Mathematics + Coding Symposium, Western University, 19-21 June 2015

Working Group: Mathematical Coding Tasks at All Levels of Education

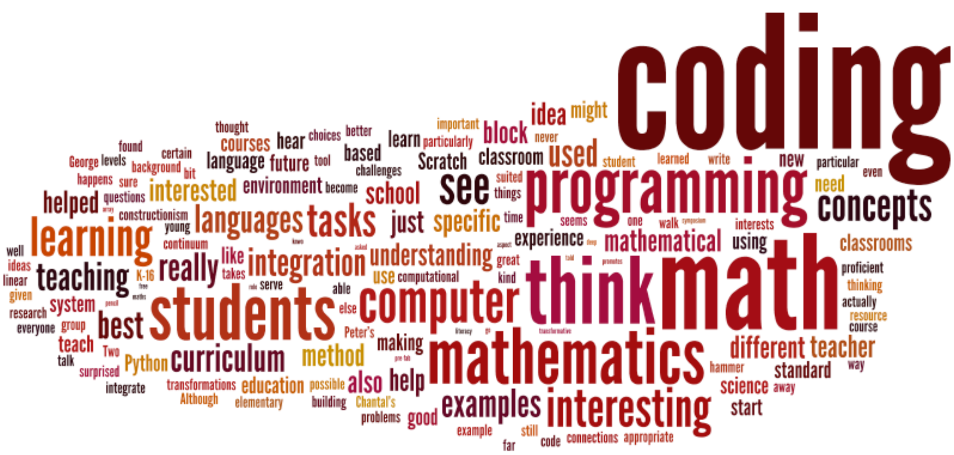
Prepared by Asia Matthews, Andrew McEachern & Diane Tepylo

Introduction

This is a report on the activities of a group interested in tasks involving mathematics and computer coding. Discussions about the integration of coding and mathematics apply to special types of courses but also generalize to a broader issue in education. As technological aptitude becomes more relevant, basic coding literacy naturally pairs well with mathematics through logic and creativity. The discussions pursued in this working group followed a general theme: coding tasks with sophisticated underlying mathematical ideas.

Summary

- The truly relevant question: "Why have coding in a mathematics class?" was visited in a variety of contexts. Different perspectives require different answers to this question, but ultimately the bottom line is the collective belief that activities involved in coding can allow for enhanced learning of difficult mathematical concepts.
- The choice of programming language can have a powerful effect on the trajectory of the curriculum.



- Having excellent mathematical coding tasks to get an enhanced understanding of mathematics can be a good way to begin. Engaging problems that both encourage algorithmic thinking and allow for creative mathematical exploration may define a new mathematics curriculum.
- When considering adding coding to a mathematics class, learning objectives need to be clear, but not limiting. This will help to define the resources that are needed and to drive the selection of tasks that allow and encourage exploration.
- The history of mathematics and coding needs to be considered, whatever course is decided for the future. The work being done in the UK at the moment is an excellent beginning, and we can use the knowledge gained from that endeavour to support and enhance any action taken in Ontario/Canada.

- There is a concern that younger children will simply be unable to handle the rigours of coding, and yet it was demonstrated that children of a very young age can and do think in an algorithmic manner. The real obstacles to getting coding in a classroom are not the students, but the resistance to change in the curriculum at all levels, and a lack of support for instructors. Getting educators from different disciplines involved will help the idea of coding in the classroom gain some traction.
- Creating a curated instructor resource would be to everyone's benefit – one that allows access to a variety of problems, and provides written and even video-recorded assistance to help with implementation in the classroom. One result of this working group is the creation of such a resource.
- Several difficult questions were asked and not answered during the working group session. Some of these questions are at the end of this document. Pedagogical and logistical concerns are at the core of most of them, and a great deal of thought will be required to get satisfactory answers.

We were fortunate to have some distinguished participants in our group who had experienced the origins of the movement to integrate mathematics and coding through Papert's LOGO. Celia Hoyles and Richard Noss shared some historical projects, stories and issues and discussed the 're-birth' of programming in school (math) classrooms. Many of the participants expressed interest in learning more of the history of programming and constructionism in Canadian mathematics education. What happened here in Canada/some provinces in Canada?

Many other participants came to the group with very little or no coding experience. These working group discussions went a long way in exposing these participants to the many issues and ideas in mathematics and coding.

> "I don't have a background in using coding in the classroom, so it was really interesting to hear what everyone else was doing. In particular, I had no idea so many people were using Python or that it had good math packages. I plan to learn more about it and think about how I'll integrate it into the classroom in the future. Also, although I have no experience with elementary math education, it was neat to hear what others are doing with coding and young children!"

Several themes emerging from our discussions are highlighted below, including the role of coding in mathematics, possible tasks, coding languages, key concepts, learning outcomes, curriculum and systematic changes, and resources (for teachers and researchers). This session provided a forum for asking questions and discussion possible answers. Many questions remain and few of these are included at the end of this report.

The Role of Coding in Mathematics

Throughout our discussions, the question, "Why coding in mathematics class?" was often revisited in a variety of contexts. The following reasons were offered in support of using code in a mathematics class. Coding can help make abstract concepts tangible, and can allow for exploration in the abstract in a way that was previously unavailable. Coding also helps develop computational thinking and algorithm design, which are useful skills in any context. Being able to explore math dynamically, making changes in the math and immediately seeing the results, coding can give students control over their mathematical learning by allowing students to decide

(within limits) what they want to explore, deciding what changes in code they want to make to code and observing the effects. It helps students build a rich understanding of important mathematical ideas (followed by a discussion of the need for greater clarity of the meaning of constructionism and how it differed from constructivism).
The group offered a few cautions about coding in mathematics. We must be very careful that coding tasks are not designed like the word problem which injects an artificial context. The act of coding should not apply a thin patina of mathematics atop a computational problem, but should add to the learner's mathematical understanding. We need to think about the math in coding tasks:

> "Honestly, sometimes I'm a little bit confused about the connections between mathematics and programming. Some of the tasks like the linear equation transformations seems to be just a math concept, and some are just computer language concepts. Maybe the connections between math and computer programming should be learnt to help students understand math better."

Tasks

Quite an interesting array of examples of tasks was presented by working group participants. These tasks seemed to fall into three categories:

1. Tasks in which coding is used as an instrument for gaining a deeper mathematical understanding. Included here would be activities that bring about visualizing, simulations and models, iterations.
2. Tasks that build competency with mathematical abstractions and conventions. Included here might be tasks that use computers ability to "make math concepts such as 2x-1 tangible (Gadanidis, 2015). Tasks here would build competency with variables, formulas, and the coordinate system.
3. Tasks in which mathematical and logical thinking is used for making more efficient code.

Coding Tasks, Mathematics and Young Children

George shared some of his work with young students showing that primary students are able to develop showing understanding of complicated mathematical ideas though coding and interactive computational thinking activities. For example, the teacher says GO, the students claps 'if' they are boys, and 'else' (if they are girls) they say GIRLS RULE. It is encouraging there has already been thought given to "unplugged" approaches as well.

By tossing coins, George used coding and graphics to construct Pascal's triangle. Although he presented it at a level suitable for a younger audience, this sort of technique could be easily scaled to a level appropriate for older students. The technique of connecting graphics and code to a given mathematical object with interesting features is one that needs to be explored more fully. Another group of elementary students explored the resulting pattern if a pencil was attached to the edge of a circle as it rolled, a math topic Asia explored in her undergraduate mathematics thesis.

Lisa shared some of her lessons with intermediate students. These lesson used coding to develop student competency with variables and measurement formulas through coding. The group discussed how, in learning to code, concepts such as variables and coordinate systems

> “In my classroom CS experience, I have seen that the coding is building math concepts such as understanding the coordinate system and variables, and I have learned to push that learning outside of computer science expectations.

Generating Ideas for Tasks

The group generated ideas for tasks that could help students to build a deeper understanding of specifically identified mathematical concepts, such as developing algorithms.

A large amount of coding activity rests on computational thinking- developing detailed efficient steps to solving a problem- developing an algorithm. Coding is generally supposed to make calculations and predictions easier to check as well. For example, predicting which day of the month a date will fall on in the future, or predicting the number of Friday the 13ths in a given year are tasks in which mathematical thinking is the key to solving the problem, not computational proficiency. That proficiency allows the explorer to check their work quickly, but not much else. Some other examples offered are:

- Fisherman problem: The problem asks students to figure out how many fish each fisherman on a boat will get if they share the catch equally, and give any remaining fish away, essentially developing the long division algorithm.
- Changing time on a 24 hour clock to time on 12 hour clock.
- Check if an integer is prime- students are motivated by the fact that mathematicians do not yet have a concise method for check if an integer is prime. The power of the computer makes dealing with large numbers less daunting.

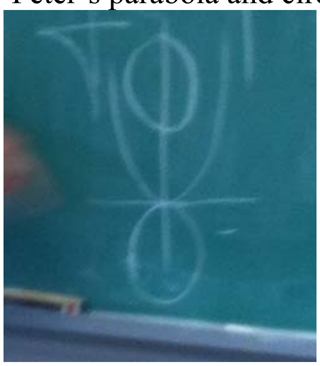
Another interesting problem was posed: Suppose I picked a random direction to walk in 2 dimensions, and this direction changes randomly at a given time step. If I am inside of a box, or other shape, how many steps will it take for me to reach a given edge? Coding for this sort of task might help a student gain insight into how this sort of thing might evolve over time. This sort of problem leads naturally into creating simulations and modifying models, for different phenomena such as traffic, ecosystems to explore interactions, biological models, exploring election data and who would win in other electoral systems.

Learning to code to produce functions, in whatever software is available (Maple, Mathematica, etc.) is an important way for us to connect algebra and geometry. Peter offered some pictures of curves as fundamental building blocks of continuous functions, and ways to understand polynomials. He also drew a picture of a circle and a parabola occupying the same 2-dimensional plane. The task: as you change the dimensions or location of the circle, in how many places would the circle intersect the parabola? This represents an interesting coding task, both in constructing the circle and the parabola, and altering the circle’s properties to see what changes occur.

- Function design – create a composite function to recreate these curves

- Peter’s parabola and circle question



- transformations of vector graphics
    - Matrices and computer graphics – scaling, rotations, translations   better picture

Following up with coding used to represent and explore interesting mathematical objects, fractals like the Heighway Dragon and the Koch Snowflake are natural places where coding is a tool to enhance understanding, rather than being the primary task.

Coding can be used to help explore optimization problems, such as the Rugby problem, in which the task is to find the optimal angle to kick a rugby ball through the uprights, was shown to have a computational solution. As a point of interest, the best solution was found in an Excel spreadsheet.

Obviously these tasks need further development before use with students. It was agreed that an online resource sharing teaching- ready resources could benefit the community, one that included specific examples of how coding might be used to teach specific math concepts. Steps were taken to begin to create such a database. George Gadanidis offered to host the site. The idea of a curated source of tasks as a resource to help facilitate this also represents interesting opportunities and challenges.

Even with the cursory nature of this list of tasks, the participants appreciated the list of tasks as a way of broadening their vision of what math and coding might look like:

> I really appreciated the examples focused on teaching specific math concepts and helped clarify possible research ideas in this area."

General Teaching Strategies – pedagogical considerations

The group also discussed some general strategies for coding tasks to learn mathematics. First, provide the code for a mathematical task. Discuss the code, the output and then ask students to make variations on the code to achieve specific results. Use clear thinking to improve an algorithm. Students can be presented with code for an algorithm that achieves most of the desired functionality, with instructions for students to improve its performance. Alternatively, start from scratch. Begin with a discussion of what the program needs to do to solve a mathematical program, then let the students write the code independently or in groups. This is expected to be the most challenging for students.

Coding Languages

The choice of coding language can have a great impact on the sort of learning that will take place. This choice involves questions of teacher training, software and hardware implementation, language transition, choice of tasks, and a likely a variety of challenges not even considered as of yet.

The factors that affect the decision to choose one language over another are context based, and must be based on the following questions, most of which do not have an easy answer. Are certain languages better suited for one group or another, and why? Which languages are best suited for teaching and learning and how do we best take advantage of these for learning purposes? How do we prepare teachers who may have little to no coding experience understand and use coding languages for their future students (at all levels, K-16)? The following quote brings up another interesting and important question:

> "I am still torn on the block method. This seems to have really caught on and I am not sure that it is necessary. George said that he had grade 3s (I think) using Python. Why would we bother starting with the block method in school? Who is making sure the students transition from block method to line based coding? Although I think the block method promotes computational thinking, I do not think it is teaching coding."

Does it make sense to use one language and then another? If that is the case, how do we properly manage and prepare both teachers and students for the transition? After all of the discussions, it became very clear that different languages are more appropriate for different activities and outcomes.

> "As an elementary teacher I was greatly inspired by the resources, like Blockly, Scratch and even pre-fab programs that can be used to deepen a child's understanding of mathematics. I am particularly motivated to even see what adaptation I can make to bring some of the secondary school concepts into the mix in a manageable form for younger learners."

A lot of the discussion centered on Python. Python is a free and cross-platform coding language. It has a good deal of power and functionality, as well as an ever growing set of libraries and utilities relating to mathematics. Is Python the coding language of the future? Is that the right question to ask, since it may be replaced by another language within 5 years?
As students become proficient in block programs like Blockly or Scratch, the transition from those to programs that use lines of code must be handled carefully.

Desired Learning Outcomes

As part of the discussion, the question: "What is it we are actually trying to teach to students?" came up in a few different contexts. Defining the learning outcomes of such an initiative should be a priority, as that will inform the choices about both language and tasks. Regardless of the choice of language and tasks, in the best system students would acquire effective possession of a standard language (basic literacy) and that would serve them, with appropriate updates, for all courses to come. Part of constructing coding tasks for mathematical learning should result from considering the shifts and foci of attention when coding for maths. There is an idea that via coding we might be able to structure good problems such that they are accessible (albeit in different ways, with differing levels of sophistication / formality / etc.) across a variety of ages and backgrounds. Some of the discussion involved learning a bit, but not enough, about how to think of what's actually learned when we use programming to teach math. For instance, Chantal's checklist of objectives/competencies related to programming is a start. Having a starting template to help us think about learning objectives, what's possible and what's not in a programming environment would be very beneficial.

The following quote sums up some of the concerns about learning outcomes:

> What interested me was how the choice of programming language affects the trajectory of the curriculum. The challenges of making coding a part of the math curriculum are daunting, but the rewards could be incredible and far reaching. We could have generations of proficient students trained to think clearly, with tools to explore new concepts we would never think of otherwise. That makes it worth the attempt, at the least.

Changing the System

As it stands, integrating coding into the curriculum seems far from reality at this point. Many problems and possibilities were shared with an eye toward the future of teaching and learning (K-16). Sharing tasks that embedded coding is one way to start to think about the role of the computer tool in shaping the knowledge and the pedagogy, but leads to more questions that don't have simple answers. Why was the task chosen? Was it particularly motivating? Surprising? What did the students learn? Not all of these will have answers that are satisfactory to every person asking them. It is interesting as well to consider what the instructor learns about mathematics and his/her students in the process of design and implementation and revision. As to how to implement coding into the curriculum continuum it remained unclear as to how things could or should change. Local 'successful' integration in some classrooms could serve as exemplar. But then it is mostly limited to at the most one-year. Coding learning for mathematics or in mathematics takes time. To see the benefits, and build on it, takes even more time. It's a vicious cycle. Where to start?

Given that all of the hurdles, both educational and logistic, could be surmounted, there could be a drastic change in teaching that would in a sense that would free not only the student but the instructor as well. Given that environment we would soon see that the way we teach standard courses that we've been teaching all our lives, calculus for example, could be transformed. Should they be transformed?

Ultimately, to get anything to happen, we need to get our colleagues on board!

Questions

- How can we really 'up the standard' of the good / deep / rich mathematics pupils engage with, through the use of coding, and given the realistic constraints of educational systems and structures in Ontario/Canada/beyond?
- When math students (in the undergraduate level) can walk away with a computer program, what does a computer science student walk away with?
- What kind of teachers do we need for math students, a math teacher or a computer science teacher, or some kind of combination that usually doesn't exist?
- When assessment happens, should it be more mathematical or computational based?
- The connection between coding and mathematical thinking is interesting; by learning them concurrently, how does that look as mathematical proficiency? Creativity? Are we limiting them by giving them one really interesting hammer? Or is this hammer a new tool set in disguise?
- Chantal's presentation was a great example of what might be, and the questions asked helped me think through follow up research. If we ask student's 2 or 3 years later about their experiences, what is it that stays with them?