Computing as a Mathematical Disciplinary Practice

Elise Lockwood	Matthew Thomas	Anna DeJarnette
Oregon State University	Ithaca College	University of Cincinnati

The role of computation continues to be prominent in the STEM fields, and the activity of computing has become an important mathematical disciplinary practice. Given the importance of computational fluency in science and mathematics, we are curious about the nature of such activity in mathematics. To study this, we interviewed six mathematicians about the role of computation in their work, and we identified several aspects of computation that sheds light on the nature of computing as a mathematical disciplinary practice. In this paper, we present examples and applications of computation for these mathematicians, highlight types of computation, provide specific examples of computation in their work, and emphasize how computation relates to mathematics in particular.

Keywords: Computation, Mathematical disciplinary practices, Mathematicians

Introduction and Motivation

What is the role of computation for doing mathematics? What does computation mean, given the broad range of settings in which mathematics is applied? How could one justify the teaching and learning of computation, given the national focus on reasoning, problem solving, and abstract thinking at all levels of mathematics? Our research is driven by questions of this type, especially in light of the technological advancements that continually blur the lines that define what counts as doing mathematics among those in the profession. The content and practices of different levels of mathematics have traditionally been aligned to varying degrees with the discipline of mathematics (Moschkovich, 2007; Rasmussen, Wawro, & Zandieh, 2015). Thus, with current efforts to incorporate the use of technology, and to see mathematics as a setting to explore relationships with computer science (Grover & Pea, 2013), a natural question is how such work is conducted and perceived by professionals within the field of mathematics. Such perspectives can help to inform answers to questions about the meaning and purpose of computation at post-secondary levels.

With this study, we explore the relationship between activities that have come to be described as *computation* and *computational thinking* and the practice of doing mathematics. We used the lens of *disciplinary practices* (Rasmussen et al., 2015) to consider ways in which mathematicians view computation as an element of their professional work. To do this, we interviewed mathematicians about computation in their research and teaching, specifically, according to the following research question: *How do mathematicians characterize and use the disciplinary practice of computing in their work?* The results of our analysis indicate that members of the mathematical community value computing as a distinct practice, and that it may be beneficial to foster computational fluency among students. These findings give disciplinary support to efforts at incorporating computing and computer science into mathematics, and they begin to suggest some of the ways in which that integration may naturally surface.

Background Literature

In computer science education research, there is a construct called *computational thinking* (CT) (Grover & Pea, 2013; Wing, 2006, 2008), which Wing initially described as "taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computer science" (2006, p. 33). Wing went on to characterize CT broadly and as encompassing many kinds of thinking and activity, such as "thinking recursively" (p. 33), "using abstraction and decomposition when attacking a large complex task or designing a large complex system" (p. 33), "using heuristic reasoning to discover a solution" (p. 34), and "making trade-offs between time and space and between processing power and storage capacity" (p. 34). Wing did not intend for computational thinking to be neatly defined, and indeed the broad characterization makes it difficult to pin down a precise definition. However, describing a notion of computational thinking provides a starting point for identifying common threads among computational activity. While we do not explicitly examine computational thinking in this paper, we acknowledge the role that this idea played in the design of our study.

In exploring the idea of defining computational thinking, Weintrop et al. (2016) developed a "taxonomy of practices focusing on the application of computational thinking to mathematics and science" (p. 128). For each of these practices, Weintrop et al. (2016) elaborated certain activities that the practice may entail. For example, they said that *Programming* "consists of understanding and modifying programs written by others, as well as composing new programs or scripts from scratch" (p. 139). For *Troubleshooting and Debugging*, they explained that there are "a number of strategies one can employ while troubleshooting a problem, including clearly identifying the issue, systematically testing the system to isolate the source of the error, and reproducing the problem so that potential solutions can be tested reliably" (p. 139).

In developing their taxonomy, Weintrop et al. (2016) started with activities that elicited computational activity and refined that framework through interviews with experts (including school teachers and scientists). Notably, though, while they interviewed scientists (biochemists, physicists, material engineers, astrophysicists, computer scientists, and biomedical engineers), they did not interview research mathematicians. Thus, even though our study bears similarities to this work – namely asking experts about the nature of computational activity – we highlight two key differences. First, rather than beginning with a set of computational activities, we begin with mathematicians' descriptions of their work, forming categories and types of computation based on their experiences and responses. Second, by interviewing research mathematicians, we focus specifically on the role of computation within discipline of mathematics and not on STEM more widely. This attention to research mathematicians is closely related and relevant to undergraduate mathematics education in ways that broader STEM and K-12 emphases are not.

Theoretical Perspective

As we will see in the results, it is not trivial to define computation, and there are many ways in which computation can be characterized and framed. However, to clarify, in this paper we provide the following broad characterization as our working definition of computation: Computing is the practice of using mathematical calculations, processes, or algorithms, often to generate products that can be interpreted, investigated, or implemented in other contexts and problems. Computing often involves the aid of technology but can also be performed by hand.

Rasmussen, Wawro, and Zandieh (2015) defined disciplinary practices as "the ways in which mathematicians go about their profession" (p. 264), which they viewed as related to Moschkovich's (2007) notion of "professional discourse practices" (p. 264). These are the practices in which mathematicians engage in their professional spheres. Examples of disciplinary practices include algorithmatizing, theoremizing, defining, and symbolizing (Rasmussen et al., 2015). In this study, we are conceiving of computing as a disciplinary practice, something that mathematicians now do. Indeed, both Rasmussen et al. and Moschkovich argued that such practices are culturally and historically situated, and "socially, culturally, and historically produced practices that have become normative" (p. 25). We feel that this is an apt way to characterize computing, because computing seems like a particularly important disciplinary practice in our increasingly computerized society. That is, in light of increasing computational requirements for mathematics majors and computational methods in mathematical research (e.g., Bagley & Rabin, 2016), we feel that computing is becoming a relevant practice that is increasingly becoming an integral part of "being a mathematician." We thus consider computing to be a disciplinary practice and use this lens in framing our study. While the term computation could refer the product of the activity of computing, we use the terms interchangeably (as the participants used the term interchangeably during the interviews).

Methods

To answer our research question, we interviewed six mathematicians in single 60-90 minute semi-structured interviews. The mathematicians were professors in mathematics departments, all holding PhDs in mathematics (see Table 1; all names are gender-preserving pseudonyms). It was a convenience sample (mathematicians to whom the authors had access and proximity), but we sought to maintain a balance of sub-disciplines of mathematics (especially pure versus applied).

Table 1. Information about the Interviewees

Mathematician	Area of specialty	Years in field	Programming Language(s)
Michael	Mathematical Biology	4 years	Mathematica, Matlab
Liliana	Applied Mathematics	30 years	Matlab, Tecplot
Paul	Numerical Analysis	12 years	Matlab, Comsol, Maple
Carter	Geometry	35 years	Mathematica
Peter	Algebraic Combinatorics	18 years	Maple
Andrea	Applied Mathematics	7 years	Matlab, Python

All interviews were audio-recorded. We began the interviews by asking the mathematicians to reflect upon various aspects of computation, including computation in their own work, the value of computing for themselves and for students, and how they might teach computing. For example, after asking some preliminary demographic questions, we asked the following: *Do you use computation in the work that you do? How so? How are you defining 'computation'?* and *What are some specific ways (or contexts) in which you use computation in your work? Could you provide an example or two?* This enabled us to get a sense of how they viewed and might use computation. We also asked whether (and why) they thought computation is important for students to learn. We concluded with discussions about whether and how they had taught computation before, and for them to weigh in on how students might learn computation.

We used a combined process of open and axial coding (Strauss & Corbin, 1998) to describe the concepts, perspectives, and processes that characterized the mathematicians' ideas about computation in mathematics. In the first phase, the first author studied the transcripts and coded them with descriptions of the core ideas or themes from the mathematicians' comments. Examples of codes that emerged through this round of analysis include "computation is related to proving," "computation is used for generating examples," and "computation requires the compartmentalization of steps."

In the second phase of analysis, the second and third authors applied the generated list of codes to the entire set of interviews. All three authors met regularly throughout this process, during which time we compared our coding of the transcripts to resolve any discrepancies, refined the meanings of the different codes, and began to articulate a set of themes according to which the codes could be organized. We returned to the interview transcripts during these meetings looking for evidence for and against the common perspectives we saw within each theme. The results of our analysis include a set of themes that can be used, broadly, to categorize our participants' comments about computation, as well as examples from the data to support the variety of viewpoints that surfaced within each of these themes.

Results

We describe three main themes that characterize the mathematicians' comments about computation. First, we offer insights about how mathematicians characterized computation in their work, including similarities to and differences from programming. Second, we discuss practical applications of computation, including particular examples of how computation arises for these mathematicians. Third, we present mathematicians' views about the relationship between computing and mathematical problem solving. Through these results we seek to paint a more complete picture of how mathematicians think about and use computation in their work. Because of space we cannot highlight every point or make every part of it clear, but we can emphasize the main findings and provide evidence from a number of the mathematicians.

Types of Computation

In order to get a sense of how mathematicians defined or exemplified computation, we asked all participants a variation of the question, "what is the computation involved in your work?" Their responses indicated that the definition of computation, even within the field of mathematics, is difficult to articulate and is context dependent. In particular, the mathematicians made a distinction between numerical computations, and what might be considered algebraic computation, as exemplified by Michael's comments below:

Michael: There's computation, for instance, like if you're proving some theorem and you need a technical lemma and you've got to work out this computation just to show that that lemma is true. So that's one way. The other way is sort of like numerical computation. Computations that you're not going to do by hand, so you get a computer to do it.

Michael gave as an example of the first type of computation the case of showing that a particular function is Lipschitz – which involves verifying a string of inequalities – in order to use that property toward proving a more involved theorem. Numerical computation itself, according to Michael, could be further broken down into two different types: a tedious calculation that might best be done by a computer (e.g., a binomial probability with a large number of events) or computations akin to mathematical modeling, for which a set of data needs to be analyzed with no predetermined algorithm or formula.

The types of computation that mathematicians saw as most relevant to their work corresponded to the specifics of different sub-disciplines of mathematics. Peter, an algebraic combinatorialist, described his use of computation primarily in terms of algebraic computation (e.g., factoring complex expressions) and numerical calculations (e.g., calculating the determinant of a matrix). Mathematicians in more applied fields described computation in terms of solving models (e.g., solving partial differential equations numerically) or using computation to analyze data or approximate solutions. It was clear from our interviews that there was no consensus on a single definition of computation, although computation can be characterized broadly by a few different types of activities.

To summarize, in response to the question of how mathematicians use computation in their work, we saw that what constitutes computation varies according to the types of problems that are relevant within different subfields of mathematics. Computation as a practice occurs at different scales, from performing symbolic manipulations and numerical calculations, to creating and implementing mathematical models.

Examples and Practical Applications of Computation

The mathematicians articulated a number of ways in which they use computation in their work, and this provides insight into how and why computation can be so useful. Table 2 shows instances of what we coded as a practical application of computing, each with a supporting and exemplifying quote from a mathematician. This gives a set of concrete examples and evidence of the variety of ways in which mathematicians use computation in their everyday work.

Table 2. Examples and quotes of practical applications of computation

Practical Application	Specific Example(s) and Supporting Quote(s)
Testing conjectures	*Peter*: I use software to enumerate combinatorial objects that satisfy certain constraints where I have, say a conjecture – a prediction of how many there should be or a predicted bound on how many there should be and I'll collect numerical data to test my results.
Visualizing	*Liliana*: And sometimes I use computers to illustrate, um, some salient features of a problem that are otherwise hard to just understand. You can formulate them using proper algebra, calculus, whatever. But, you know, how common is it for someone to understand a complicated feature of a problem, um, using just, um, a formal, a very formal statement that involves, I don't know, derivatives or something like this? We can do that, but people really—if you have a finite amount of time to describe a problem, um, visualization is an important component. Uh, I think of what it means to convey some ideas. It does take a lot of time, and it is frequently unappreciated as part of that, um, research. But I think it's a very important part.
Communicating	*Paul*: Yes, definitely. I mean, you wouldn't be able to do proofs if you couldn't do that and you wouldn't be able to do computation if you couldn't do that and you couldn't make clear arguments to convince people of other types of conclusions that you're trying to make if you didn't take your arguments through logical steps. So, if you're trying to convince anybody of something and you need to tell them that your solution or your idea does what you think it does and nothing else. And that's exactly what a code's supposed to do too. *Interviewer*: Okay. Nice. And so, getting experience with that kind of coding could basically model that kind of experience of being able to give an argument and show that logical process of it. *Paul*: Right. I'm saying that it not only helps with doing math, it also helps with communicating math.

Recovering from mistakes	*Liliana*: And so that's the other ability. That, and, well, of course, and there is the other ability of being able to recover from mistakes. Which, in computing, is fundamental. And not being too frustrated and just keep going back and forth. And trying to morph something that you know worked to something you know should work.
Using Computation in Teaching	*Carter*: I use the computer to check exam solutions when I teach calculus. I use the computer to draw graphics that I use both in my own textbooks and in my own teaching.

Table 2 gives a sense of the variety of ways that mathematicians use computing in their work. Some of these applications (such as *using computation in teaching* or for *visualization* of ideas in research) help the mathematicians accomplish specific, practical goals. Other applications (such as *communicating* and *recovering from mistakes*) facilitate the development of other essential, broader practices and habits of mind. These examples inform why computation is a valuable and useful practice for mathematicians and suggest why it should be developed among students.

Relationship between Computation and Mathematics

The mathematicians also talked about the relationship between computation and additional mathematical practices. For example, some articulated that computation was related to proof, problem solving, and other aspects of mathematical thinking and research. In this section, we highlight some excerpts that raise some salient points of discussion about the role of computing in the field of mathematics specifically.

In the following exchange, Liliana describes the back and forth relationship between the mathematical and theoretical analysis she does and the computing in which she engages. We see that Liliana describes how she applies computational activity to mathematical analysis of a problem, which suggests that she is relating her computational activity back to the mathematical processes on which she is working.

Liliana: So, the other part, the theoretical part, is different. It can be supported by experimentation, so let's say you're not quite sure how the solution to this equation is going to behave. I don't know, nonlinear equation depending on the parameter, so you're not quite sure. You can do some analysis, and that can be tedious, but you can also explore it computationally. Which will suggest what tools you would use to analyze that. Or which would sort of verify some of the intuitions you had which you can later use here. Um, but generally—or you're deriving some kind of a theoretical result, which actually typically now we're, uh, research—you should verify, you should show some experiments that verify that indeed the convergence rates are this or that. If you can't get it to confirm, that's bad.

Interviewer: Mmhmm, because it suggests something is wrong your analysis.

Liliana: Something is either wrong with the implementation, or with the analysis.

Interviewer: Yeah.

Liliana: If you get a better result, then it's okay. I mean, if experimentally you're getting a higher rate than the one you did, then it means you didn't get sharp results. But if it was the other way, then you should go back to the drawing board because there was an unrealistic assumption you made or something like this.

Peter also emphasizes this relationship between the computation and the theoretical mathematical research he does. It is worth noting that Peter viewed computation as a way to generate examples and explore conjectures (as noted in the first row of Table 2 above). The following quote highlights that he views computational activity as providing necessary content that he can then use and apply in his mathematical research.

Peter: The act of doing the computations and the results that are obtained by computations are the content of mathematics. And so, the theorems and relationships we're describing are only good to the extent that they reflect something that either is calculated or that you're evaluating by not calculating directly or – you know, yeah. It's like if a poet has no life experience they can't write good poems. If a mathematician doesn't do computation or have the results of computations, they don't really have the subject that they're supposed to be reasoning about in writing their theorems?

Andrea also spoke to the important relationship between computational activity and theoretical mathematical work. In discussing teaching computing to students and what she wants them to learn through the practice of computing, she indicated that she would want them to be able to relate the programming they do to the mathematical ideas.

Andrea: Like, knowing a program, or knowing how to, say, code in Python is a skill. I think it's not really useful unless you can – like, by itself I don't think is useful. So, I would hope that the student also knows how to do mathematics on paper.

Interviewer: So, what else do they need besides just that ability to program in Python?

Andrew: I guess it's what I would say any mathematician or math student would need to have, and it's the ability to – I'm not going to say this right because this is not my area, but – to be mathematically skilled. But I don't really know what I would say that is. It's like, the ability to solve any type of problem. Maybe the ability to approach a math problem with some insight into which direction to go in.

In this section, we see that the mathematicians understand that computation in their work must be connected to the theoretical mathematics, and we gain insight into how this practice of computing can complement and enrich the mathematical work that they do.

Conclusions and Avenues for Future Research

In this paper, we have reported on interviews in which we asked mathematicians about the role of computation in their work and in the field. We discussed their characterizations of computing, examples and practical applications of computing, and the relationship between computing and the theoretical mathematics they do in their research. Together these findings paint a picture of the varied ways that computation arises in mathematicians' work, and they highlight the important role that computation plays. In this way, we feel that our findings make a case for computing as a key mathematical disciplinary practice, helping us to justify the importance of developing this practice in undergraduate mathematics classrooms. Indeed, the fact that mathematicians use computation in a number of ways underscores that it is a practice that deserves more attention in mathematics education research.

In light of these findings, we have several ideas for further research. First, we feel that we should broaden our set of mathematicians, perhaps interviewing or surveying greater numbers of mathematicians and mathematicians in industry. Second, but we are interested in exploring the notion of computational thinking in mathematics. We contend that there may be certain ways of thinking that facilitate computation in mathematical contexts, and we want to investigate what such a way of thinking might entail. Third, we would like to investigate undergraduate students' characterizations and uses of computation in mathematics.

References

Bagley, S., & Rabin, J. M. (2015). Students' Use of Computational Thinking in Linear Algebra. International Journal of Research in Undergraduate Mathematics Education, 83–104. https://doi.org/10.1007/s40753-015-0022-x

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

Moschkovich, J. (2007). Examining mathematical discourse practices. *For the Learning of Mathematics, 27*(1), 24-30.

Rasmussen, C., Wawro, M., & Zandieh, M. (2015). Examining individual and collective mathematical progress. *Educational Studies in Mathematics, 88,* 259-281.

Strauss, A. & Corbin, J. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (2nd ed.). Thousand Oaks, California: Sage Publications, Inc.

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 and Education Technology, 25*, 127-147. Doi: 10.1007/s10956-015-0581-5.

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

Wing. J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A, 366,* 3717-3725.