by the teacher by developing sociomathematical norms around the mathematical quality of these models. As we argued earlier, this is a fundamentally dialogical (Bakhtin 1983) re-positioning of coding (Sengupta et al. in press). It is through this disciplinary re-positioning that coding becomes modeling in the science classroom, as individual voicings of students' ViMAP code become meaningful as computational utterances (Bakhtin 1983; Todorov 1984) through their negotiation with sociomathematical norms.

Pragmatically, such forms of integration can be truly synergistic for the K-12 science classroom. The focus on constructing mathematical measures can serve as a unifying force across the representational and disciplinary heterogeneity that is at the heart of Bakhtinian dialogicality (Bakhtin 1983; Sengupta, et al. in press), as well as phenomenologically grounded (Sengupta, et al. 2018) as evident in Emma's classroom. As we have reported elsewhere (Sengupta et al. 2015), interpreting and constructing mathematical measures (for example, units of measurement and graphs) is a commonly experienced difficulty for students in science classes. Manipulating units is emphasized in statewide standardized assessments, securing its status as an important learning goal to teachers and school leaders. Agent-based programming can help students overcome these challenges as learners need to define events in the form of discrete mathematical measures. By engaging in iterative cycles of building, sharing, refining, and verifying these models, students refine their understanding of what actions and interactions of agents represent an "event," which are then displayed on graphs of change over time as the simulation "runs". This provides students with the opportunity to explore different kinds of measures of change and connect them meaningfully with the simulation. Teacher initiated *sociomathematical* norms, such as the ones reported in this study, when taken up in student work through joint action, can help students harness and deploy the epistemic and representational power of agent-based computing as a "language" for doing science. New literacies such as computational modeling and programming can thus be meaningfully and seamlessly negotiated with day-to-day needs in the science classroom.

Methodological Concerns: Teacher Voice and Conceptual Dissonance in Researcher-Teacher Partnerships

Design-based researchers have recently begun advocating for greater teacher voice and agency in research studies, which in turn reframes studies as researcher-teacher partnerships (Severance et al. 2016). Our study is certainly an example where teacher voice often led the direction of research; but it also raises an important methodological and epistemological question: how should we address conceptual dissonances between the researchers and the teachers? For example, in our study, the teacher's framing of "accuracy"—i.e., students' models must be "mathematically accurate"—was largely based on her intuitive conceptualization of the term. Let us now imagine answering this question as educational researchers and epistemologists. "Accuracy" will take on a very different meaning, and perhaps will even have a negative connotation—because an essential characteristic of models, according to the epistemologists of science, is that they are incomplete. In fact, a few months later, the teacher did introduce the notion of incompleteness (albeit in her own language, and in a different context)—in Phase III, while modeling ecological interdependence. The notion of accuracy, though, lingers throughout the academic year.

We will take up this issue in more detail in a different paper. But we do want to raise the following question here: what should we as partnering researchers do in such situations? Should we have intervened and coached the teachers about the professional vision of scientists and epistemologists about accuracy and incompleteness of models? This study is an example where we did not intervene to bridge conceptual dissonance on this issue. This indeterminacy, we posit, is central to the heterogeneity that is essential for viewing coding in classrooms as dialogical (Sengupta et al. in press). The meaning of words that appear disciplinarily profound to researchers must be negotiated with teachers who may have different and colloquial perspectives pertaining to the word. The word, after all, in a deeply Bakhtinian sense, is only half ours, as it is populated with intentions of others (Bakhtin 1983). We must maintain or establish teachers' position such that they have agentive roles and intentionality in shaping the use of disciplinary words and their meanings, and this is particularly true in contexts of educational computing. We believe that researchers must fundamentally work toward positioning teachers as *directors* in research-practice partnerships—rather than at an "equal" footing with the researcher. An equitable partnership may not be one in which everyone has equal say. Instead, an equitable partnership in educational computing research must seek to support teachers in voicing (and re-voicing) computation from their own perspectives, with curricular mandates and classroom constraints in mind. The story of socio-mathematical norms that we have presented here is also the story of a teacher re-voicing code in her science and math classroom.

Unfortunately, researchers in educational computing—in particular, researchers who design and implement programming languages for children—have traditionally not engaged with the issue of curricular integration *from the perspective of K-12 teachers*. Research studies in this field (including some of our earlier work), therefore, largely carry out a strong interventionist agenda where teacher voice is often overshadowed by the researchers. In contrast, we have come to see the K-12 public school classroom as a complex, interdependent system, where teachers, students, curricula, and curricular mandates must all be considered alongside one another, especially if we set out to integrate any new literacy and/or technology with the classroom. So, if our goal is to make programming and computational modeling ubiquitous in the K-12 science classroom, we posit that researchers and designers of programming languages for the K-12 classrooms must learn to see the world through the eyes of the teachers, especially when it involves conceptual dissonance between researchers and teachers. It is through carefully studying the unfolding of such dissonances over longer periods of time (i.e., not a short intervention study), especially when teachers are working with new technologies and literacies (such as programming and computational modeling), that we (as researchers) will learn to design technological and activity systems that will be aligned with the perspectives of the teachers, and therefore, have a greater chance of becoming a mainstay in their classrooms.

Acknowledgments The authors gratefully acknowledge the contributions of Cherifa Ghassoul, Emily Brutcher and Leah Embry.

Funding Information This project was partially supported by US National Science Foundation (NSF CAREER Award #1150230) and funds from the Imperial Foundation (Canada).

Compliance with Ethical Standards All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. Informed consent was obtained from all individual participants included in the study.

Conflict of Interest The authors declare that they have no conflicts of interest.

Disclaimer All opinions expressed in this paper are of the authors' and are not endorsed by the National Science Foundation or the Imperial Oil Company.

References

Arnovsky, J. (2008). Wild tracks!: A guide to nature[s footprints. New York, NY: Sterling Children's Books.

Bakhtin, M. M. (1983). *The dialogic imagination: Four essays*. Austin, TX: University of texas Press.

Basu, S., Dickes, A., Kinnebrew, J. S., Sengupta, P., & Biswas, G. (2013). CTSiM: A computational thinking environment for learning science through simulation and modeling. In *CSEDU* (pp. 369–378).

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.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (pp. 1–25).

Cobb, P., Yackel, E., & Wood, T. (1989). Young children's emotional acts while engaged in mathematical problem solving. In *Affect and mathematical problem solving* (pp. 117–148). New York, NY: Springer.

Cobb, P., Wood, T., Yackel, E., & McNeal, B. (1992). Characteristics of classroom mathematics traditions: An interactional analysis. *American Educational Research Journal, 29*(3), 573–604.

Cobb, P., Gresalfi, M., & Hodge, L. L. (2009). A design research perspective on the identities that students are developing in mathematics classrooms. *Transformation of knowledge through classroom interaction*, 223–243.

Coburn, C. E., Toure, J., & Yamashita, M. (2009). Evidence, interpretation, and persuasion: Instructional decision making at the district central office. *Teachers College Record, 111*(4), 1115–1161.

Coburn, C. E., Penuel, W. R., & Geil, K. E. (2013). *Research-practice partnerships: A strategy for leveraging research for educational improvement in school districts*. New York: William T. Grant Foundation.

Coburn, C. E., & Penuel, W. R. (2016). Research–practice partnerships in education: Outcomes, dynamics, and open questions. *Educational Researcher, 45*(1), 48–54.

Dickes, A. & Farris, A. (2019). Beyond Isolated Competencies: Development of Computational Literacy in an Elementary Science Classroom. In: Sengupta, P., Shanahan, M.-C., & Kim, B. (Eds). Critical, Transdisciplinary and Embodied Approaches in STEM Education . (pp 131 - 150). Springer: New York.

Dickes, A., Sengupta, P., Farris, A. V., & Basu, S. (2016). Development of mechanistic reasoning and multi-level explanations in 3rd grade biology using multi-agent based models. *Science Education, 100*(4), 734–776.

diSessa, A., Hammer, D., Sherin, B., & Kolpakowski, T. (1991). Inventing graphing: Children's meta-representational expertise. *The Journal of Mathematical Behavior, 10*(2), 117–160.

Farris, A. V., & Sengupta, P. (2016). Democratizing children's computation: Learning computational science as aesthetic experience. *Educational Theory, 66*(1-2), 279–296.

Ford, M. J., & Forman, E. A. (2006). Redefining disciplinary learning in classroom contexts. *Review of Research in Education, 30*, 1–32.

Geertz, C. (1983). *Local knowledge: Further essays in interpretive anthropology* (Vol. 5110). New York: Basic Books.

Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldire.

Guzdial, M. (2006). Software-Realized Scaffolding to Facilitate Programming for Science Learning. *Interactive Learning Environments, 4*(1), 001–044.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43.

Harel, I. E., & Papert, S. E. (1991). Constructionism. Ablex Publishing. di Sessa, A. A. (1991). An overview of Boxer. Journal of Mathematical Behavior, 10(1), 3-15.

Huberman, M. A., & Miles, M. B. (1994). Data management and analysis methods. In N. Denzin & Y. S. Lincoln (Eds.), Handbook of Qualitative Research. Thousand Oaks, CA: Sage.

Ilic, U., Haseski, H. İ., & Tugtekin, U. (2018). Publication Trends Over 10 Years of Computational Thinking Research. *Contemporary Educational Technology, 9*(2), 131–153.

Kafai, Y., & Harel, I. (1991). Children learning through consulting: When mathematical ideas, knowledge of programming and design, and playful discourse are intertwined. *Constructionism*, 110–140.

Kafai, Y. B., & Burke, Q. (2013). The social turn in K-12 programming: moving from computational thinking to computational participation. In *Proceeding of the 44th ACM technical symposium on computer science education* (pp. 603–608). ACM.

Kafai, Y., & Harel, I. (1991). Learning through design and teaching: Exploring social and collaborative aspects of constructionsm.

Lead States, N. G. S. S. (2013). *Next generation science standards: for states, by states*. Washington, DC: National Academies Press www.nextgenscience.org/nextgeneration-science-standards.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., et al. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 32–37.

Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K–8 curriculum. *ACM Inroads, 5*(4), 64–71.

Lehrer, R., & Schauble, L. (2000). Developing model-based reasoning in mathematics and science. *Journal of Applied Developmental Psychology, 21*(1), 39–48.

Lehrer, R., & Schauble, L. (2006). *Cultivating Model-Based Reasoning in Science Education*. Cambridge: Cambridge University Press.

Lehrer, R., Schauble, L., Strom, D., & Pligge, M. (2001). Similarity of form and substance: Modeling material kind. *Cognition and instruction: Twenty-five years of progress*, (pp. 39–74).

Lehrer, R., Schauble, L., & Lucas, D. (2008). Supporting development of the epistemology of inquiry. *Cognitive Development, 23*(4), 512–529.

Lesh, R. A., & Doerr, H. M. (2003). *Beyond constructivism: Models and modeling perspectives on mathematics problem solving, learning, and teaching*. Abingdon: Routledge.

McClain, K., & Cobb, P. (2001). An analysis of development of sociomathematical norms in one first-grade classroom. *Journal for Research in Mathematics Education, 32*(3), 236–266.

NRC (National Research Council). (2007). *Tools and methods for estimating populations at risk from natural disasters and complex humanitarian crises. Report by the National Academy of Sciences* (p. 264). Washington, DC: National Academy Press.

Nersessian, N. (2008). Model-based reasoning in scientific practice. In Teaching scientific inquiry. *Brill Sense*, 57–79.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc..

Peppler, K., Danish, J., Zaitlen, B., Glosson, D., Jacobs, A., & Phelps, D. (2010). BeeSim: leveraging wearable computers in participatory simulations with young children. In *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 246–249). ACM.

Pickering, A. (1995). *The mangle of practice: Time, agency, and science*. Chicago: University of Chicago Press.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67.

Roschelle, J., & Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning* (pp. 69–97). Berlin, Heidelberg: Springer.

Schwarz, C. V., & White, B. Y. (2005). Metamodeling knowledge: Developing students' understanding of scientific modeling. *Cognition and Instruction, 23*(2), 165–205.

Sengupta, P., & Wright, M. (2010). ViMAP. In *Computer software*. Nashville: Mind, Matter & Media Lab, Vanderbilt University.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351–380.

Sengupta, P., Dickes, A., Farris, A. V., Karan, A., Martin, D., & Wright, M. (2015). Education: Programming in K-12 science classrooms. *Communications of the ACM, 58*(11), 33–35.

Sengupta, P., Dickes, A., & Farris, A. V. (2018). Toward a Phenomenology of Computational Thinking in STEM. In M. S. Khine (Ed.), *Computational Thinking in STEM: Research Highlights* (pp. 49–72). New York: Springer.

Sengupta, P., Dickes, A., & Farris, A. V. (In Press). *Voicing Code in STEM: A Dialogical Imagination*. Cambridge, MA: MIT Press.

Severance, S., Penuel, W. R., Sumner, T., & Leary, H. (2016). Organizing for teacher agency in curricular co-design. *The Journal of the Learning Sciences*.

Sherin, B., diSessa, A. A., & Hammer, D. (1993). Dynaturtle revisited: Learning physics through collaborative design of a computer model. *Interactive Learning Environments, 3*(2), 91–118.

Spillane, J. P. (1998). State policy and the non-monolithic nature of the local school district: Organizational and professional considerations. *American Educational Research Journal, 35*(1), 33–63.

StarLogo TNG website; http://education.mit.edu/projects/starlogo-tng. Accessed 30 July 2014.

Todorov, T. (1984). *Mikhail Bakhtin: the dialogical principle* (Vol. 13). Manchester University Press.

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. (1999). {NetLogo}.

Wilkerson-Jerde, M., Wagh, A., & Wilensky, U. (2015). Balancing curricular and pedagogical needs in computational construction kits: Lessons from the deltatick project. *Science Education, 99*(3), 465–499.

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

Yackel, E., & Cobb, P. (1996). Sociomathematical norms, argumentation, and autonomy in mathematics. *Journal for Research in Mathematics Education, 22*(4), 390–408.

Yackel, E., Cobb, P., & Wood, T. (1991). Small-group interactions as a source of learning opportunities in second-grade mathematics. *Journal for research in mathematics education, 22*(5), 390–408.

Yin, R. K. (1994). Discovering the future of the case study. Method in evaluation research. *Evaluation practice, 15*(3), 283–290.