Connecting Computation: Mediating Mathematical Knowledge Through Computational Modules

Sarah D. Castle
Michigan State University

This study examined the nature of Calculus II student's engagement with MATLAB modules and the ways in which the computational modules mediated student's understanding of Taylor series. Through analysis of observations and interviews, using an instrumental genesis framework, a pattern of student's views on the relationship between mathematics and computation developed in relation to student ability to conjecture and engage in mediated epistemic interactions with the MATLAB modules. This study highlights how the conceptualization of the two areas as distinct poses barriers and challenges for students, thereby questioning how to develop computational environments that support mediated epistemic interactions.

Keywords: Computation, Coding, Instrumental Genesis, MATLAB, Epistemic Mediated Interaction

New technological advances and the rise of data, coding, and computing have had drastic changes on the workforce infrastructure and the skills needed within the 21st century. There is an international shortage of those trained in the fundamentals of computational methods to fill jobs across academia and industry alike (SIAM Education, 2011), necessitating an examination of current educational methods to develop well-educated citizens in a computing intensive world (Cobb, 2015; CSTA, 2017). In preparing these students computational skills are needed, especially within mathematics (Lockwood et al., 2019). Computation is not only advantageous for student's future careers, but also has pedagogical motivation to deepen student's learning of mathematics by reinforcing concepts like abstraction, a key skill for mathematical proof and underpinning of computational thinking (Wing, 2011). Computation can reinforce desirable habits of mind such as communication and persistence (Lockwood et al., 2019). Despite the potential for the co-construction of mathematical and computational knowledge current studies in literature have focused on a proof by existence for this combination (Lockwood et al., 2019; Valdés-Fernández et al., 2019). Students not only have to grapple with the language of mathematics and the mathematical register but also comprehend the new grammatical strategies within programming simultaneously, resulting in the need to in essence translate between two technical languages (DeJarnette, 2019). Further, the blending of computation into mathematics does not always occur smoothly and can leave students feeling frustrated and confused as there is a disconnect between the two (Krause et al., 2019).

Universities have responded to this call for computing in a variety of ways including the new majors (Silvia et al., 2019) and new computational labs and modules (Chabay & Sherwood, 2008; Psycharis, 2016). The development of modules is a highly attractive as these can be paired with pre-existing curricula, are feasible for universities of any size, and have a low cost. Despite the potential promise, what is less explored in literature are the ways that computational modules can mediate mathematics learning. In order to gain a more nuanced understanding, I will address the following research questions: (1) What is the nature of student's engagement with computational modules employed during a Calculus II course? (2) How do the computational modules mediate student's mathematical learning regarding series during a Calculus II course?

Setting

This study focused on a specific MATLAB module developed for a large, midwestern R1 university as a part of the mathematical laboratory series for Calculus II courses. The course had no prerequisites for any computational or coding experience. The course met four times a week, where three days were lecture and the remaining day was a recitation. During the recitation, students alternated between taking a quiz or engaging with a MATLAB module. This course was chosen due to the pre-developed computational modules that had gone through years of section-focused reform and had been scaled up to a course-wide requirement during Fall 2019.

Theoretical Framework

In order to conceptualize the relationship between computational modules and students, an instrumental genesis framework (Béguin & Rabardel; Guin & Trouche, 1998; Vernillion & Rabardel, 1995) is utilized to focus on student learning in instrument-mediated activity. Students are able to either directly interact with the mathematics, or engage in a mediated interaction, as shown in Figure 1, where the computational module (the artifact) facilitates mathematical understanding (Lonchamp, 2012; Pargman et al., 2018). Through repeated mediated interactions, the artifact begins to reorganize the ways students engage with the mathematics, specifically that the nature of the knowledge a student generates has changed. As this continues, the artifact changes into an instrument, which does not exist in itself, but rather a student is able to appropriate and integrate it into their conceptions of mathematics (Verillion & Rabardel, 1995). During this process, which is referred to as instrumentation, students begin to understand the constraints, affordances, and the procedures linked to the artifact, and which leads to differentiated usages and ultimately to the psychological construct of the instrument (Guin & Trouche, 1998). Operationalization of this process is accomplished through instrumentalization levels. Longchamp (2012) delineates three levels, where the first is when the artifact is used for a particular action and under limited circumstances. The second level is when the artifact is linked to a class of situations, and the final level is when the artifact is modified substantially and permanently, as to perform a new function. These three levels provide insight into where a student is during the instrumentation process and relating the student's actions within the mathematical context.

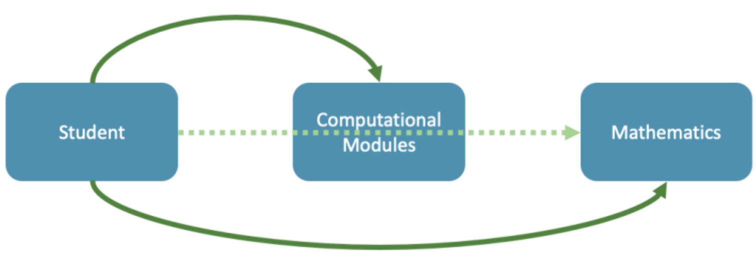

Figure 1. Students can engage with the mathematics and/or the computation directly, as indicated by the dark green solid arrows, or they can engage in a mediated interaction in mathematics through the computational module.

Methods

Participants and Data Context

Student participants were recruited across all sections, and a convenience sample of five students was selected. This sample spanned majors, computational experiences, and genders in order to bring a variety of perspectives and experience to the study. Due to Covid-19, all courses were moved completely online, including the MATLAB modules, so all data collection occurred through online video conferencing and screen recordings. It is important to note that since this change occurred mid-semester, students had been completing these modules with a group in person during a finite time period. After the switch to online, students were to complete the lab individually over the course of a few days, but still could get help through an online forum.

Taylor Series Module of Interest

The focal laboratory was the final MATLAB module focused on series. As students progressed through the course, the assumption was that there would be a greater proficiency with MATLAB. Therefore, since the familiarity with an artifact impacts the instrumentation process (Trouche, 1999), the final module was chosen in hope that there would be greater discussion about the mathematics, whether direct or mediated, rather than initial MATLAB environment questions. Additionally, during pilot interviews, students commented on enjoying seeing the applicability of series within the computational environment, rather than simply existing as an abstract mathematical concept. Students were provided with a MATLAB .mlx file which contained the bulk of code and instructions. They were responsible for modifying and potentially adding existing code in order to answer different mathematical questions.

Within the module, there were four tasks for the students to complete. The first task was to write a line of code that would calculate the maximum value of the difference between the fifth-degree approximation for ln(1+x) and the full Maclaurin series, then repeating this for the 200th degree approximation. Students were provided with a graph of the approximation and ln(2) plotted on the interval [-0.9, 0.9] with a slider to change the degree up to N=9. The second task focused on approximating ln(2). Students used a predefined function that calculated the first N terms of the approximation of ln(1+x), where x = 1 and then implement a trial-and-error strategy so that the difference between the Nth degree polynomial and ln(2) was no more than 10^{-8}. The third task had students rewrite the logarithm ln(2) = -ln(1/2) and then use the predefined function to calculate the Nth degree Maclaurin series polynomial at x = -1/2. The goal was to calculate the degree needed so that the error was no more than 10^{-8}. The code also generated plots to show the error plotted against the degree. Finally, students were to figure out why the second method needed fewer operations. Students were provided with the code to generate a plot of ln(1+x) plotted against the 10th order Maclaurin polynomial on the interval [-0.8,1.2].

Data Collection and Analysis

Each participant went through a POP cycle, which refers to a pre-observation interview, observation, and then a post-observation interview. The pre-interview focused on establishing a baseline for the participant, both with respect to their mathematical and computational backgrounds as well as to how they generally engage with the modules. Participants were also asked about Taylor series and their importance. This interview occurred 1-2 days prior to their observation. Observations lasted approximately one hour, in order to mimic the conditions prior to the transition to online learning and provide a time boundary for students. During this time, I was a passive observer, and participants could pursue any line of inquiry and use any resource to complete their module. Observational notes were taken in addition to the video and screen recordings to record any poignant points to clarify during the post-interview.

The post-observation was conducted within two days following the observation, allowing participants to reflect on their experience and ensure that the details were fresh in their minds. The same set of questions surrounding Taylor series were asked in addition to video-stimulated reflections. Short clips of interest from their observation were shown to clarify participant actions and motivation, allowing for illumination of student thinking during the observations.

After completing the POP cycles for all participants, general profiles of students were developed. Participant observations were coded so that each participant action was the unit of analysis, including, but not limited to, reading code, running code, modifying a variable, adding a new line of code, and scrolling. These actions were then coded as either direct or mediated as prescribed by the theoretical framework. Longchamp's (2012) dimensions of mediation were applied, in that during interactions, students would either engage in epistemic or pragmatic mediations. Epistemic mediation seeks to know the object (MATLAB module) and to transform it into an instrument in relation to the mathematical context, whereas a pragmatic mediation focuses on comprehending a specific task and context. Looking across the individual actions led to the grouping of actions by topic and/or participant focus; these were referred to as conversations. These conversations were then coded as either pragmatic or epistemic. A conversation could contain both epistemic and/or pragmatic actions despite the overall coding.

Participant Overview

Five students were a part of this study: Clint, Tony, Carol, Janet, and Margaret. These students had varied computational and mathematical experiences, as well as differing majors. An overview of the participants is presented in Table 1.

Table 1. A table summarizing the participant's majors, prior computational and mathematical experiences.

Participant	Major	Prior Computation	Prior Mathematics
Carol	Chemical Engineering	MATLAB & Python	Calculus I
Clint	Computer Science	Java, SQL, Python & C++	Calculus I
Janet	Chemistry	None	Calculus I
Margaret	Packaging	None	Calculus I & ITP
Tony	Mathematics	Calculus II Modules	Calculus II

Results

During the interviews and observations, what became increasingly apparent was that it was extremely difficult to mediate mathematics learning if participants viewed computation and mathematics as two separate entities. That is, when students tended to separate the mathematics into one domain and the computational interactions into another, it was difficult to engage in mediated epistemic interactions. Therefore, although the goal of the modules was to enhance student's understanding of Taylor series, when this divide existed, the computational activity was not necessarily developing student's understandings. It is important to note that this separation does not refer to the previous computational experiences, such as those that learned coding in the context of mathematics or physics versus those that took independent coding courses. Rather, the separation is referring to the in-the-moment actions and the ways in which students expressed their thought process during the interviews. Further, although this presentation is seemingly dichotomous, the reality is that the ways that each student viewed the relationship, or lack thereof, with mathematics and computation ebbed and flowed during the observations. Nonetheless, I contend that the separation of these two entities contributed to frustration and created barriers for some students to engage in mediated epistemic interactions.

In order to highlight the separation, I will initially focus on the work of Tony, who had previously taken Calculus II and had engaged in some prior MATLAB experience. For Tony, he began by exploring the pieces of code, and interacted with a built-in slider and visualization output, but he expressed some initial frustration in that the code was "really difficult for [him] to understand". He switched to using a pen and paper since he felt that he could not "get it with the MATLAB code and was trying to cheat [his] way through with the MATLAB code". He recorded his answer after some hand calculations and hoped that he did not "do all that math for nothing", referring to his pen and paper approach instead of the code, and continued onto the next section. Both of these actions were direct and pragmatic interactions with computation and mathematics respectively. During the remainder of the observation, and the reflection, Tony continually made remarks such as the "code would have been helpful if I could have figured out what it had been telling me" or that he "honestly [found] this to be extremely confusing" when reading code. When working on the second exercise, Tony focused on a graph of error versus the degree of the polynomial and then stated:

> "Okay so I assume they mean it is some number in here. *[Pointing to the higher degrees and clicking on different points which display the number of terms and the corresponding error of the approximation]* Oh! Oh sweet. It is going to be way higher than 250 then. Can I get a better like approximation if I just change some code or something"

He began by changing a value in the code from 250 to 5,000, and then stated "I don't really feel like guessing and checking like they are asking me to. It seems tedious… Hmm only three digits in *[referring to the error]* so let's try 50,000". He continued with this method but encountered roadblocks through some runtime errors and syntax issues, as there were different locations to potentially modify the existing code. After a brief pause, he reread the question, consulted some resources, and noted that

> "I thought I understood the question and I don't think I actually do. I am trying to see if that particular function, or the Taylor expansion follows the function ln(x) or ln(1+x) all the way to infinity and it doesn't. It follows it to 1. I was trying to find its radius of convergence."

He returned to the code and stated "I think that I might just do what I was doing before and try to find it when it is a huge number. I don't understand... It seems right somehow... but I don't really understand how." He continued to try different approaches and was met with new errors until the time of the observation had expired. During this part of the observation, Tony was primarily focused in on the coding portion and separated it from the mathematics. By viewing the coding as 'cheating' with MATLAB, this underscored an assumption that somehow the computational element was not genuine mathematics, or that there was some sort of deceit or fraud occurring. Rather than having computation as an environment for, or an extension of mathematics, this opposition formed a barrier for Tony. Even in his frustrations, he noted that it is the coding and after encountering errors, he then focused on the mathematics independent of the coding context. Tony continued to jump between the two different realms, rather than using one to mediate the understanding of the other and this transition brought about tensions.

This tension was not isolated to Tony, rather both Margaret and Janet had similar experiences with the separation. Margaret had encountered technical issues in the beginning and was unable to modify code, but she decided to read through the code and she stated that she was "just trying to read and understand the jargon" and followed up with " I am trying to figure out the coding so that I can make it work for me so I can figure out uh the sum and figure out how I can change x and therefore change the term". The observation ended early due to these difficulties, but before her follow up interview, she completed all the tasks by hand using pen and paper. Despite the difficulties, she emphasized that the goal was to make the coding work for her and in her follow up interview elaborated that she did not understand why the computational modules were required. She has in essence put coding and mathematics in distinct categories, mitigating the potential overlap and mediated epistemic interactions.

In her pre-observation interviews, Janet had noted that typically with the modules she walked away with “confusion and no real insight into calculus concepts” as many times she would “look for patterns in the code and change things”. During the observation, when comparing the two different approaches to approximations, she stated that “there must be something in the code which is why it probably makes sense” and then proceeded to work with the code in order to answer the question. During these encounters, Janet had separated into either looking at the code, or looking at the mathematics. When she used the different resources available to her, such as a group chat or online forum, she expressed that she was “just trying to figure out how do you know what you are doing” when modifying the code. Although there were periods for participants where the mathematics and computation were separate, there were also points which highlighted a connection between the two and using the computational environment to understand the mathematics.

For example, consider the case of Carol. During the third problem, she initially changed the code to try and center the Maclaurin series at .5; however, this resulted in a complex number, an unexpected result. She adjusted her array to have negative terms but commented “uhm that’s weird I have never summed something where N is negative… Well, we will go for it and see what happens”. This resulted in immediate MATLAB errors, and she commented “I feel like this is degrees because this is like an array that it is evaluating [approximation]”. After these MATLAB warnings, she discovered an error, reset her code using positive indices, and then fine-tuned her approximation. During this mediated interaction, Carol’s understanding of mathematics informed her approach to the code, in that it seemed irregular for her to have a negative N, and also her understanding of coding informed her mathematical understanding. She knew what an array did, and she began to link the terms of the approximation to this computational construct. The allowed overlap permitted mediated interaction. A key moment for Carol was during the final problem, where she reasoned:

> “A Maclaurin series is centered at 0 and radius of convergence is 1. So when x=1, you’re pretty much on the endpoint of where the function is converging *[long pause]* no could be converging since R=1, you have an open interval of convergence of (-1,1)... the question is does it converge on 1 because if [not] it won’t be very good at approximating”

After an initially reasoning, she carefully went through the lines of code and examined the outputs stating "okay I think I am right because it is telling you to look at this right here *[pointing at x=1]* and it really begins to really diverge here.. Out of curiosity though, what happens when you increase the number of terms… Oh it definitely diverges”. Carol was able to use the computational environment in order to develop a deeper understanding of the mathematics, namely what it means for a series to diverge. For her, the code was a realm in which she could explore the mathematics and conjecture, which led to a fundamental change in how she viewed diverging endpoints. She expressed that she had always wondered what a diverging endpoint meant and understood and could visualize it. This mediated epistemic interaction proceeded to develop a higher level of instrumentation as her understanding of diverging endpoints as a whole has been changed and she now expressed her understanding in relation to the MATLAB module.

Finally, Clint expressed statements where he would adjust the number of terms and related it back to the degree of a polynomial and error. When gesturing towards the error output, he stated the error is the difference between "the actual answer and the approximation and the lesser the difference, the better for us" and increasing the number of terms would "make the difference much much less… because it approximates it better". The ways in which he engaged in the computation related back to the mathematics he was investigating. He attributed this to his prior coding experience as he could "just figure out [the mathematics] at the start and get right to the point as opposed to figuring out the code itself". For him, he was able to use the computation as a way to investigate the mathematics. When adjusting the number of terms, Clint would conjecture a potential output and what the code would do, specifically surrounding the error and the quality of approximation.

Discussion and Conclusion

One of the primary results from this study was the challenge for students to engage in mediated epistemic interactions when viewing computation and mathematics as two distinct entities. This finding is reified in the literature as others have noted the perceived disconnect between the two (Krause et al., 2019) and the tendency to delineate struggles into either understanding the code or the mathematics (DeJarnette, 2019). Further, an interesting point arose in the mediated epistemic interactions, such as that of Carol and a diverging endpoint. Carol tended to view the computation and mathematics as connected, and then conjecture about the mathematics in this computational context. The act of conjecturing preceded the mediated epistemic interaction, and this occurred for Clint as well. This process of conjecturing, experimenting, and testing understanding of code and mathematics, are all in line with the practices detailed by Lockwood et al. (2019) that mathematicians use. Therefore, if there is an environment where computation and mathematics are viewed as connected, then potentially incorporating conjecture may lead to developing a deeper understanding of mathematics. Nonetheless, there is a key implicit assumption in that there is not a disconnect between the two. For example, Tony had engaged in conjecture and was beginning to engage with both the code, and the mathematics, but was missing the key link between the two that would have potentially allowed his conjecture to initiate an mediated epistemic interaction. If the perceived structure is that computation exists outside of the main lecture hall, as nearly all students commented that it was not mentioned during 'normal lectures, then this reinforces a view that mathematics and computation are separate, which in terms makes mediated epistemic interactions far more difficult. This delineation calls into question the notion of simply attaching labs to a mathematics class. If the goal is to understand the mathematics more deeply or be able to engage with it in a new context, then the dichotomy poses a serious threat. It seems as if an integrated computational component could better address the overlap and why computation is beneficial to mathematics, in turn providing an environment for mediated epistemic interactions to occur.

When designing computational experiences in the mathematics classroom, a key question that needs to be addressed is how to create an environment that foster's students understanding of both mathematics, computation, and their unique relationship. The danger of having a sole coding introduction at the beginning of the semester is that students are exposed to MATLAB in limited circumstances and linked to particular actions, thereby making it difficult for students to proceed to the second or third level of instrumentation. Further, if this initial introduction positions the coding and computation as distinct from the mathematics, then students may not be in an environment that promotes epistemic interactions with the mathematics. Further study on the relationship to the prior coding experiences and view on mathematics would be beneficial to further understand how students coordinate learning mathematics and computation simultaneously and use computing to explore the mathematics.

References

Béguin, P., & Rabardel, P. (2000). Designing for instrument-mediated activity. *Scandinavian Journal of Information Systems*, 12(1), 1.

Chabay, R., & Sherwood, B. (2008). Computational physics in the introductory calculus-based course. *American Journal of Physics*, 76(4), 307-313.

Cobb, G. (2015). Mere renovation is too little too late: We need to rethink our undergraduate curriculum from the ground up. *The American Statistician*, 69(4), 266-282.

DeJarnette, A. F. (2019). Students' challenges with symbols and diagrams when using a programming environment in mathematics. *Digital Experiences in Mathematics Education*, 5(1), 36-58.

CSTA (2017). K-12 Computer Science Standards, Revised 2017. *Computer Science Teachers Association*, USA.

Guin, D., & Trouche, L. (1998). The complex process of converting tools into mathematical instruments: The case of calculators. *International Journal of Computers for Mathematical Learning*, 3(3), 195-227.

Krause, A. J., Maccombs, R. J., & Wong, W. W. (2021). Experiencing Calculus Through Computational Labs: Our Department's Cultural Drift Toward Modernizing Mathematics Instruction. *PRIMUS, 31*(3-5), 434-448.

Lockwood, E., DeJarnette, A. F., & Thomas, M. (2019). Computing as a mathematical disciplinary practice. *The Journal of Mathematical Behavio*r, 54, 100688.

Lockwood, E. (2019). Using a Computational Context to Investigate Student Reasoning About Whether "Order Matters" in Counting Problems. In A. Weinberg, D. Moore-Russo, H. Soto, & M. Wawro, *Proceedings of the SIGMAA for Research in Undergraduate Mathematics Education* (pp. 385-392) Oklahoma City, OK

Lonchamp, J. (2012). An instrumental perspective on CSCL systems. *International Journal of Computer-Supported Collaborative Learning*, 7(2), 211-237.

Pargman, T., Nouri, J., & Milrad, M. (2018). Taking an instrumental genesis lens: New insights into collaborative mobile learning. *British Journal of Educational Technology*, 49(2), 219-234.

Psycharis, S. (2016). Inquiry based-computational experiment, acquisition of threshold concepts and argumentation in science and mathematics education. *Journal of Educational Technology & Society*, 19(3), 282.

SIAM Education, W. G. O. C. U., Co-Chairs, P. T. A. L. P., Shiflet, A., Vakalis, I., Jordan, K., & John, S. S. (2011). Undergraduate computational science and engineering education. *SIAM review*, 53(3), 561-574.

Silvia, D., O'Shea, B., & Danielak, B. (2019, June). A Learner-Centered Approach to Teaching Computational Modeling, Data Analysis, and Programming. In *International Conference on Computational Science* (pp. 374-388). Springer, Cham.

Valdés-Fernández, S., Lockwood, E., Fernández, J. (2019). Computational Thinking Mediating Connections Among Representations in Counting. In A. Weinberg, D. Moore-Russo, H. Soto, & M. Wawro, *Proceedings of the SIGMAA for Research in Undergraduate Mathematics Education* (pp. 1193-1194) Oklahoma City, OK

Verillon, P., & Rabardel, P. (1995). Cognition and artifacts: A contribution to the study of though in relation to instrumented activity. *European journal of psychology of education*, 10(1), 77-101.

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine*, 20-23.