



Using Programming to Strengthen Mathematics Learning in 9th Grade Algebra Classes

Dr. Eric A Freudenthal, University of Texas, El Paso

Eric Freudenthal is an Associate Professor of Computer Science.

Dr. Kien H. Lim, University of Texas at El Paso

Dr. Lim's research interests are on students' problem-solving disposition and instructional strategies to advance their ways of thinking. Dr. Lim is particularly interested in impulsive disposition, students' propensity to act out the first thing that comes to mind. Dr. Lim's research goal centers on helping students advance from impulsive disposition to analytic disposition. Dr. Lim and colleagues are currently developing, testing, and refining a survey instrument to assess students' impulsive-analytic disposition. They have been investigating instructional strategies, such as use of prediction items and classroom voting with clicker technology, to help students become aware of their impulsivity and to elicit and address mathematical misconceptions. Dr. Lim is also exploring the use of mathematical tasks to provoke students' intellectual need for the concepts they are expected to learn. Lately, Dr. Lim has been involved in the iMPaCT-Math project to investigate the use of programming activities to foster student learning of foundational algebraic concepts.

Mrs. Sharie Kranz, Coronado HS-EPISD

Catherine Tabor, El Paso ISD and University of Texas at El Paso

Catherine Tabor is a high school math and science teacher in El Paso, Texas. She is also a Ph.D. candidate in Computer Science at the University of Texas at El Paso. Her main focus is student achievement in STEM courses.

Mr. Jeremy L Ramirez, University of Texas at El Paso

An Early Report on Challenges Related to Dissemination of Programming-Centric Mathematics Lessons into 9th Grade Algebra Classes

*Eric A. Freudenthal
Kien H. Lim, Jeremy Ramirez
University of Texas at El Paso*

*Sharie Kranz
Catherine Tabor
El Paso Independent School District*

Algebra I is a gateway course for high school and college STEM. Ninth graders who develop mastery of Algebra I learning outcomes are well prepared to comprehend and succeed in the quantitative STEM courses such as physics. Those who do not are likely to encounter difficulty in subsequent STEM coursework.

Our “iMPaCT-Math” activities can potentially transform high school algebra classrooms by incorporating programming, enhancing students’ conceptual knowledge of core algebraic ideas, and reducing the chronic underrepresentation of Hispanics and women who study computation. The iMPaCT-Math project has developed a sequence of programming based learning modules (LMs) intended to complement traditional instruction within traditional 9th grade Algebra I classrooms.

During the summer of 2012, a dozen ninth grade mathematics teachers from urban high schools attended a three day professional development program (PDP) that prepared them to integrate the iMPaCT-Math LMs within their classes. The PDP was designed to help teacher participants to develop stronger pedagogical content knowledge. They experienced “teacher versions” of IM activities that facilitate their identification and correction of misconceptions and exploration of basic concepts. In the following fall semester, we observed teacher implementations of the lessons to determine how teachers implement the IM activities in their classroom and how students respond to those activities.

We report on initial findings from classroom observations. Our findings indicate challenges in preparing teachers to effectively implement aspects of the activities intended to elicit analytical engagement. Teachers who are primarily focused on procedural fluency tend to provide hints that circumvent analysis. These results have led the design team to begin consideration of alternate ways for implementing the activities including the development of fully automated teaching environments.

This paper is organized as follows. The first section provide a brief historical account of the emergence of IM and our initial efforts. The second section provides an overview of IM. In the third section, we describe the professional development program and early results from teacher observations. We conclude with a summary of planned extension activities.

1. From Media Computation to iMPaCT-Math

iMPaCT-Math (IM) is an approximate acronym for Media-Propelled Computational Thinking for Mathematics Classrooms, which fairly reflects our ambition and our stance – engagement

with graphical programming challenges that focus student attention towards exploring mathematics principles will propel students towards exploration of science, computational thinking and engineered design.

IM consists of threaded sequences of games and project-based-learning activities that can be infused within conventional high school and college mathematics courses. Our studies have motivated us to rethink **how** computation can engage students as active learners of mathematics, and enhance their appreciation of what they can accomplish with their own growing skills. In IM exercises, students explore, modify and extend tiny programs that render graphics using algorithms that expose foundational mathematical concepts in an intuitive manner. The enrichment exercises are designed to provide visceral, real-world intuition for the textbook abstractions of math concepts such as slope, intercepts, and acceleration.

IM was developed at the University of Texas at El Paso, a primarily Hispanic-Serving Institution serving the bi-national El Paso – Ciudad Juarez metropolitan area. IM's first substantial dissemination is to Algebra-1 classrooms in three El Paso high schools during the 2011-2012 academic year involving approximately seven hundred students.

IM was developed as an unexpected synergy of two NSF-funded research efforts conducted by the first author at The University of Texas at El Paso (UTEP). IM's pedagogy evolved from an ultra-elementary programming thread called "Media Exposed" incorporated into UTEP's Entering Students' Program. Media Exposed targeted students taking (and often stuck retaking) algebra and pre-calculus courses. The course initially used a Python-based programming environment and libraries that enable students to focus on achieving dramatic media artifacts rather than the underlying mathematics. As observed at other institutions, many students without prior interest or exposure to programming who took this course reported increased interest in programming.^{17, 12}

Focus groups with our target group (intending engineers and CS majors) indicated dissatisfaction. These students were frustrated by the graphically focused activities. They indicated that, as intending engineering, they wanted opportunities to learn how the graphical library was implemented, not just how to use it. This observation triggered a sequence of refinements that eventually resulted in a new course that uses the programming of simple mathematical algorithms that render graphics and simulate kinematics. These tiny programs focus student attention on exploring principles underlying (and building "gut level" intuitions related to) the content of high-school level algebra and pre-calculus courses that three quarters of pre-engineering students attend, and half generally fail.

Results were dramatic:^{9,10,13,16}

- The failure rate among engineering students concurrently enrolled in the remedial algebra and pre-calculus course was halved. *Their failure rate in a subsequently calculus course was also halved.*
- At course completion, three quarters of attendees from engineering, science, liberal arts, and nursing reported positive attitudes towards programming and problem-solving, including quantitative reasoning, *independent of ethnicity, gender, and intended major.*

- Enrollment in the post-remedial CS-1 more than doubled.

Reflecting its evolving focus, Media Exposed was renamed twice: “Computational CS-Zero” became “Media Propelled Computational Thinking” (iMPaCT). Initial pilots of iMPaCT were so successful at improving the pass rates and enhancing interest in UTEP engineering programs that there are now four different versions offered for students in a variety of majors. We currently have variants of iMPaCT activities for mechanical and electrical engineers, and variants are offered by other universities both to their students and as outreach to high schoolers.

1.1. iMPaCT-Math for High Schools

Most mathematics examined in iMPaCT exercises are taught in high school. A modest pilot program to infuse iMPaCT lessons into Algebra I classes by two high school teachers was initiated after we discovered at that iMPaCT lessons were engaging and effective with high (and even middle) school students attending outreach programs hosted at UTEP’s campus. Initial results from this pilot program outstripped our expectations. In iMPaCT-enriched high school math classes that were otherwise unmodified (in terms of classroom time, textbook, and curriculum):

- The majority of students who had failed Algebra I multiple times passed district benchmark exams calibrated against state end-of-course exams after only one half year of iMPaCT-enriched instruction.
- More than one quarter of eligible students exposed to programming in math courses that incorporated iMPaCT lessons voluntarily enrolled in AP-CS with representative gender and ethnic distributions.

Our effort targeted Algebra I because its core lessons focus on Algebra I outcomes, and because Algebra I is a gateway course for STEM studies. Unfortunately, algebra is inherently difficult for students because of its abstract, decontextualized symbolism.^{2,5,23,29,35} Much of high school and college science and math depends upon a sturdy scaffolding of core procedural and conceptual Algebra I outcomes such as variables, graphing, slope, and linear functions; thus it is a *Gateway to a Technological Future*.²³ Making the abstraction of the fundamental ideas in this course more comprehensible to students has the potential for greatly increasing the quantity and quality of students in STEM fields.

A curriculum writing team (CWT) of three Algebra I teachers was recruited to develop a set of Learning Modules (LMs) during the Summer of 2011. These LMs include activities that scaffold core concepts and provide opportunities to practice skills common to Algebra I curricula. Each LM contains a sequence of learning and assessment activities, and some of the assessment components are embedded into learning activities. Activities are in the form of games or woven into stories and artistic projects in a manner intended to motivate student engagement and expose connections between mathematical concepts and procedures. Sample PowerPoint presentations and worksheets are also included. Each module also contains a teacher-support document that provides guidance on its intended implementation.

1.2. Initial Dissemination

We initially disseminated to two schools that employ members of our CWT. Algebra I teams from both schools had encouraged their administration to collaborate with us after attending hands-on hour-long workshops on iMPaCT's approach. However, despite the LM's availability, most of these math teachers had little prior experience with programming and felt underprepared to implement the lessons. Fortunately, the math coach at one of the schools is a member of the CWT. With her one-on-one coaching, most members of her team began to implement the lessons. Only one teacher at the other school (also a CWT member whose schedule did not permit similar one-on-one support) implemented the lessons.

We learned much from this experiment. The teaching coach who trained an entire teaching team provided numerous observations and suggestions regarding teacher preparation. In addition, we collected written feedback and conducted a focus group meeting with her team of teachers. They expressed high enthusiasm towards iMPaCT as well as challenges they encountered in the implementation of those modules. They specifically indicated that a professional development program (PDP) would be needed for the project to succeed. In addition, they requested the development of additional support materials to help students practice skills and concepts.

1.3. PDP and Second Dissemination (AY 2012-2013)

In August 2012, we hosted a three-day professional development program for Algebra I teachers from three regional schools. The design of the workshop was intended to reinforce teachers' comprehension of mathematics concepts related to the lessons. Pre and post workshop survey instruments were used to determine participant enthusiasm for the program and assess teacher mastery of underlying mathematics concepts. We tracked teachers at one of the schools and learned that, despite the PDP, implementation fidelity varied dramatically.

Approximately half of the teachers circumvented students' exploration by prompting them to recall and apply a previously memorized procedure that computes slope to determine coefficients. Our observers noted that these teachers employed a procedural (rather than analytical) teacher-centered teaching style for both their explanations of iMPaCT and traditional Algebra-I topics. Both teachers and observers in these classes reported indicators of low student engagement. In these classes, we expect that few, if any, learning gains will be advanced by the activities. Our future plans include the examination of strategies for reducing the need for extensive instructor preparation through the development of media and instructional software that support effective implementations of our activities.

2. A Taste of iMPaCT-Math

A main strength of iMPaCT is that it provides an experiential-visual context for students to make connections across multiple representations: (a) statements in a program, (b) computational process; (c) graphical output, and (d) underlying mathematical concepts.

For example, to interpret plots of functions (a major learning outcome of Algebra I), students must be proficient at reading graphs including accurately identifying and differentiating the x and y axes and understanding that a point's position is defined by a unique (x, y) coordinate. In

an early exercise titled “Missing Piece,” students are provided the program appearing in the leftmost pane of Figure 1 that draws dots at specified coordinates and, when run, draws the image in the middle pane. The assignment: fix the “broken heart.” Teachers report that students’ attention is captivated by this activity and its successor where students are challenged to create a drawing or a piece of art such as the apple in the rightmost frame. More importantly, teachers report that most students quickly and painlessly develop the lesson’s intended outcomes regarding Cartesian coordinates. The teachers also indicated that this activity more effectively and painlessly introduced students to the calculator’s keyboard and graphical display than exercises they had previously used.

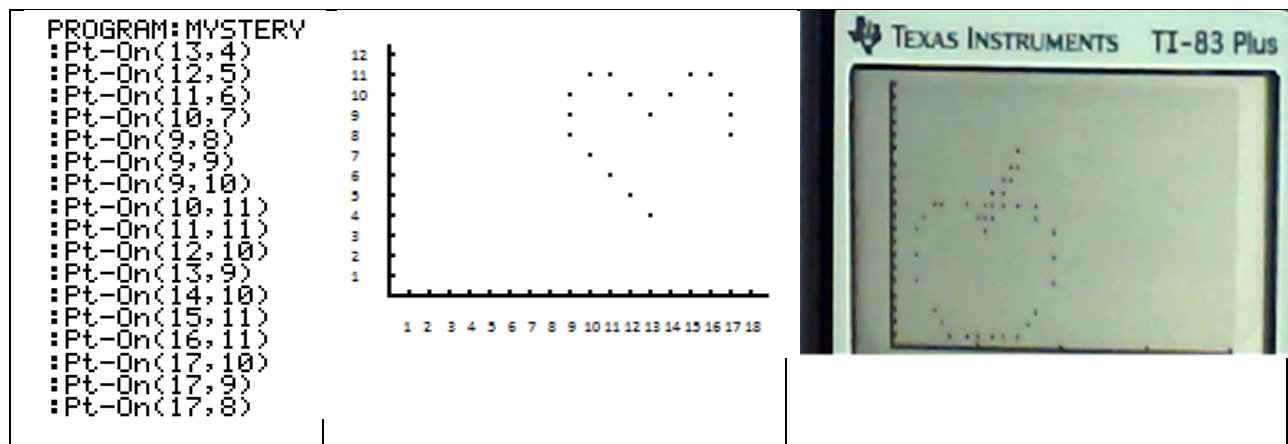


Figure 1. Components from activities related to Cartesian coordinates

Figure 2 contains several frames from an IM LM related to lines.

Prior to this lesson, students have reviewed Cartesian coordinates and have examined arithmetic expressions and variables in a program that tracks the amount of money in a story-book character’s wallet during a “trip to the mall.”

This introduction to lines begins by challenging students to predict-then-verify program LINE0’s output. LINE1 introduces iteration in the form of “while loops” as a convenience for drawing long lines. After exercises that challenge the students to draw various graphical output (and force them to examine the function of each statement in LINE1), student teams are then challenged to predict-then-verify LINE2’s output. The prediction portion generally elicits much discussion, disagreement, and confusion, followed by many “aha’s” as students discover, realize, and explain to each other why the line tilts. When challenged to draw a “less steep line,” most will decrease the y -increment to one and beam at their easy success.

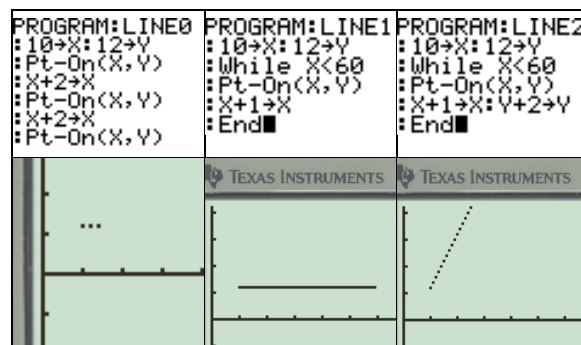


Figure 2. Programs introducing linear relationships

Their “concrete” actions of making dots appear steeper or less steep allows students to viscerally experience essential properties of linearity such as constant rate-of-change, which underlies the

slope formula's concepts of "rise" and "run." Some students inevitably increase the x-increment and achieve similar results. Even this minor variation in student implementation provides a teachable moment regarding the effects of changing the relative rate of x- and y-change that are relevant to Algebra I learning outcomes regarding slope.

Games, creative, and story-based activities motivate and challenge students to figure out how to (1) draw parallel lines by changing the initial coordinate, but not x- and y-increments, (2) draw segments that avoid obstacles in a maze or connecting specified points, (3) draw vertical lines by incrementing y but not x, and (4) quickly compute the location of the n^{th} dot using multiplication. The diversity of activities poses repeated reasoning tasks exposing nuance such as why slope is an inappropriate metric for vertical lines.

Each of these discoveries and connections corresponds to an important piece of conceptual scaffolding needed to render standard equation forms such as $y = mx + b$ comprehensible.

Sometimes the best way to learn that a concept is critical is to examine what happens if it is violated. The program in Figure 3 doesn't draw a line because the y-increment is explicitly changed. Later the same program can expose why a constant second difference makes a function quadratic, and why the graph of a quadratic function has a vertex.

We initially developed these programming-based lessons for college students who had memorized equations but not understood why they worked. College students literally squealed with delight as they realized that they had personally re-invented $y = mx + b$. IM is attempting to reverse this process to scaffold these equations using programming in the class where they are first taught.

Below are the design objectives and design principles our team of four faculty from two departments, three high school teachers, and three college students have adopted in the development of IM LM.

2.1. Design Objectives of IMPaCT-Math Learning Modules

Easy adoption: By design, IM is a very low-overhead enrichment program that is easy to integrate with, as opposed to replace, the Algebra I curricula that are already in use in nearly every high school. To keep the adoption costs down to zero (apart from teacher training), the graphics programming, which originally required computers, was also ported to graphing calculators. To ensure that the programming is accessible to students in pre-existing math classes, our lessons use a tiny subset of the programming language. While their understanding may initially be superficial, students are generally having fun learning to program graphics that authentically contextualize and reinforce math concepts.

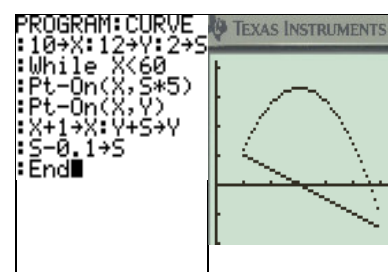


Figure 3. Non-constant rates of change result in curves!

Engagement and Exploration: Rather than relying on a graphing calculator (or other plotting system) to plot a function, students are challenged to experiment with mathematical instructions that tell the machine what to do, and explore and explain mathematical ideas that can connect graphical attributes to the computational procedures. Students learn as much from their errors as their successes. For example, when the picture comes out all wrong, the students review their programs (which are just sequences of mathematical instructions) to fix the errors themselves.

2.2. Design Principles for IMPaCT-Math Learning Modules

IM LMs are designed around three informally tested principles.

- **Graphics are used to focus students' attention on the intended mathematical challenge.** At each stage, the graphical challenges require students to either reinforce or extend their prior knowledge. The graphical context makes the nature of student mistakes become more apparent. Due to the direct coupling of the students' program and the graphical output, a correctly drawn image serves both as certification of and reward for a problem well solved.
- **Programming** (but not the problem-solving) **is taught via scaffoldings** that the students can adapt as they advance through a series of challenging exercises (see Figure 1). This keeps the programming overhead very low and allows the students to focus on the underlying math.
- **Intuition complements formalisms.** The program-computation-graphical output connections expose ground-level understandings that scaffold the powerful mathematical abstractions taught in Algebra I.

Mathematics as a creative outlet. Based on the feedback from teachers who piloted the iMPaCT activities, we learned that students enjoy being creative and like to have a sense of ownership. Whenever appropriate, our activity will offer students choices. For example, in an LM related to Cartesian coordinates, the draw-a-picture activity forces students to focus on the primary learning outcome while permitting them to create any design they like.

Additional Design Principles. Based on Carlson's principles for designing a perfect Science Short⁴ (i.e., staged demonstration), we will use the following design principles to guide the development of iMPaCT activities: (a) create cognitive conflict to engender learning; (b) stimulate curiosity; (c) provide epiphany that builds intuition and positive emotion; (d) keep it simple, brief, and clear (e.g., not too many things at a time); and (e) empower students to investigate independently or collaboratively. We will adhere to additional characteristics of good tasks, such as those identified by Clements:⁷ (a) are meaningful to students; (b) build on students' prior knowledge; (c) encourage students to devise solutions; (d) lead to important mathematical ideas; and (e) instigate further investigation and generation of new problems.

2.3. Differences between IMPaCT-Math and other software-based tools to enhance math learning

Several software programs can graph a line, or generate the coordinates of points on a line. Their strengths may be broadly categorized into algebra, geometry, tables, and applications. Here is how IM complements these approaches:

Algebra. Graphing calculators (GC) and programs such as Mathematica, Matlab, and Maple will graph any function that can be defined by formulas, including (non-vertical) lines. Texas high schools and math courses at UTEP use GCs in this manner. As faculty who teach upper division STEM courses, we observe that many of our students have memorized the effects of changing m and b without understanding *why* the plot is changed. IM explicitly forces students to come to grips with what makes a line straight, and what slope really means, as the ratio of change in x to change in y between "consecutive" points.

IM's implicit use of parametric forms (x and y depend upon a step parameter) exposes the idea that there are many ways to express the same slope (e.g. proportionally changing x 's and y 's rates of change). Graphing y as a function of x doesn't allow this because the y coefficient is always 1. Other high-level programming languages (including graphing calculators) also allow parametric equations, but the syntax is much more involved. Finally, when pedagogically appropriate, an IM LM can use the same programming syntax to plot formulae, for example, directly computing y from a slope-intercept equation whose input is variable x .

Geometry. Programs such as Geometer's Sketchpad and Geogebra can construct lines geometrically, for instance drawing the unique line that passes between two specified points. But the line-drawing command is built into the program; students do not write this command or examine how it works.

Tables and Functional Programming. Like many IM lessons, spreadsheets (and spreadsheet-like programs such as Fathom) and functional programming languages have been used to enable students to examine mathematical processes through summation. Like other computer representations of algebraic concepts, the abstractions for variables and expressions in spreadsheet cells and various programming languages all differ significantly from algebra and may teach different lessons to different learners. Distinct features of the programming model used by iMPaCT for this context include (a) its direct and compact representation of the sequence in which computations are performed, and (b) its enabling of a productive gateway towards the study of the most popular programming languages, and potentially the study of computer science.

Applications. Dynamic virtual manipulatives can illustrate simple simulations of linear relationships, and even plot graphs. These can make the abstraction of lines and linear relationships more real. In contrast, IM forces students to explicitly specify the process of changing both x and y , thus exposing why and how these abstractions are related. Furthermore, IM uses the same programming syntax in multiple lessons to teach (and relate) different concepts. This avoids the need to spend class time teaching students how to use multiple tools and may convince students that programming is relevant.

Finally, IM runs on (but is not limited to) handheld calculators, which are much more common than general purpose computers in high school classrooms.

IM's approach is informed by current research in mathematics learning and professional development.

2.4. Five Strands of Mathematical Proficiency

While integration of procedural fluency and conceptual depth is critical for success in quantitative STEM studies it is notoriously difficult to achieve because memorizing procedures provides students the short-term benefit of passing tests.

Attempts to address this problematic procedural-conceptual relationship are reflected in the evolution of learning objectives defined by the major efforts to reform math education. Connally, Gleason and Hughes-Hallett¹¹ identified three procedural fluencies related to manipulating and interpreting illustrated in the upper left box of . While the designers of this *rule of three* explicitly described relationships among these various representations, The current “five intertwined strands” model of mathematical proficiency”²⁸ aggregates skills from the rule-of-three as *procedural fluency* and enumerates three new attributes— *conceptual understanding*, *strategic competence*, and *adaptive reasoning*—that constitute conceptual depth in our math-computation proficiency framework (see **Error! Reference source not found.**). Together, they can enable students to develop *productive disposition*, an affective trait related to self-efficacy resulting from a combination of fluency at applying standard computational techniques with a deep understanding of how and why they can be applied.

Stigler and Hiebert³² characterized current U.S. teaching as “learning terms and practicing procedures”, as opposed to Japanese teaching’s “structured problem solving.” Ma’s study of U.S. and Chinese elementary math teaching²⁶ observed related phenomena. More than 60% of U.S. teachers described *skills* as the *main thing* that they wanted their students to learn. Co-PI Dr. Lim works with teachers in El Paso and has observed many teaching procedures without a conceptual focus. Aharoni¹ observes that it is difficult to build advanced conceptual understandings upon a foundation of rote procedure and advocated that students must examine and internalize each operation’s multiple meanings We hypothesize that IM’s programming will engage learners in exploring, explaining, and reflecting and enhance their procedural-conceptual connections.

2.5. Harel’s DNR-based Instruction

The iMPaCT LMs and PDPs are in accordance with Harel’s Duality, Necessity, and Repeated-Reasoning (DNR) conceptual framework.¹⁹ “DNR is a conceptual framework extending Piaget’s and Vygotsky’s notions of cognitive development to stipulate conditions for achieving critical goals such as provoking students’ intellectual need to learn mathematics, helping them to construct mathematical ways of understanding and ways of thinking, and assuring that they internalize and retain the mathematics they learn.”²¹

- Duality: Content knowledge and thinking processes must be developed together. *IM LMs enhance students’ conceptual depth and procedural fluency by engaging them in mathematical thinking (e.g., conjecturing, investigating causality). Students’ capacity to think mathematically is enhanced through the conjectures they formulate, explanations they provide, procedures they develop, and solutions they produce.*

- **Necessity:** To learn a mathematical concept, students must have an intellectual need to understand it. *Changing the slant of a line to draw a roof may necessitate examination of principles related to slope.*
- **Repeated Reasoning:** Reasoning (not just tasks) must be practiced. *Graphical programming provides an engaging domain for superficially different but conceptually equivalent problems.*

In summary, *learning* is defined as “a continuum of disequilibrium–equilibrium phases” that (a) are instigated by intellectual and psychological needs, (b) require application of prior knowledge and mathematical thinking, and (c) result in enhancement of knowledge and mathematical thinking.

Mathematical Habits of Mind. When ways of thinking like mathematicians are internalized and become spontaneous, they can be regarded as mathematical *habits of mind*. The emphasis on mathematical habits of mind is reflected in the *Process Standards* in NCTM’s Principles and Standards for School Mathematics²⁷ as well as in the more recent Standards for Mathematical Practice in the Common Core State Standards in Mathematics⁸ document. Examples of Mathematical Practice standards include “make sense of problems and persevere in solving them”, “look for and make use of structure” and “look for and express regularity in repeated reasoning” (p. 8). IM activities pose authentic challenges amenable to reflective problem solving *at the students’ level*,

Tarr et al. found that a learning environment in which students make conjectures, explain their strategies, and work towards shared understandings have a positive iMPaCT on students’ achievement on assessments that measure mathematical reasoning and problem solving.³³ Their results suggest that teaching such curricula is more challenging and teachers would require more professional development.

3. Research-based Professional Development

iMPaCT activities are designed for students to reason about procedures. They are considered high *cognitive demand* tasks. Effective implementation of high cognitive demand tasks are inherently challenging : Stein, Grover, and Henningsen reported that 63% of math lessons intended to require high levels of cognitive engagement are actually implemented in a manner that replaced cognitive demand with unsystematic exploration, procedures without connections, and the absence of mathematics.³¹ To ensure teachers implemented the iMPaCT materials as intended, we developed the PDP and conducted regular support meetings. Discussions highlighted factors that were found to be associated with maintenance of high-level cognitive demands,²² including building on students’ prior knowledge, providing sufficient time for students to explore, providing means for students to monitor their own progress, scaffolding of student thinking and reasoning, pressing for justifications and explanations, and drawing conceptual connections.

In professional development workshops, the context of programming offers teachers a novel learning experience where they can be challenged to make explicit the mathematical concept(s) that underlie(s) the program, the embedded computation, and the corresponding graphical output.

Our intention was that the teacher version of IM activities would enhance teachers' conceptual understandings.

3.1. Structure of the PDP

A dozen teachers from four public high schools in El Paso attended the PDP during August 2012. Each of the three IM units (coordinates, equations, and linear functions) was examined in a three-phase process: In the first phase, the teachers were attended slightly modified teacher-versions of the activities. The second phase focused on connections to Algebra I learning outcomes and included exercises designed to deepen participants' pedagogical content knowledge. The third phase was intended to support teachers' development of relevant content knowledge for teaching and included exercises that enabled participants to explore how they would present both the activities and their mathematical connections to students.

3.2. Initial Observations of Teacher Implementations

We observed the implementation of IM activities by seven Algebra 1 teachers from one high-school, four teachers attended the summer PDP workshop and three did not. These Algebra 1 teachers work as a collaborative team and meet each week for an hour to plan and decide what to cover each week. The team leader, who attended the summer PDP workshop, would generally go over each IM activities selected for implementation, highlighting the important aspects of the activity. Some teachers seemed to treat these as enrichment activities that are peripheral to the core curriculum. Since an IM activity might be implemented in different weeks by different teachers, the observer has an impression that these activities were implemented by some teachers when they can spare a period. Nevertheless some teachers, especially those who attended the PDP workshop, seemed to appreciate the power the IM activities have in engaging students to think and to reason.

Initial observations of teacher implementations indicate a range of emphases. Approximately half of the teachers followed our intended model of exploiting the graphical programming lessons to contextualize discussion of algebraic concepts such as rise and run. Others seemed more concerned that their students program on the graphing calculator correctly, obtain the expected graphical output, and complete the worksheet. Consequently students did not explore on their own and make the own mathematical connections; in some classrooms students did not know how to because they are not used to inquiry-based learning..

3.3. Dissemination

With the ultimate goal of widespread adoption of the iMPaCT activities, we are proceeding cautiously in order to ensure the best quality products and processes are available. Before widespread dissemination, we must ensure that an strategy for achieving effective implementations has been refined and evaluated. Thus, we only implemented a small-scale dissemination focusing on the dozen teachers who attended the PDP. Challenges observed from that process lead us to now consider the incorporation of iMPaCT activities within an integrated teaching system that can incorporate elements from successful presentations by instructors whose lessons have been successful.

We recognize the challenges in shifting teachers' paradigms about teaching and learning, especially when their primary concern is to ensure students "know" the procedures for solving problems in the high-stake examinations. We are aware that although a 3-day workshop can help teachers understand the components and the objectives of IM activities, it is unlikely to change the way teachers teach, from a teacher-oriented approach to an inquiry-based approach. To achieve the latter, require long-term professional development of a different nature which is beyond our scope.

4. Synopsis

We seize the opportunity to use introductory programming within a math course as a vehicle to present mathematical concepts from an intuitive and engaging perspective that appears to strengthen interest in, and understanding of, technical content.

Pilot studies support our hypothesis that that the embedding of programmed computation into math courses can provide a viable path towards increased understanding of foundational math concepts and near universal understanding of the nature of programmed computation. Through this marriage, computation will expose mathematics' best side (as a way to understand, predict, and manipulate phenomena), as mathematics exposes computation's best side (as a way to simplify through explicit specification of automatable processes).

These pilot studies also indicate challenges in the preparation of math teachers to effectively implement them. The effort's next steps will include examination of the construction of an instructional software system that incorporates iMPaCT's most successful activities in a manner that reduces the depth of content knowledge for teaching that educators must develop prior to effectively utilizing these activities in their classes.

The learning modules are available to educators who would like to collaborate in their testing and refinement.

More information on IM is available online at <http://iMPaCT-Math.org>.

Acknowledgement

This report is based on work supported by the National Science Foundation through grants IIS-0829683 and DUE-0717877 and gifts from Microsoft Inc, Texas Instruments Inc. and Calculex, Inc. Assistance in project planning and evaluation was provided by the NSF-BPC Computing Alliance of Hispanic Serving Institutions (CNS-0837556). Any opinions, findings, and conclusions or recommendations expressed in the paper are those of the authors and do not necessarily reflect the views of the NSF or other sponsors.

References

1. Aharoni, R. (2007). Arithmetic for Parents: A Book for Grownups about Children's Mathematics. El Cerrito: Sumizdat.

2. Booth, L. R. (1988). Children's difficulties in beginning algebra. In A.F. Coxford (Ed.), *The ideas of algebra, K-12: 1988 Yearbook* (pp. 20-32). Reston, VA: National Council of Teachers of Mathematics.
3. Campe, K. D. (2011). Do it right: Strategies for implementing technology. *Mathematics Teacher*, 104 (8), 620-625.
4. Carlson, S. (2011), *Secrets of Stellar Science Demos*. Unpublished manuscript.
5. Chazan, D. & Yerushalmy, M. (2003). On appreciating the cognitive complexity of school algebra: Research on algebra learning and directions of curricular change. In Kilpatrick, J., Schifter, D. & G. Martin (Eds.), *A Research Companion to the Principles and Standards for School Mathematics* (pp. 123-135). Reston: National Council of Teachers of Mathematics.
6. Clarke, D. (1994). Ten key principles from research on the professional development of mathematics teachers. In D. B. Aichele & A. F. Coxford (Eds.), *Professional development for teachers of mathematics* (pp. 37-48). Reston: National Council of Teachers of Mathematics.
7. Clements, D. H. (2000). From exercises and tasks to problems and projects: Unique contributions of computers to innovative mathematics education. *Journal of Mathematical Behavior*, 19(1), 9-47. doi:10.1016/S0732-3123(00)00036-5
8. Common Core State Standards Initiative (2010). Retrieved from http://www.isbe.state.il.us/common_core/pdf/Math_common_core_standards.pdf
9. Computing Alliance of Hispanic Serving Institutions. (2008). NSF 2008 Annual Report. (NSF BPC-A Award # 0540592). El Paso, Texas: University of Texas at El Paso. Retrieved from <http://cahsi.cs.utep.edu/Portals/0/AnnualReport2008.pdf>
10. Computing Alliance of Hispanic Serving Institutions. (2009). NSF 2009 Annual Report. (NSF BPC-AE Award # 0837556). El Paso, Texas: University of Texas at El Paso. Retrieved from <http://cahsi.cs.utep.edu/LinkClick.aspx?fileticket=ITsF-nmnN8k%3D&tabid=102>
11. Connally, E., Hughes-Hallett, D., Gleason, A. M., Cheifetz, P., Davidian, A., Flath, D. E., & et al. (2006). *Functions modeling change - A preparation for Calculus* (3rd ed.). Wiley.
12. Freudenthal, E., Roy, M. K., Ogrey, A. N., and Gates, A. Q. (2009). A creatively engaging introductory course in Computer Science that gently motivates exploration of mathematical concepts. (AC 2009-2188). *Proceedings of ASEE Annual Conference*.
13. Freudenthal, E., Roy, M. K., Ogrey, A., Magoc, T., & Siegel, A. (2010). Media Propelled Computational Thinking. *Proceedings of the 41st ACM technical symposium on Computer Science* (pp. 37-42). New York: ACM.
14. Freudenthal, E., Ogrey, A., & Gonzalez, R. (2010). Work in progress – Eliciting integrated understandings of high school STEM curricula through programming. *Frontiers in Education Conference (FIE), 2010 IEEE* (pp. F2H-1-F2H-2). Washington, D.C.: FIE. doi: 10.1109/FIE.2010.5673203
15. Freudenthal, E., Ogrey, A. N., Roy, M. K., & Siegel, A. (2010). A computational introduction to STEM studies. *Education Engineering (EDUCON), 2010 IEEE* (pp. 663-672). Madrid, Spain: EDUCON. doi: 10.1109/EDUCON.2010.5492514
16. Freudenthal, E., Duval, A., Ogrey, A., Lim, K., Hug, S., Tabor, C., Gonzalez, R. Q., & Siegel, A. (2011). Planting the seeds of computational thinking: An introduction to programming suitable for inclusion in STEM curricula. (AC 2011-2702). *Proceedings of ASEE Annual Conference*.
17. Guzdial, M. (2005). *Introduction to computing and programming in Python: A multimedia approach*. New Jersey: Pearson-Prentice Hall.
18. Harel, G. (2008a). What is mathematics? A pedagogical answer to a philosophical question. In R. B. Gold, & R. Simons (Eds.), *Current issues in the philosophy of mathematics from the perspective of mathematicians* (pp. 265-290). Mathematics Association of America.
19. Harel, G. (2008b). DNR perspective on mathematics curriculum and instruction: Focus on proving, Part I. *Zentralblatt fuer Didaktik der Mathematik*, 40, 487-500. doi:10.1007/s11858-008-0104-1

20. Harel, G. (2008c). DNR perspective on mathematics curriculum and instruction: With reference to teacher's knowledge base, Part II. *Zentralblatt fuer Didaktik der Mathematik*, 40, 893–907. doi: 10.1007/s11858-008-0146-4
21. Harel, G., & Koichu, B. (2010). An operational definition of learning. *Journal of Mathematical Behavior*, 29, 115-124. doi:10.1016/j.jmathb.2010.06.002
22. Henningsen, M., & Stein, M. K. (1997). Mathematical tasks and student cognition: Classroom-based factors that support and inhibit high-level mathematical thinking and reasoning. *Journal for Research in Mathematics Education*, 28(5), 524-549.
23. Katz, V. (Ed.) (2007). *Algebra: Gateway to a technological future*. Washington, D.C.: The Mathematical Association of America.
24. Lego Education North America. (2011). Lego education. Retrieved from <http://legoeducation.us>
25. Lim, K. H., Buendía, G., Kim, O.K., Cordero, F., & Kasmer, L. (2010). The role of prediction in the teaching and learning of mathematics. *International Journal of Mathematical Education in Science and Technology*, 41(5), 595-608.
26. Ma, L. (1999). *Knowing and Teaching Elementary Mathematics*. Mahwah: Laurence Erlbaum and Associates.
27. National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics*. Reston, VA: National Council of Teachers of Mathematics.
28. National Research Council (2001). *Adding it up: Helping children learn mathematics*. Kilpatrick, J., Swafford, J., & Findell, B. (Eds.). Washington, D.C.: National Academy Press.
29. Sfard, A., & Linchevski, L. (1994). The gains and the pitfalls of reification—The case of algebra. *Educational Studies in Mathematics*, 26, 191-228.
30. Smith, M.S., & Stein, M. K. (2011). *5 Practices for orchestrating productive mathematics discussions*. Reston, VA: National Council for Teachers of Mathematics.
31. Stein, M.K., Grover, B.W., & Henningsen, M. (1996). Building student capacity for mathematical thinking and reasoning: An analysis of mathematical tasks used in reform classrooms. *American Educational Research Journal*, 33(2), 455-488.
32. Stigler and Hiebert (1999), *The Teaching Gap: Best Ideas from the World's Teachers for Improving Education in the Classroom*, Free Press.
33. Tarr, J. E., Reys, R. E., Reys, B. J., Chavez, O., Shih, J., & Osterlind, S. (2008). The iMPaCT of middles grades mathematics curricula on student achievement and the classroom learning environment. *Journal for Research in Mathematics Education*, 39, 247-280.
34. Texas Education Agency. (2010). Student Assessment Division, STAAR Algebra I Blueprint. Retrieved from <http://www.tea.state.tx.us>
35. Vinner, S. (1997). The pseudo-conceptual and the pseudo-analytical thought processes in mathematics learning. *Educational Studies in Mathematics*, 34, 97-129.
36. Carlo Amato, Sharie Kranz, Catherine Tabor, Juan Castillo, Alexandria Ogrey, Adrian Veliz, Art Duval, Kien Lim, Amy Wagler, and Eric Freudenthal, iMPaCT-STEM: games & activities that motivate exploration of foundational algebra concepts—while inadvertently scaffolding computational thinking and engineered design, Proc. ASEE 2012.