AC 2011-2702: PLANTING THE SEEDS OF COMPUTATIONAL THINKING: AN INTRODUCTION TO PROGRAMMING SUITABLE FOR INCLUSION IN STEM CURRICULA

Eric A Freudenthal, University of Texas, El Paso

Dr. Freudenthal is an Associate Professor of computer science at the University of Texas at El Paso

Dr. Art Duval, University of Texas at El Paso

Art Duval is a Professor in the Department of Mathematical Sciences at the University of Texas at El Paso.

Dr. Sarah Hug, University of Colorado, Boulder

Dr. Sarah Hug is Research Associate at the Alliance for Technology, Learning, and Society (ATLAS) Institute, University of Colorado at Boulder. Dr. Hug earned her PhD in Educational Psychology at the University of Colorado, Boulder. Her research and evaluation efforts focus on learning science, technology, engineering, and mathematics, with a special interest in communities of practice, creativity, and experiences of underrepresented groups in these fields across multiple contexts.

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

Kien H. Lim is an assistant professor at the University of Texas at El Paso. He is interested in students' mathematical thinking and disposition. He enjoys creating problems that can challenge students to think, that deepen their mathematical understanding, and that enhance their problem-solving ability.

Catherine Tabor, El Paso Independent School District

Catherine Tabor is a high school math and computer science teacher in El Paso, Texas. She holds a BA in mathematics, a BS in physics , and a MA in Science Teaching (emphasis physics).

Rebeca Q. Gonzalez, UTEP-Graduate Student and EPISD-Teacher

A former Electrical engineering from ITCJ in Mexico currently teaching 9-12 pre-engineering courses and computer science and a master of arts in teaching science graduate student from University of Texas at El Paso.

Prof. Alan Siegel, New York University

Alan Siegel is a professor in the department of computer science and NYU. His research is in the mathematical foundations of computer algorithms, and in the pedagogical approaches used to teach introductory mathematical topics and algorithms in the US and abroad.

Planting the seeds of computational thinking:

An introduction to programming suitable for inclusion in STEM curricula

Abstract

Inadequate math preparation discourages many capable students – especially those from traditionally underrepresented groups – from pursuing or succeeding in STEM academic programs. iMPaCT is a family of "Media Propelled" courses and course enrichment activities that introduce students to "Computational Thinking." iMPaCT integrates exploration of math and programmed computation by engaging students in the design and modification of tiny programs that render raster graphics and simulate familiar kinematics. Through these exercises, students gain experience and confidence with foundational math concepts necessary for success in STEM studies, and an understanding of programmed computation.

This paper presents early results from our formal evaluation of semester-length iMPaCT courses indicating improved academic success in concurrently and subsequently attended math courses. They also indicate changes to the nature of student engagement with problem solving using mathematics. This paper also describes iMPaCT-STEM, a nascent effort of computer science and mathematics faculty to distill iMPaCT's pedagogy into sequences of short learning activities designed to teach and reinforce a variety of mathematical and kinematic concepts that can be directly integrated into math and science courses.

1. Introduction

The iMPaCT (Media Propelled Computational Thinking) project[4,5,6,7,8,12] is an increasingly broad effort to enhance engagement with and success in STEM disciplines for a wide demographic of students -- including those with inadequate mathematics backgrounds. The difficulties in such an endeavor are well known, and the societal benefits well understood.

As a consequence, many projects have been designed with the intention of enhancing interest in STEM studies. iMPaCT addresses the complementary challenge of providing under-prepared students with the reflective understandings of mathematical relationships needed to succeed in engineering programs and other core STEM subjects. In the process, iMPaCT introduces these students, who often have no prior exposure to imperative programming, to the basics of computational thinking motivated by problems they understand and care about.

The original semester-length Jython-based iMPaCT course[5] has been decomposed into a network of threaded sequences of educational modules suitable for inclusion within conventional mathematics and science courses. The overarching idea is to teach very lightweight computing that begins with a declaration-free language to write dots on a raster display. iMPaCT, which is an approximate acronym for Media-Propelled Computational Thinking, which fairly reflects our ambitions – that engagement with graphical programming will propel students towards exploration of computational thinking.

While iMPaCT-STEM is a work-in-progress, there is sufficient teaching material and evidence of its effectiveness to motivate further efforts to replicate, extend and more deeply examine its pedagogy. Figure 1 is a prerequisite graph of iMPaCT learning modules, most of which are being incrementally translated to use the programming environment provided by the TI 83/84 calculators that are already present in many high-school and college classrooms,

This paper relates iMPaCT's pedagogy to long-standing challenges in math education, recent educational reform objectives, and describes efforts to integrate iMPaCT's pedagogy into high school and college math and science courses. We conclude with early results from an ongoing longitudinal study that examines the influence of early participation in iMPaCT upon academic trajectory
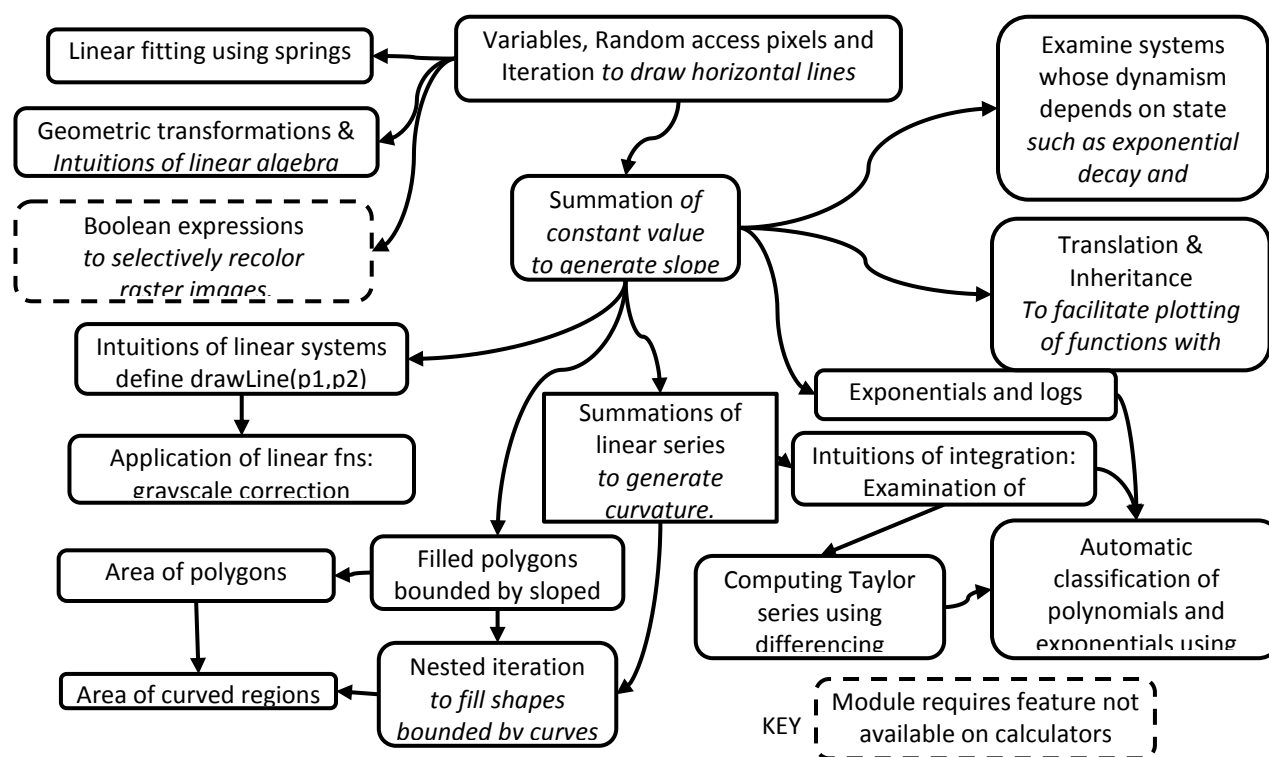

Figure 1. Flowchart of iMPaCT topics

2. An Example of iMPaCT's Pedagogy

This section provides an overview of lessons from iMPaCT that explore the behavior of polynomials that are being integrated into an existing pre-calculus curriculum at the University of Texas at El Paso. Most of the class and homework consists of conventional problem sets that examine and manipulate algebraic equations and properties of functions. iMPaCT modules are primarily used to convey key concepts in an intuitive and disarming manner prior to their examination using traditional frameworks.

The principal observation underlying iMPaCT's lessons related to polynomial functions is that the rate of change of a polynomial of degree n is a polynomial of degree n-1 is more accessible to students than

the original polynomial, especially in low degrees. Furthermore, behaviors of polynomials are more accessible when examined in the context of their rate of change, especially for first degree polynomials.

As described below, iMPaCT lessons build understandings of higher degree polynomials from exploration of lower degree polynomials. These lower degree polynomials effectively integrated through summation. Additional lessons have been developed that examine properties of exponents, logs, trigonometric functions, linear algebra, and geometric transformation which are not described in this paper.

2.1 Linear Functions

After a twenty minute initial introduction to the programmable calculators, students construct programs that compute first degree polynomials by summing constants as illustrated by the program LSUMMATION in Figure 2. Here, the behavior of linear functions with positive, zero, and negative slopes (represented by the variable R for "Rise") is completely transparent, and most students are able to modify this program to draw lines with increasing or decreasing inclinations without assistance. Through a succession of discovery-learning exercises, students learn to "connect the dots" and develop the conceptual understandings of algebraic expressions they had manipulated but not understood.
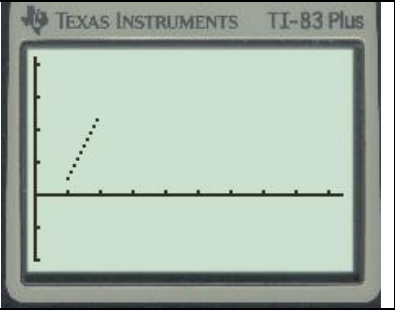
Summation	Direct Algebraic	Graphical output
PROGRAM:LSUMMATION	PROGRAM:LALGEBRAIC	
:10→X:5→Y:2→R	:10→X:−15→B:2→M	
:While X < 20	:While X < 20	
:Set−Pt(X,Y)	:Set−Pt(X,M*X+B)	
:X+1→X:Y+R→Y	:X+1→X	
:End	:End	

Figure 2. Rendering lines using summation and direct algebraic computation on a TI-83/84 calculator

This progression ends with the students discovering the principles underlying standard algebraic forms such as illustrated by the program LALGEBRAIC in the same figure's right pane which, like LSUMMATION, they are able to manipulate in order to achieve desired ends. This sequence differs dramatically from our earlier attempts to start with a direct algebraic structure. While the students recognized *y=mx+b* as an equation for a line, they demonstrated rote rather than conceptual understandings. For example, when asked to "make the line steeper", students relied upon fading memories of which variable was responsible for setting slope rather than determining this from first principles. The root cause appears to have been students' shallow understandings of algebra including the semantics of multiplication —-which our summation-first approach reinforced before extending.

2.2 Higher degree polynomials

Lessons examining second and third degree polynomials are built upon lessons learned when examining the behavior of polynomials of one lower degree. As illustrated by program PSUM of Figure 3, quadratics are introduced as a solution to the challenge of modifying a program that renders a line to instead render

a curve. Through guided discussion, students are led to achieve this goal by incrementally changing the curve's "slope," through repeated addition of a constant "slope acceleration" rate which also corresponds to ballistic acceleration. Students can be led to easily understand a variety of key attributes of quadratic functions such as the characteristics of slope acceleration that result in convex or concave curves, and manipulation of inflection points through engaging exercises such as the rendering of symmetrical or intersecting parabolas depicting smiles or frowns, or open mouths.
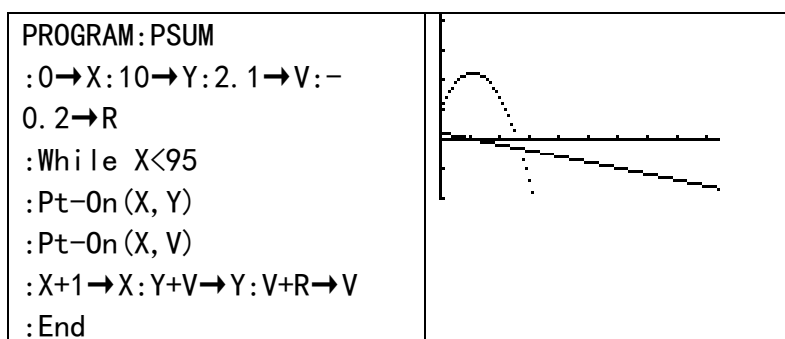


```
PROGRAM:PSUM
:0➜X:10➜Y:2.1➜V:-
0.2➜R
:While X<95
:Pt-On(X,Y)
:Pt-On(X,V)
:X+1➜X:Y+V➜Y:V+R➜V
:End
```

Figure 3. Generation of Parabolas on a TI-83/84 Calculator

By the time students examine parabolas in iMPaCT, they will be comfortable with linear functions and will recognize that for PSUM, V is the step-size of Y, that V changes linearly, and in particular, V = -0.2 X + 2.1. Furthermore, they should be able to compute that V is negative for X < 10.5 and positive for x > 10.5. Since V is Y's step size, they will be prepared to relate the plots of V and Y, and will infer that Y's maximum value should be reached when V is zero and therefore when X is approximately 10.5. This builds intuitions related to minima/maxima, inflection points, and regions of increasing/decreasing value.

```
PROGRAM:PALGEBRAIC              PROGRAM:SALGEBRAIC
:0➜X                           :0➜X
:-0.1➜A:2➜B:10➜C  # polynomical  :0.1➜A              # slope
coefficients                   acceleration
:While X<95                    :10➜P:20➜Q    # vertex
:Pt-On(X,A*X^2 + B*X + C)       :While X<95
:X+1➜X                         :Pt-On(X,A*(X-P)^2+Q)
:End                           :X+1➜X
                               :End
```

Figure 4. Alternative algebraic approaches to generating same parabola as PSUM (above)

Core learning outcomes of pre-calculus courses include the ability to characterize such functions as second degree polynomials as illustrated by program PALGEBRAIC and which also can be computed algebraically from inflection points as in SALGEBRAIC in Figure 4. Both of these programs render the same parabola as PSUM of Figure 3. These concepts are straightforward for students who understand the relationship between the parabolas and its changing slope – which are directly coupled in iMPaCT and might otherwise require knowledge of the fundamentals of differential calculus.

Figure 5 provides a accessible geometric proof that the sum of a sequence of linearly changing terms is quadratic and is used to link student understandings of simple curve generation to quadratic functions and their standard form $Y=a(X-p)^2+q$ where (p,q) is a parabola's vertex. Subsequent lessons compute and render the equations for parabolas previously examined by summation as is illustrated by programs PALGEBRAIC and SALGEBRAIC. Observe that this parabola's vertex is at X=10, which differs from student's anticipated value by 0.5. This difference is due to the Euler approximation method implemented in the program PSUM, and provides a useful lesson regarding modeling error that can be used to motivate descriptions of both calculus and numerical methods.
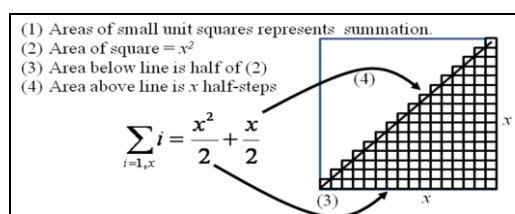

**Figure 5. Graphic from lesson examining relationship linear
acceleration and quadratic functions**

The same process is repeated to examine the properties of higher degree polynomials. In this context, concepts like minima and maxima are applications of concepts already explored and thus become opportunities to practice their application thereby building competence and confidence.

3. How iMPaCT-STEM addresses challenges in mathematics education

Integration of procedural fluency and conceptual understandings related to math are critical for success in quantitative STEM studies and notoriously difficult to achieve.

This problematic relationship is reflected in the evolution of learning objectives defined by the major efforts to reform math education. Gleason and Hughes-Hallett [2] identified three procedural fluencies related to manipulating and interpreting (1) algebraic (symbolic), (2), graphic (geometric), and (3) computational (numeric) representations of mathematical concepts. While the designers of this "rule of three" framework explicitly described relationships that students should understand among these various representations, few students developed the strong conceptual understandings the framework's designers intended.

This observation resulted in the addition of a fourth "verbal" skill defined as the ability to describe relationships among those proficiencies. The current "five intertwined strands" model of mathematical learning, [10] in addition to explicitly specifying procedural fluency, and conceptual understanding, includes two additional strands that reflect reflective depth: "Strategic competence" roughly corresponds to the ability to design and choose solutions and "adaptive reasoning" roughly corresponds to the ability to design and justify solution strategies. Together, they create an opportunity for students to develop an authentic"productive disposition," which is an affective trait related to self-efficacy that results from a composition of fluency at applying standard computational techniques with a deep understanding of how and why they can be applied to solve a range of applied problems.

While these frameworks describe comprehensive learning objectives, that integrate procedural and conceptual understandings - they do not prescribe how these integrated understandings should be achieved.

Our instructors observe that many students possess fluencies at manipulating and even converting among foundational math representations such as algebraic formulae for and graphical interpretations of lines and parabolas without underlying conceptual understandings. Our collaborators who teach math in college and high school indicate that students are invested in their ability to memorize these procedures and are uninterested in examining underlying principles, which are essential for success in advanced STEM coursework and careers.

As observed by Ron Aharoni, [1] it is difficult to build advanced conceptual understandings upon a foundation of rote procedure, and instead students must be led to examine and internalize each operation's multiple meanings. By using an easily mastered computer language as specifications for simple numerical computations that students understand to illustrate foundational principles underlying mathematics, iMPaCT exploits programming towards the objective of eliciting conceptual understandings of foundational math. Furthermore, the expression of computations as a short imperative programs facilitates metacognitive processes – for example, determination of whether an erroneous output is due to a clerical or conceptual error, and relating that error to the programs resulting (generally graphical) output.

The raster-based graphical output of students' programs -- though primitive -- is sufficient to engage students in these activities, to give them immediate visual evidence of misunderstandings, and -- more importantly -- a tangible sense of these fundamental phenomena -- which seems to have been the key element missing from their memorization-based approach to the subject. Furthermore, instructors report that college students have literally squealed with delight as they discovered the simple principles underlying familiar but opaque algebraic formulae they had memorized how to manipulate.

4. Transition from iMPaCT to iMPaCT-STEM

The majority of iMPaCT modules were initially developed in Jython using a modified Jython Environment for Students, [3] and many are in the process of being adapted to instead utilize the programming functions of graphing calculators already present in many high school and college math classrooms.

In Spring 2010, two high school teachers were recruited to examine the potential effectiveness of iMPaCT curricula in High Schools. As described in [7], these lessons provided a discovery-learning lab experience complementing a STEM course that conventionally does not include programming such as is illustrated by the ballistic simulation of Figure 3.

The limited availability of computer workstations in high schools initially limited the dissemination of iMPaCT to classes with access to computer labs. The present iMPaCT-STEM project was born from the realization that, as illustrated throughout this paper, the BASIC interpreter within commodity graphing calculators already present in high school and college math classrooms is suitable for implementing most IMPACT labs.

Pilot experiments of iMPaCT-STEM in high school classrooms began in Spring 2010. During the Spring of 2011, a section of pre-calculus at the University of Texas at El Paso has incorporated iMPaCT's lessons. While the number of students affected by this pilot study is small, early outcomes indicate a potential of substantial impacts including:

- **Improved academic outcomes for students who have repeatedly failed Algebra 1.** iMPaCT was integrated into multiple high school math courses including a remedial section of 9th grade algebra. Several students were repeating this course for the second to fourth time. District-wide testing indicates that by mid-year, almost all (7 of 8) of these students developed sufficient algebra skills to pass the course and state exam on linear functions.

- **Increasing enrollment of both genders in computing classes.** Approximately 80 of the high school students attended math courses incorporating iMPaCT-STEM during the 2010-2011 academic year were eligible to enroll in AP-CS during the 2011-2012 academic year. One quarter of these students enrolled in AP-CS for the 2011-2012 academic year. 40% of these students were female. This contrasts starkly with several preceding years, for which AP-CS was cancelled due to low enrollment.

4.1 A sample iMPaCT-STEM Lesson

A high school teacher developed a "Transformations" module in which students are provided a program on their TI calculators that draws a house (see Figure 6). Students are asked to translate the whole house in various directions by altering the program code. It is important to note that different parts of the house are controlled by different loops within the code, and in order to move the whole house, students must alter more than one line of code. Students then record the operations needed to move the house to its new location and generalize a transformation based upon observation, allowing for a deeper and more meaningful interaction with the concept.

Through minor but pervasive changes to the program such as translation and dilation students are introduced to concepts underlying domain and range transforms. These labs are tied to class lessons examining the relationships between changes in domain versus range, and are introduced to algorithmic logic and program tracing in a familiar setting. These transformations are considered to be a major concept in the NCTM national standards, appearing in both the Algebra and Geometry strands as a requirement for all 9th to 12th grade students in the United States.
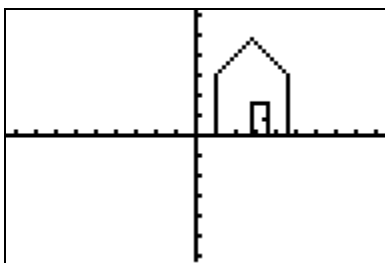

Figure 6. Example from transformations module.

As described in [6], other curricular modules examine the principles underlying ballistic motion, physical and electrical resonance, and other dynamic phenomena.

5. Evaluation

Our expectation is that students who attend iMPaCT will have:

- Increased competency and confidence at "computational thinking tasks" including programming and math concepts.

- Relevant experiences applying math applied towards problems that include computational thinking early enough to affect choices regarding academic major and career.

- Greater success in subsequent coursework that includes mathematics or quantitative reasoning.

There is a growing body of evidence that these expectations are being met. This section briefly details evaluation results collected from spring and fall 2010 semesters of iMPaCT taught to college freshmen at the University of Texas at El Paso, an urban predominantly Hispanic serving institution serving the El Paso – Ciudad Juarez bi-national metropolitan area. Additional information is available in evaluation reports on iMPaCT and the Computing Alliance of Hispanic Serving Institutions, of which we are a member. [15,13]

5.1 Perceived computational ability

Following the course, the evaluation surveys asked students to report if they were able to complete computational thinking tasks explored in iMPaCT. Students described whether or not the course positively affected their ability to complete named tasks. Results illustrated in Figure 7 indicate that the course led to students' perception that the developed programming skills in particular—between 69 and 78% of students indicated they were better able to complete programming tasks listed following the course.
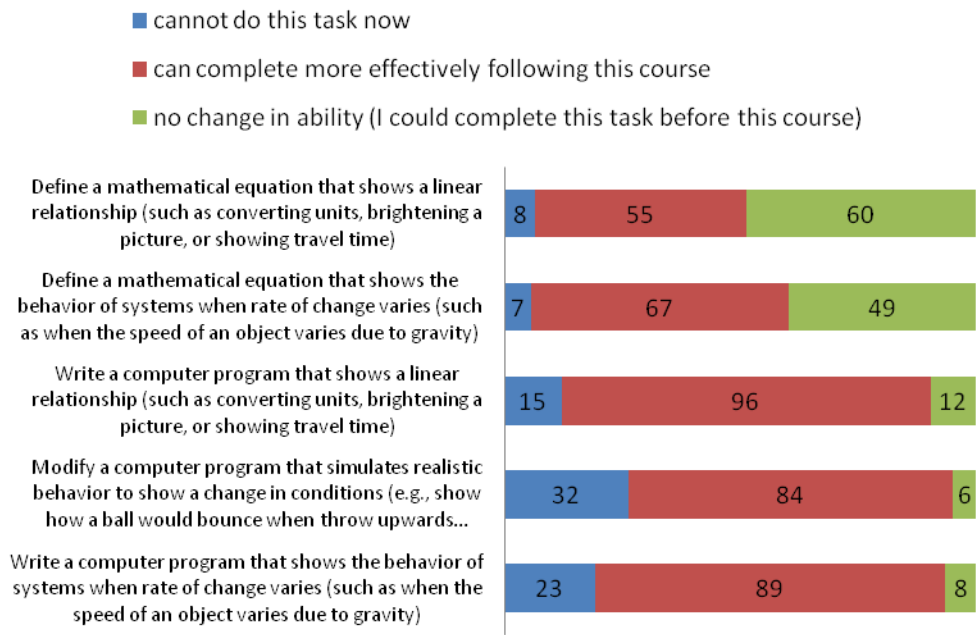

- ■ cannot do this task now
- ■ can complete more effectively following this course
- ■ no change in ability (I could complete this task before this course)

Task	cannot do this task now	can complete more effectively following this course	no change in ability
Define a mathematical equation that shows a linear relationship (such as converting units, brightening a picture, or showing travel time)	8	55	60
Define a mathematical equation that shows the behavior of systems when rate of change varies (such as when the speed of an object varies due to gravity)	7	67	49
Write a computer program that shows a linear relationship (such as converting units, brightening a picture, or showing travel time)	15	96	12
Modify a computer program that simulates realistic behavior to show a change in conditions (e.g., show how a ball would bounce when throw upwards...	32	84	6
Write a computer program that shows the behavior of systems when rate of change varies (such as when the speed of an object varies due to gravity)	23	89	8

Figure 7. Self assessment of confidence and ability (n=120..123)

5.2 Students' experience of the course

"We sometimes used mathematics to make a graph. This class requires more thinking and analysis to make things work and in math we just have to memorize and understand equations to solve a problem."

"(IMPACT is) fun, more applicable. I like how I can use my math skills as a tool to make something."

"(IMPACT is) more of a test and see what happens, which is more effective in learning."

The quotes above are typical of students' responses to the IMPACT course in comparison with typical mathematics courses. The responses indicate that iMPaCT is eliciting active engagement in analytical problem solving that is qualitatively different than their engagement in math courses. In fact, most students recommended the IMPACT approach for learning about computing. Overall, the majority of students (92, 81% of post survey respondents) would recommend the approach to learning computation concepts used in IMPACT. Similarly, the majority of students would recommend the Jython programming approach for learning about systems that change (94, 82% of post survey respondents). See Figure 8 and 9.

5.3 Student engagement with computation

iMPaCT holds promise as a gateway for broadening the accessibility of computational thinking. While iMPaCT's mathematical problem-solving focus was designed to respond to the proclivities of intending engineers and computer scientists, [4] surveys examining attitudes towards programmed computation indicate that, independent of gender, ethnicity, or intending major, approximately 75% of attendees express interest in attending subsequent courses that include quantitative analysis. The sampling of liberal arts students who attended iMPaCT sections of a required "University Studies" course was not influenced by prior interest in computation; Students in these sections were, at the time of course selection, unaware of the course's focus on computational thinking, and scheduling was the principal motivation for selecting iMPaCT sections. See Figures 8 and 9.
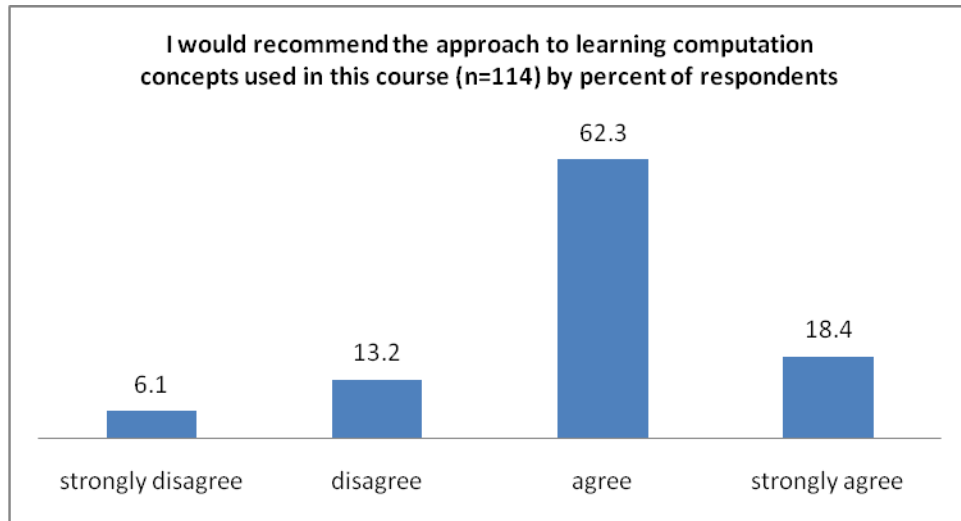
Figure 8. Student recommendations regarding iMPaCT approach to learning about computation.
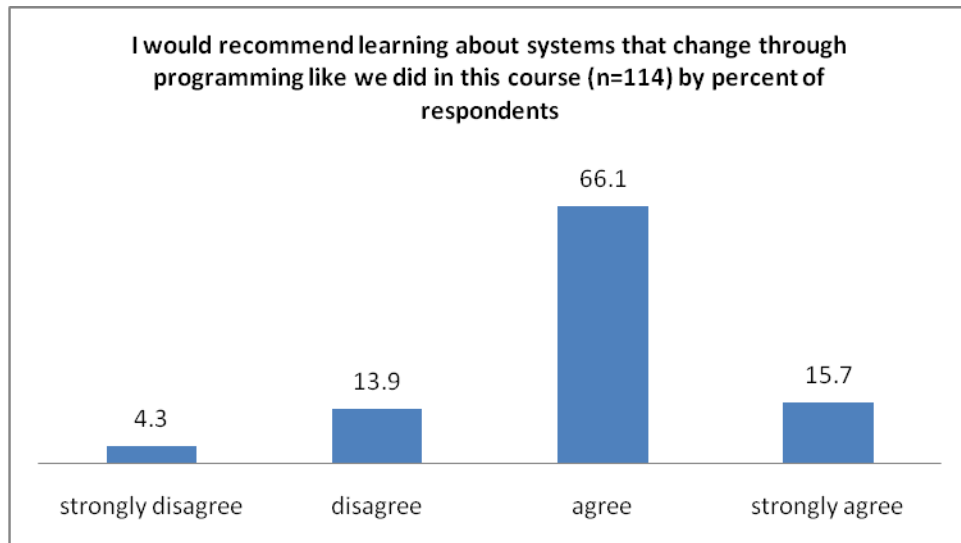

Figure 9. Student recommendations regarding iMPaCT approach to learning about dynamic systems.

5.4 Impact upon academic success in math courses.

In spring 2007-summer 2010 semesters at UTEP, 2,760 students enrolled in of Math 1411 (Calculus 1) for the first time. Fifty eight percent passed the course with a C or better on the first attempt, twenty-two percent passed the course with a "P" showing adequate progress (with additional semesters needed to complete the course), and 20% withdrew, or earned a D or F on their first attempt at calculus. These rates were established as "typical" for UTEP over the past four years. In contrast, of the 58 IMPACT students who subsequently enrolled in MATH 1411 for the first time, 11 more students passed MATH 1411 with a C or better than would be expected, given the results of their peers in the same course enrollment semesters. Six fewer students withdrew or failed the course than expected, and five fewer students passed the class with a "P" than would be expected given course patterns. These differences

were statistically significant $\chi 2(2,N=58)=8.44,p=.015$. Chi square analyses were 3X2, with "Progressing/Pass(P)"; "Graded Pass(A,B,C)", and "fail/withdrawal(D,F,W)" options seen as separate outcomes for students. Given the importance of mathematical thinking in the development of computational thought processes, it is clear that supporting mathematical conceptual understanding may be a necessary step in fostering students' computational thinking ability.

6. Conclusion

Early results from the formal evaluation of semester-length iMPaCT courses indicate that they have dramatically affected the nature of student engagement with problem solving using mathematics and improve academic success in concurrently and subsequently attended math courses. Preliminary outcomes from the high school pilot study suggest that embedding computation within high school mathematics courses can dramatically increase the number of students, including women, who choose to study computation. While iMPaCT-STEM is a work-in-progress, there is sufficient teaching material and evidence of its effectiveness to motivate further efforts to replicate, extend and more deeply examine its pedagogy. More information about iMPaCT-STEM is posted online: http://sites.google.com/site/impactstem. Lesson plans are released at no cost to educators who contact this paper's first author.

References

[1] Aharoni, Ron, Arithmetic for Parents, Sumizdat, 2001
[2] Connally, Hughes-Hallett, Gleason et al. "Functions Modeling Change - A Preparation for Calculus. Wiley; 3rd edition, 2006.
[3] Guzdial, Computing and Programming with Python, a Multimedia Approach, Prentice Hall, 2006.
[4] Freudenthal, Eric, Roy, Mary K. and Ann Q. Gates, Work in Progress – The Synergistic Integration of an Entering Students Program with an Engaging Introductory Course in Programming, Proc, Frontiers in Education, Fall, 2009.
[5] Freudenthal, Eric, Roy, Mary K., Alexandria Ogrey, Tanja Magoc, and Alan Siegel, *IMPACT – Media Propelled Computational Thinking*, Proc. Annual Symposium of the Special Interest Group on Computer Science Education (ACM SIGCSE), 2010.

[6] Freudenthal, Eric, Ogrey, Alexandria N., Mary "Kay" Roy, Sarah Hug, and Gonzalez, Rebeca, *A Computational Introduction to STEM Studies,* Proc. ASEE Annual Conference, 2010.

[7] Freudenthal, Eric, Alexandria Ogrey, Gonzalez, Rebeca, Work in progress – Eliciting Integrated Understandings of High School Stem Curricula Through Programming, Proc. FIE 2010.

[8] Gonzalez, Virgilio and Freudenthal, Eric, Work in progress: Adoption of CCSO computational methods and circuit analysis techniques into an introductory programming course for electrical engineers, in 2010 ASEE Annual Conference and Exposition, June 20, 2010 - June 23, 2010, Louisville, KY, United states, 2010.

[9] Hug, S., Thiry, H. & Weston, T. (2010),. *CAHSI annual report: Recruiting, retaining and advancing Hispanics in computing*. (Report to CAHSI/SACI). Boulder, CO: University of Colorado at Boulder, ATLAS Institute and Ethnography & Evaluation Research.

[10] Mathematics Learning Study Committee, *Adding It Up: Helping Children Learn Mathematics*. Jeremy Kilpatrick, Jane Swafford, Bradford Findell, Editors, National Academy of Sciences, 2001.

[11] Kalman, *Elementary Mathematical Models*, Mathematical Association of America (Press), 1997.

[12] Tanja Magoc, Freudenthal, Eric, and Modave, Francois, *Computation for Science and Engineering*, EDUCON (IEEE), 2010.

[13] Thiry, Barker, and Hug, *CAHSI Evaluation Progress Report, The Computing Alliance for Hispanic Serving Institutions,* 2009, http://cahsi.cs.utep.edu/Portals/0/2008InterimEvaluationReport.pdf

[14] Suskavcevic, Kosheleva, Gates, and Freudenthal, Preliminary Assessment of Attitudes towards Mathematics for a Non-STEM Section of Computational Computer Science Zero, UTEP CS Technical Report UTEP-CS-09-13, 2009.

[15] Soloway, Elliot, Bonar, Jeffrey, & Ehrlich, Kate. (1983). Cognitive strategies and looping constructs: an empirical study. CACM, 26(11), 853-860.