(Legitimate peripheral) computational thinking in mathematics

Laura Broley

Chantal Buteau
Brock University

Eric Muller
Africa University

(Legitimate peripheral) computational thinking in mathematics

Laura Broley[1], Chantal Buteau[2] and Eric Muller[2]

[1]Concordia University, Canada; l_brole@live.concordia.ca

[2]Brock University, Canada; cbuteau@brocku.ca; emuller@brocku.ca

"Computational thinking" is a hot topic in math education, among teachers whose curricula now include the term, and researchers who wish to pinpoint what it means and how it could be promoted in classrooms. A recent study resulted in a theoretical model of the computational practices of professional mathematicians and scientists, with the aim of offering teachers a set of competencies around which to build activities for their students. Nonetheless, concrete examples that validate the model and exemplify its use in math classrooms have yet to be discussed. We wish to open up this discussion, which we see as crucial to understanding how to empower students to participate in the computational thinking that has become integral to the mathematics community and beyond.

Keywords: Computational thinking, mathematicians' practices, legitimate peripheral participation.

Introduction

Parallel to the invention of the personal computer, Papert (1980) envisioned a world where children fluently use the tool as young mathematicians. Some thirty years later, we've witnessed a widespread resurgence of interest in that vision, taking shape in educational reforms (e.g., in Europe; Bocconi, Chioccariello, Dettori, Ferrari, & Engelhardt, 2016) and research regimes (cf., www.ctmath.ca) in the name of *computational thinking*, deemed a 21st century skill. Yet, there is little consensus on what this "new" term encompasses or how/if it should be conceptualized within subject areas beyond computer science (Grover & Pea, 2013). In response to these issues, Weintrop et al. (2016) developed a taxonomy of computational thinking practices geared towards science and mathematics. They based their work on a literature review, an analysis of learning activities, and interviews with "biochemists, physicists, material engineers, astrophysicists, computer scientists, and biomedical engineers" (p. 134). They also show, through concrete examples, how the practices might be promoted in physics, biology, and chemistry classrooms. To build on this work, we could ask: What might the computational thinking practices look like in *mathematics* classrooms? Moreover, are they representative of professional *mathematicians*' practices?

In this contribution, we attempt to provide some answers to these questions by drawing on two resources: 1) fifteen years of experience in a sequence of three undergraduate Mathematics Integrated with Computers and Applications (MICA) courses at Brock University, where students create and use computer environments to explore mathematics concepts or real-world situations; and 2) reflections of mathematicians whose research falls within an area recognized by the European Mathematical Society in 2011: "Together with theory and experimentation, a third pillar of scientific inquiry of complex systems has emerged in the form of […] modeling, simulation, optimization, and visualization" (p. 2). The next section outlines the perspective underlining our work and our approach in preparing this paper. We then present our results, i.e., we exemplify (empowering legitimate peripheral) computational thinking practices in mathematics.

Theory and methods

The way we interpret "Learning mathematics with technology" in the context of this paper is well explained by Lave and Wenger's (1991) concept of "legitimate peripheral participation", whereby students are invited to become mathematicians through engaging in their shared practices. "Mathematics", then, is not seen as a body of knowledge to be acquired by the student, but rather as a social community to which the student gradually gains membership. Hence, we do not discuss computer "technology" from a cognitive point of view, for example, as a helpful tool in illustrating concepts. We focus, instead, on how mathematicians, the old-timers of their discipline, and students, the new-comers, create and use computer tools to engage in practices considered to be integral to the mathematical community.

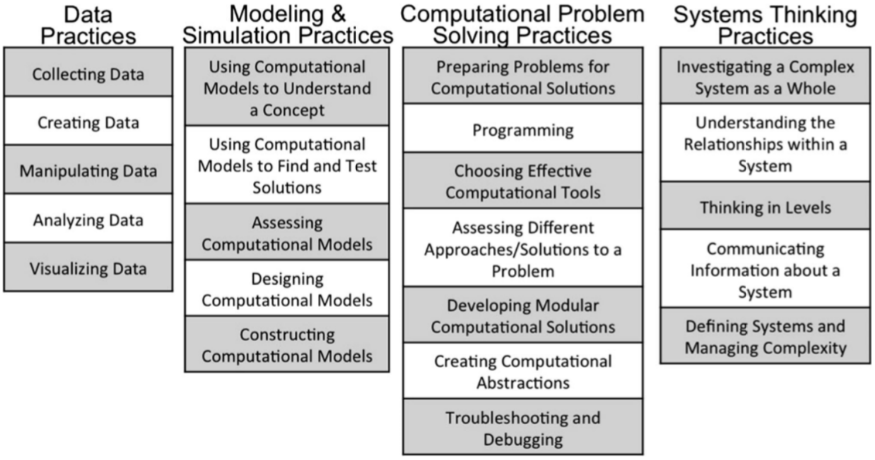

Figure 1: Computational Thinking Practices, taken from Weintrop et al., 2016, p. 135

In recent work, Weintrop et al. (2016) outline what they believe to be these integral practices (Figure 1). Their framework provides a detailed description, specific to STEM (i.e., science, technology, engineering, and mathematics), of one of three dimensions introduced by Brennan and Resnick (2012) to characterize "computational thinking": namely, computational concepts, practices, and perspectives. Such frameworks seek to elaborate on the general definitions on which they are grounded; for instance, that of Cuny, Snyder, and Wing, who describe computational thinking as "[t]he thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (2010). Computer programming plays a particularly important role, as it "is not only a fundamental skill of [computer science] and a key tool for supporting the cognitive tasks involved in [computational thinking] but a demonstration of computational competencies as well" (Grover & Pea, 2013, p. 40). Such ideas invoked in us vivid images of our own experiences with/as mathematicians and students creating and using computer tools to do mathematics. And this inspired us to address an apparent void: that is, a comparative picture that highlights the powerful computational thinking employed by professional mathematicians, on the one hand, and, on the other, the potential of students to participate in the same kind of thinking.

To prepare this picture, we re-examined Broley's (2015) research, which explored the use of programming by 14 mathematicians in their research and teaching. In an interview, each participant described research where they developed and used computer tools. Unbeknownst to us at the time, this provided examples of how some full-membership mathematicians engage in computational thinking. Amongst them, we chose four that align with the groupings in Weintrop et al.'s (2016) taxonomy. We then reconsidered the data from another study (Buteau, Muller, Marshall, Sacristán, & Mgombelo, 2016) – the 14 MICA projects completed by one student, Ramona – as a source of four examples of peripheral computational thinking. MICA's goal of responding to society's need for professionals proficient in programmable technology made it a natural database for comparison.

Results

This section provides examples of computational thinking as it might be experienced by full and peripheral participants in the mathematics community. For each category identified by Weintrop et al. (2016), we describe a mathematician's project that we feel effectively exemplifies it. This is contrasted to a MICA project that we see as providing access to the same kind of practices.

Data practices

Adèle uses her expertise in mathematics and computing to solve problems in financial engineering. In one project, she developed a model that enables investors to judge the investment potential of various market entities. In particular, the model calculates the risk that an investor will lose money because the investee is unable to pay back what they owe. The tricky part is that most investees have never had such financial problems (e.g., with bankruptcy). To assess a given company or individual, Adèle considers their portfolio: She *collects* their history of actions (e.g., investments, bonds, shares) on the financial market. The basic idea is that as others agreed to invest a certain amount of money in the company or individual, they implicitly demanded to be compensated for the risks they were taking, thereby predicting the probability that the investment would be a good one. Mathematically-speaking, the problem is of an extremely high dimension: Adèle's model contains over 20 parameters that must be estimated for each portfolio by *manipulating* the corresponding data with optimization techniques. Intensive numerical methods are then applied to the specified model to *generate* the data necessary for evaluating the risk of the investee(s) being considered. The result is not a simple measure of the average risk. Adèle must perform a nuanced *analysis* to meet her clients' needs, calculating and *visualizing* probability distributions in order to portray the best and worst case scenarios. About the place of computation in her work, Adèle was blunt: She said there would be no project without it. In fact, when tasked with assessing the risk associated to hundreds of companies at once, she must use computer clusters to get the job done.

In their second of three MICA courses, students in Ramona's cohort were assigned a project similar to Adèle's. During lectures, they were introduced to mathematical ideas related to the stock market. In regards to programming, they also learned how to read data from files. Up to this point, they had worked with data they *created* through simulation; but during this project, they had to use data from Stock Market sources. During two (two-hour) lab sessions, the students initiated their individual work by *collecting* the S&P index, a measure of market conditions, from 1950 to 2002, and writing a program to *manipulate, visualize,* and *analyze the data* using standard statistical techniques. Students were also required to select ten stocks and, like Adèle, make recommendations to a fictive client based on their own *analysis*. In her report, Ramona grounded her recommendation on the mean and average yearly percentage of her stock selection. Then, as requested, she conducted a regression analysis of a stock over a decade and described how *visualizing the data* as a cloud of points confirmed her interpretation of the coefficient as representing a weak correlation.

Modelling and simulation practices

Alice’s projects are often inspired by a collaborator in need of her modelling and simulation skills. She spoke, for example, of a kinesiologist who initiated a multi-year project about muscles. Alice began by learning about the application, which she knew little about. She could then *design* a system of equations that would allow her to study the features of interest, i.e., tensions, bulges, and fibers; but only once the model was implemented on a computer. During her interview, Alice joked that computation was essential because, unfortunately, the solution to a real-world mathematical model never simplifies to the quadratic formula. While some researchers use existing simulators to gain access to their models’ solutions, Alice prefers to have the control of *constructing* her own. This comes at a price: Even if her team starts with an existing code, they still have to think very hard about how they implement their equations, import data, generate meshes, and so on. But all this hard work apparently paid off in this project: Alice described the resulting computational model as “the most complex simulator of its kind”, and was hesitant to share its massive code during her interview. This tool was used systematically to investigate issues the researchers initially sought out to *understand*. But by varying parameters in an exploratory mode, they also *found and tested solutions* to an unreported problem: the forming of well-defined fiber structures. During her interview, it was clear that Alice was excited by this discovery, for her collaborator had observed the formation of the exact same fiber structures, but in a real human! In the end, the data collected during this ultrasound experiment of a person on a bicycle *assessed* Alice’s model, confirming that it represented “the real thing” in more ways than expected.

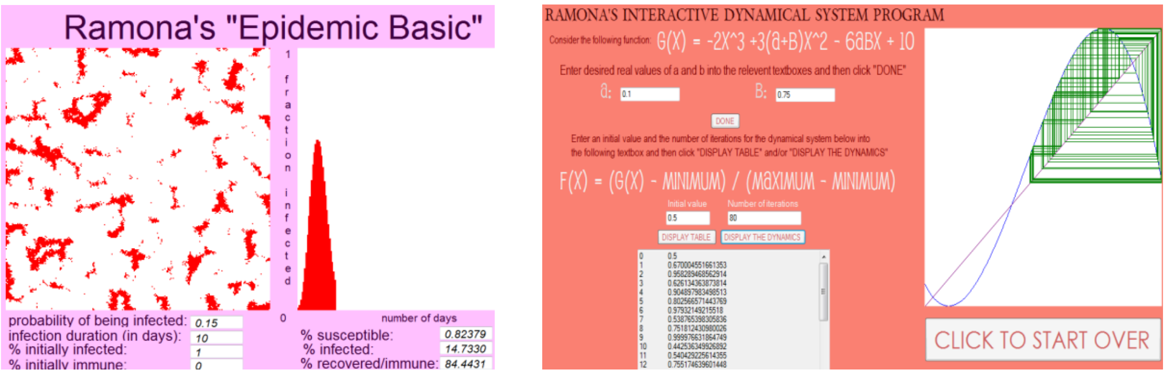

Figure 2: Ramona’s epidemic simulator (left) and discrete dynamical system program (right)

Modelling and simulation practices that resemble Alice’s are central to the MICA courses. At the end of the third course, students in Ramona’s cohort were asked to use the theory of cellular automata to model and simulate the spread of an epidemic. Students worked individually to *construct* the computational model (i.e., to implement it in VB.Net), complete with a dynamic visualization of the cellular automatum and a complementary graph (Figure 2, left). Using this model, students were invited to observe real-time simulations of certain scenarios with the goal of coming *to understand* the effects of vaccination on the proliferation and diminution of epidemics. They were then told how to extend their models to include the cost of immunization and medical treatment, so to *find* (estimate) *the solution* of a minimal medical cost problem. In her report, Ramona went beyond finding the solution; as required, she also *assessed* the ability of her extended computational model to provide an accurate estimate, finishing with suggestions for improvement.

Computational problem solving practices

To understand Norman's pure mathematics research, some *preparation* is in order. In his work, a permutation of length n is just a string, $\sigma = \sigma_1\sigma_2\ldots\sigma_n$, where each σ_i is a unique element from the set $\{1, 2, \ldots, n\}$; for example, $\alpha = 624531$ is a permutation of length 6. Given another permutation, e.g. $\beta = 231$, we say that $\propto$ contains the pattern β if we can find in $\propto$ a subsequence (not necessarily consecutive) whose numbers have the same relative order as 231. The fact that $\propto$ contains the subsequence 451 – 1 is the smallest number, 5 is the highest, and 4 is in between – means that it contains β (we could have equally used subsequences 241, 251, 231, or 453). If a permutation does not contain a pattern, it is said to avoid it; for instance, α avoids 1234. An interesting problem for mathematicians is to determine the number of permutations p_n of length n that avoid a given pattern. It is known that p_n grows almost exponentially with n. The growth rate, however, is still unknown for many patterns. In search of one such rate, Norman's team had to build a complex computer tool. The *programming* was delegated to a student, whose life was simplified by the *development of a modular solution* based on an existing subroutine for another pattern. The creation of the entire algorithm, nonetheless, was a team effort, for it involved the careful *assessment of different approaches and solutions*. One option was to calculate the exact value of p_n for as many n as possible and then extrapolate the growth rate. But according to Norman, this approach was inefficient: At the time of his project, they could calculate the exact values only for $n \leq 25$, which was not enough to provide an acceptable solution. The mathematicians hence chose a probabilistic approach that uses estimates for p_n rather than exact values. This enabled them to calculate more data points; but their program was still slow. Seeking to *troubleshoot and debug* the problem, Norman suggested that his team try to visualize the permutations. Their decision to represent a permutation $\sigma = \sigma_1\sigma_2\ldots\sigma_n$ as a function that sends i to σ_i led to the discovery of an unexpectedly striking structure (Figure 3, left). Norman insisted on the importance of *creating this particular computational abstraction*: The pattern would not have been observable, for example, had they produced only a list of matrix entries. And then Norman might have missed out on a novel research direction that occupied him for many years.

Since all MICA projects involve *programming*, computational problem solving practices like Norman's always form a major part of their completion. Starting in the first MICA course, students discuss what makes a math problem amenable to exploration through programming. Since this is new to most of them, they are also led to develop their computational skills through a carefully selected progression of projects, which increase in complexity in terms of both the mathematical content and the programming requirements. For example, Ramona and her peers learned about discrete dynamical systems alongside techniques of displaying graphics in VB.Net, which they applied by creating a program to numerically and graphically explore the logistic map. In a later project, the students were asked to build on this work (i.e., borrow *modular computational solutions* from it) and *program* a tool to explore the system of a two-parameter cubic (Figure 2, right). This new problem required more serious *preparation for a computational solution*, as the domain of the cubic called for the consideration of different cases. Inherent to the programming process was also *troubleshooting and debugging*, *creating computational abstractions*, and *assessing different kinds of solutions*, which may have contributed to Ramona's conclusion in her written report that "creating and working with this program has assisted me to fully grasp the way a dynamical system works by observing the table, the graphs, and the cobweb with countless test values."

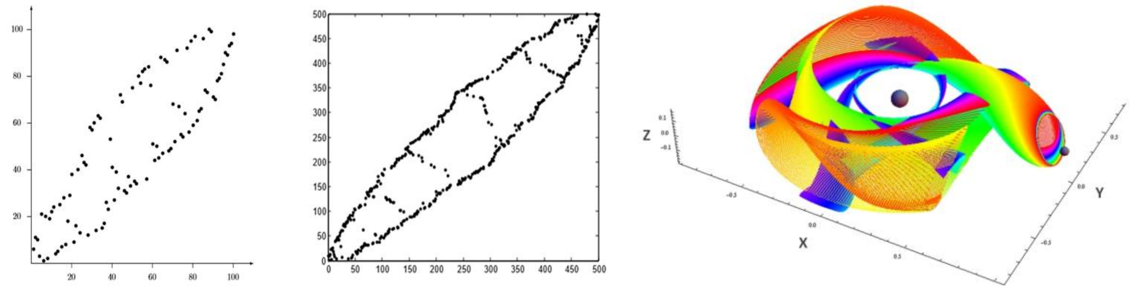

Figure 3: Norman's discovery of structure in pattern-avoiding permutations (left) and Albert's computation of trajectories resulting from a perturbation off an orbit close to the moon (right)

Systems thinking practices

Albert has studied many complex systems, including those *defined* in celestial mechanics. The three-body problem, for example, seeks to describe the motion of a spaceship in the presence of two bodies, like the Earth and its moon. The complexity of the system is *managed* by ignoring the presence of other bodies, taking the spaceship to have negligible mass, and assuming that the massive bodies move in circular orbits. These explicit boundaries do not render the system useless. In fact, the model has provided initial approximations for real space missions. Moreover, it serves as a rich source of problems that allow Albert to show off the mathematically and numerically sophisticated software he has developed, software that according to him can compute "amazing things" that are simply "not computable" by traditional methods. Albert's team has computed the uncomputable at different *levels.* Macroscopically-speaking, they have *investigated the three-body system as a whole* by finding and classifying an infinity of its periodic solutions (i.e., closed trajectories where a spaceship could remain in orbit). On a microscopic level, they have explored these orbits in family groups and individually. This latter consideration also helps them *understand some relationships between elements within the system*: For a given orbit, the researchers can determine the set of trajectories that a spaceship could follow after experiencing a slight perturbation. The resulting tube-like structures are like highways that enable space travel to far-away places with minimal effort (and money). One image (Figure 3, right) is enough to convey the importance of visualization in *communicating* Albert's results.

In each MICA course, the last two weeks are dedicated to challenging original projects wherein students select topics of interest to them. Ramona's terminal (14^{th}) project is an example of how students might engage in practices similar to Albert's and, as the MICA course creators aimed, "develop their own strategies for handling complex real world problems" (Buteau et al., 2016, p. 144). With two of her colleagues, Ramona *investigated, as a whole, the complex system* associated to the water level changes in Lake Erie (Canada). In particular, they were interested in explaining how and why the level changes over time (i.e., in *understanding the relationships within the system*). They described their initial research in existing literature as "a crucial starting point in [their] project, allowing [them] to obtain an understanding for the changes in the water supply of Lake Erie." Based on the information gathered, they designed and programmed stochastic and deterministic models of the phenomenon. They then performed an analysis, through simulation, of six case studies, representing the system in various ways on *a different, more microscopic, level*. They used their initial research to justify the assumptions they made, the parameters they chose, and the case studies they considered in order to *manage the complexity* of the system. This explanation was part of the 26-page report where Ramona's group *communicated their results*.

Discussion and conclusions

The four pairs of examples provided above aim to render Weintrop et al.'s (2016) framework more concrete, validate its correspondence with a diverse set of authentic professional practices, and provide some insight as to how students might be invited to gain access to them, all within the context of mathematics. Ramona's work differed from the mathematicians' in its magnitude: Her projects were more restricted in scope and length, her computer programs were more naïve, and her findings had less immediate value for the community at large. This is not surprising since Ramona was in a peripheral phase of participating in the mathematics community, where she was simultaneously negotiating entrance into a community of students at a particular university, with its own norms limiting engagement in full-membership mathematical activity. Nonetheless, in exposing Ramona to the computational practices of mathematicians, programs like MICA support a nuanced discussion of what it means to integrate digital tools in students' learning of mathematics.

Many scholars have reported on the ways in which building and/or interacting with digital tools might assist students in meaningfully acquiring mathematical ideas or ways of thinking that are embedded in current curricula. The collection of papers presented in the working group on learning mathematics with technology at this year's CERME conference provides numerous examples. In fact, the main framework used in this paper was built on the premise that learning activities involving computational thinking practices can enrich students' understanding of mathematics and science (Weintrop et al., 2016). This said, the framework was equally inspired by the ever-increasing computational nature of STEM-related disciplines. As evidenced by our examples, and much work that precedes us, the power of the computer has had a major impact on the way that STEM professionals (can) do their work. And so, the *computational thinking* trend presents an opportunity (or perhaps a necessity) for mathematics educators at all levels to reconsider not just the "how" of mathematics teaching, but also the "what", i.e., the knowledge and skills to be taught. After all, students' participation in the computational thinking practices of mathematicians might not just prepare them for a computational future in general; it may also widen their perspectives of the nature of mathematics and who is capable of learning (and doing) it.

Both research and experience tell us that reflecting on the above issues, developing curricula to address them, and enacting that curricula in classrooms are quite different feats. Detailed and extensive frameworks like the one developed by Weintrop et al. (2016) can certainly help support researchers, curricula developers, and teachers. But there is still a need to examine more closely and completely the experiences of students who are peripheral participants in computational communities of practice: What skilled knowledge (i.e., practices) do they *actually* develop? Moreover, how do they identify with communities they are both entering and (eventually) influencing? Given our analysis in this paper, the MICA program provides a rich context within which to study such questions. The answers could lead to an enlightening discussion about challenges and opportunities in bringing about a nuanced technology-rich mathematics education.

References

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education: Implications for policy and practice*. European Union. Retrieved from: http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf

Brennan, K. & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking.* Paper presented at Annual Meeting of the American Educational Research Association, Vancouver, BC, Canada. Retrieved from: http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Broley, L. (2015). *La programmation informatique dans la recherche et la formation en mathématiques au niveau universitaire.* Unpublished master's thesis, Université de Montréal, Montréal, Canada. Available at: https://papyrus.bib.umontreal.ca/xmlui/handle/1866/12574

Buteau, C., Muller, E., Marshall, N., Sacristán, A.I., & Mgombelo, J. (2016). Undergraduate mathematics students appropriating programming as a tool for modelling, simulation, and visualization: A case study. *Digital Experience in Mathematics Education*, *2*(2), 142–156.

Cuny, J., Snyder, L., & Wing, J.M. (2010). *Demystifying computational thinking for non-computer scientists.* Unpublished manuscript. Cited in: https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

European Mathematical Society. (2011). *Position paper of the European Mathematical Society on the European Commission's contributions to European research.* Retrieved from: http://ec.europa.eu/research/horizon2020/pdf/contributions/post/european_organisations/european_mathematical_society.pdf

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

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. New York, NY: Cambridge University Press.

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

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*, 127–147.