CODING FOR CREATIVITY: EXPLORING THE IMPACT OF COMPUTING ENACTED THROUGH CODING ON STUDENTS' MATHEMATICAL CREATIVITY IN LINEAR ALGEBRA

By

Sarah Dorothy Castle

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mathematics Education – Doctor of Philosophy

2023

ABSTRACT

Most mathematicians state mathematics is a field of beauty, creativity, and innovation, yet what is echoed by students is not how mathematics allows students to explore, but rather the ways in which it constrains and does not necessitate creativity. In comparison, many students associate coding with freedom, and the ability to create and express themselves. Therefore, a natural question that arises is what if computation, enacted through coding, could offer mathematics education more than just an application or skill? What if computation was a pedagogical design tool for mathematical creativity? This dissertation used cultural-historical activity theory as a theoretical perspective alongside case study methodology to answer the guiding research question: *How can computation enacted through coding provide opportunities for students to develop and express their mathematical creativity specifically in the context of learning linear algebra?* This study followed students recruited from an introduction to computational modeling course as they engaged with a sequence of eight Jupyter computational notebooks that introduce linear algebra. Each computational notebook was designed using an understanding by design framework with multiple ways to elicit student understanding and engagement with mathematical creativity. Further, use-modify-create cycles were incorporated to scaffold student learning and foster opportunities for mathematical creativity. These notebooks were deployed during an iterative pilot study, after which they were modified for the full study. Students engaged in weekly two-hour observations to complete the modules with their small group, followed by weekly experiential reflections. Students also participated in pre/post surveys and interviews. The data analysis included iterative coding methods using operationalized dimensions of creativity: fluency, originality, flexibility, visualization, elaboration, and risk, as well as framework linking student experiences in mathematics and computation to their relationship with the discipline. Five central

claims resulted from this study. The first claim was that computation, enacted in a creative environment, enabled opportunities for experimentation within mathematics through prediction and reflection cycles. Second, computation, enacted through coding, aided in the facilitation of connecting multiple representations. Third, computation provided new opportunities for students to expand their views of the nature of mathematics. Fourth, computation provided a novel environment that challenged previous negative mathematical experiences and allowed for shifts in students' mathematical self-image. Finally, computation enacted through coding provided the opportunity for students to develop new mathematical habits and strategies. These claims highlighted the potential power that computation has to foster opportunities for mathematical creativity, specifically through computational prediction and reflection cycles, visualization capabilities, and computational practices that align with mathematical creativity. This study serves as a proof of existence case for computation, enacted through coding in a creative environment, enabling mathematical creativity. Additionally, this study provides an alternate pathway for the integration of computation and mathematics. Rather than bringing computation into a mathematics classroom, this study introduces mathematics in a computation setting. This shift provides new potential integration pathways and is especially important for the computational education and computer science education communities. It also counters a deficit view of students, and leverages their assets, namely computation, as way of learning linear algebra. This work serves to contribute to the growing work exploring the integration of computation within post-secondary mathematics while simultaneously advocating for researchers and educators to think beyond computing as a tool, but also as a potential pedagogical approach to transform the undergraduate mathematics experience.

For my husband, Chris, who has ceaselessly loved and supported me –
For my mother, my first role model, who showed me the beauty in mathematics –
For my father, who has been my biggest supporter and number one fan –
For every person who has believed in me and supported me –
This is dedicated to you.

To God be the Glory
Now and Forever

ACKNOWLEDGEMENTS

Words are simply not enough to express how deeply grateful I am for the remarkable community that has supported me throughout my life. I know that I would not be the same woman, researcher, or educator that I am today without each and every one of you. May this acknowledgement section serve as a testament to the incredible community that has surrounded me with love and continually uplifted me, not only over the past five years, but over the course of my life.

To my advisor, Shiv Karunakaran, thank you for taking a chance on me after you received a random email asking to chat about mathematics education. You and Monica were one of the main reasons why I chose MSU, and I am so glad that I did. I am deeply indebted to the ways that you continually guided me and helped me become the scholar that I am today. You eased fears that I had during some of my most anxious moments and also gave me tools to succeed as a researcher, educator, and an individual. Thank you for your continual guidance.

To my committee, thank you for your knowledge, guidance, and numerous ways that you supported me over the years!

To Jenny Green, thank you for humanizing mathematics and statistics and being a constant source of peace and reassurance. There were so many times I would come into your office overwhelmed and scattered, and you simply sat with me and listened. You bring a peace that allows for calm in some of the most stressful moments and you made me feel seen during this journey. I cannot thank you enough and you are the type of mentor and faculty member that I strive to be as I move on in academia. Your light is seen by all, and I am so thankful to have gotten to know you at MSU.

To Vince Melfi, thank you for your inquisitive and empowering leadership over the past five years. I am so thankful for the support and feedback you have provided. I know I have grown as a scholar, especially from my practicum to now! I appreciate the deep questions you ask while also reassuring me of my potential and empowering me in my future research.

To Devin Silvia, thank you for the ways you supported my journey into computing education research. You opened the CERL community, made me feel right at home, and helped counter my own imposter phenomenon. You have a welcoming presence and a way of uplifting others that is remarkable. I am amazed at the ways that you support students, myself included, and the ways that you have also humanized the research process. Also thank you for all the new computational tips and tricks!

To Danny Caballero, I cannot thank you enough for the educational and research wisdom you have imparted. You were always available when I was in the weeds of my theoretical framework, and I appreciate your simultaneous patience and inquisitive nature. Further, I appreciate the ways that you subvert unjust systems and call out inequities. It is incredible to see as a student and encourages me as I continue on in academia.

To my participants, thank you for all of your time and hard work over the course of this study. It was truly a delight getting to know each of you. I am so thankful for the joy you brought to the study and all of your tenacity and excitement.

I would also like to thank Lisa Kelle, Freda Cruel, and Kelly Fenn. So often I would have academic questions, finance questions, or just some general PRIME questions and all of these incredible women were always eager to help! Thank you for the ways that you constantly care for PRIME students and always look out for us.

To Jihye Hwang, these past five years would not have been the same without you. Thank you for all the times that you listened to my research questions and supported me in figuring out how to proceed. Thank you for the support that you showed during some of the hardest points of the program. Thank you for simply being yourself and I am so lucky to have been able to share an office with you! I am so excited for our future conference reunions, and I cannot express how grateful I am for you!

To Sofía Abreu, thank you for your effervescent personality. You had a way of making me feel seen and heard during a time where that is not always the case. You have a way of radiating joy while calling out systems of oppression and I am so thankful for the ways you have impacted me and I cannot wait to see what is in store for you!

To Katie Westby, I am so thankful for your continual support and encouragement. Thank you for the ways in which you reminded me I was not alone and helped me navigate different academic structures. You have continually reminded me of what I am capable of and always spurred me on, even when I was in a state of paralysis. I cannot thank you for all that you did, especially while dissertating and on the job hunt!

To Chuck and Hannah Fessler, thank you for continually uplifting Chris and I during this process. I know that this doctoral program would not have been the same without both of you. The many game nights, Friendsgivings, trips, and cross country ski adventures were some of the highlights of our time in Michigan. Chris and I are so thankful for your continued friendship and support!

To my cohort, Jon Gregg, Rileigh Luczak, Sunyoung Park, and Jihye Hwang. Thank you for your continual curiosity and deep conversations. You all continually challenged my thinking and helped me develop as a mathematics education scholar!

To the Shivites, I am truly so thankful for the vibrant community. I always felt welcomed and supported by all! A special thank you to Bob Elmore and Valentin Küchle for all the guidance and wisdom that you bestowed over the years!

To the T2P team, thank you for taking me on during my very first year and continuing to support me! I am so glad that I was able to have mentors such as Jack Smith and Mari Levin who continually asked deep questions and pushed my thinking as a scholar and an individual. Thank you to all for constantly seeking a holistic research agenda that also enabled the development of graduate students as scholars.

To my dearest CERL squirrels, Tom Finzell, Rachel Roca, Rachel Frisbie, Patti Hamerski, Emily Bolger, Cassie Lem, and Amanda Bowerman, as well as the elder squirrels, Brian O'Shea, Aman Yadav, Devin Silvia, and Danny Caballero, I cannot express what this community means to me. I am so thankful that I joined this group because it has been a constant source of joy and encouragement. I was able to authentically be myself and I always looked forward to our meetings! This research group challenged me to be a better scholar and to use my position to advocate for justice and liberation, something not often found as a unifying call within a group. This group helped me get through the isolation of the pandemic and gave me an academic home and community. I will forever be grateful for the role that each of you have had in my life.

To my SEISMIC community including Carson Byrd, Becky Matz, and Meagan Pearson, thank you for the ways that you helped me develop as a scholar and an educator pursuing equity within the STEM context. You all have encouraged and emboldened me to take academic risks and developed my writing and research skills. Thank you for all of the opportunities and support.

To my Grad IV community including Caitlin and Matthew Bates, Julie Ruark, Chris Herrera, Adam Gleichman, Michelle Bullock, and so many others, thank you for helping me navigate my faith journey while in academia. You all constantly reminded me of my identity and walked beside both Chris and I during some incredibly difficult years.

To my church family at People's Church including Jordan & Jennifer Holmes, and Elisha Smith, and so many others, thank you for welcoming Chris and I with open arms! You all were the demonstration of love and constantly grounded us. The joy, love, and ever-present community were so incredible and thank you for your continual prayer and support.

To the many teachers, instructors, and professors I have had along the years, thank you for your continual support and the ways that you opened doors. I know I would not be here without all of you. Thank you to Dr. Comi for the continued support since geometry! Thank you to Dr. Kamesh Sankaran, Dr. John Larkin, Dr. Marcus Ong, and Dr. Richard Stevens for constantly believing in me, always having open office doors, and making me always feel at home. Thank you to Anne Trefry, Dr. Pete Tucker, Dr. Ed Walker, and all the other incredible faculty members of the Whitworth Mathematics and Computer Science Department for instilling in me a love of computing and mathematics.

To the other half of my brain, Hannah Edstrom, thank you for being one of my biggest mathematical supports. I know my love of mathematics would not be what it is today without your constant enthusiasm – especially when it comes to factorials! I am so glad that we had to work together back in geometry because you are now one of my dearest friends and biggest advocates.

To my friends near and far who have supported me through care packages, letters, prayers, FaceTime calls, and so many other ways – thank you. I cannot even begin to express my love and thanks to Kevin & Sarah Brown, Jordanne Mathena, Shelby Palmer, Emily Larsen, Tori Roberts, Lauren Spencer, Charlotte Taylor, Linnea Zavala Russel, Cass Sell, Jack Shannon, Jon Rubertas, Kate & Mark List, Matthew Wells, and so many others. I am constantly supported and uplifted by those named and unnamed, and I would not be here today without my village.

To all of my family members, I am so thankful for the roles you have played in my life and academic journey. To my late mother-in-law, Stephanie Castle, thank you for your continual support and guidance as I navigated what it meant to focus on work-life prioritization. Your ceaseless love will always be carried on by all those who knew you, and I am forever grateful for the ways that you empowered me as I pursued my doctorate. To Craig Castle, thank you for the many car-ride talks, check-ins, and the ways in which you supported Chris and I, especially as we begin our next chapter! To Sydney Castle, thank you for the constant encouragement and for my very first doctoral pen! To Jim Kuhens (Pa), thank you for always supporting Chris and I through your time and resources. I have felt so loved and I am so thankful for all the summer golf games, the stays at your house, and the ways that made Wabash a little piece of relaxation during this process!

To my parents, Martha and Steve Gady, words cannot express how thankful I am for the ways in which you supported and guided me. You encouraged me to explore, to be curious, and to be creative – lessons that I will also treasure. Mom, you were my very first example of a mathematician, and thank you for the ways in which you supported my love of mathematics and computing. I know I would not be here today if it were not for that support. Dad, thank you for the innumerable lessons you taught me and the ways in which you constantly showed love and support through many letters, care packages, flowers, and little gifts. Your creativity is something that inspires me, and I will always remember my 16th birthday red Honda fit cake!

Finally, I would like to express my deepest gratitude, appreciation, and love to my husband, Chris Castle. You have been a constant source of support and demonstrated your unwavering and unconditional love during this entire process. You walked alongside me and supported me when I did not think I could keep going and continually reminded me of not only how capable I was, but also reminded me that my identity was so much more than academia. Chris, words cannot express how grateful I am for your love and continual friendship. I could not have done this without my best friend! As you have been by my side during some of my darkest times, it is only fitting that I celebrate this accomplishment with you. I love you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

Over the last decade, technology has evolved, coding has become more ubiquitous, and computation is now a legitimate third pillar to theory and experimentation within the sciences. With the resulting evolution of computational resources, coding and computation are being brought into classrooms across disciplines and across academic levels. Specifically, when looking at undergraduate STEM education, coding elements are being introduced alongside disciplinary content, integrated throughout a department's curricula, or even spurring the development of new computationally focused departments. However, there are questions surrounding the efficacy of approaches and especially around the notion of having students not only learn new content, such as a mathematics concept, but simultaneously being asked to learn and understand a new language – coding, and then asked to combine this together all at once. Previous work highlighted the complexity associated with student mathematical discourse when the students are co-constructing an understanding of coding and mathematics (DeJarnette, 2016). Further, mathematics has faced a creativity crisis, in that many students view mathematics as a discipline that is simply the application of procedures and rules, with no space to explore (Boaler, 2018; Silver, 1997). This research aims to address these constraints and gaps within the literature by identifying and evaluating ways in which computation enacted through coding can aid mathematical understanding and promote mathematical creativity, while still preparing students for careers in the upcoming computational world. This chapter will introduce the study by first situating the research context, background, and key terminology. This will be followed by the research aims and questions, the significance of this study, and the limitations.

SITUATING THE RESEARCH CONTEXT

Prior to continuing, there are some key terms that may initially seem apparent to the reader, but due to the convolution of everyday vernacular and plethora of vague definitions, I will provide definitions that will be used throughout the remainder of the paper. I will then briefly detail the relationship between mathematics and computation and provide some of the context for the conception of the study.

Terminology

Mathematical creativity has a plethora of definitions in the literature (Henriksen et al., 2015) but I define mathematical creativity as:

> *A* ***process*** *that can be fostered, is context-dependent, and manifests in fluency, flexibility, and originality with respect to the mathematics which results in the connection and abstraction of ideas that bring a new perspective that is of use to the community and is many times the result of prolonged work and reflection.*

Further, the terms coding, computation, and computational thinking have become modern-day buzzwords. Throughout the paper I make an explicit delineation and assert that I am focusing on computation enacted through coding. Breaking this down, coding is “using the concepts of program, sequence of instructions, variables, recursion, etc., to write solutions” (Liao & Bright, 2005, p. 253). An example is a student being able to construct a for loop within a given program. There is no explicit mention of the efficiency of this implementation, nor is there any connection to the context in which the for loop was constructed. Computation is “an information process in which the transitions from one element of the sequence to the next are controlled by a representation” (Denning, 2010, p. 10) where the essence is to develop a representation, or a model, and use an information-based method to solve a particular problem. When considering computing in the context of coding, this then focuses on using a programming language or visual programing environment to develop a representation or model to solve a specific problem. Further, this computational environment could foster but not guarantee computational thinking, which is the thought processes involved in framing problems and elegantly designing their solutions in order that solutions are represented in a form that can be effectively carried out by an information-processing agent (Tedre & Denning, 2016; Wing, 2006). It is important to emphasize computational thinking is not simply focusing on obtaining a solution, rather it is engaging within the process, designing the method, and making a choice about how to wisely go about implementing this method so an information-processing agent can carry out the calculations. Further, there is no delineation or recommendation of programming environment as visual programming can be just as beneficial with mathematical learning and computational development (Romero et al., 2017), especially at younger levels. However, for the purposes of this study, I am specifically focusing on the computation enacted through coding.

The Rise of Mathematics and Computation

To contextualize the research, I will give a brief overview of some of the different approaches to integrating mathematics and computation. This should help situate the reader with a foundational understanding of the research area and provide for a richer and more nuanced read. The rise of computation and mathematics is not novel and was spurred on in the late 1900s by the push of Seymour Papert, who was a mathematician, computer scientist, and educator from MIT. He co-developed Logo, an educational programming language designed for children, and he pushed for the idea of students learning mathematics as naturally as they learn their native language without formal instruction. This push helped researchers and educators to ask questions about the integration of information technology and education and provided one of the initial examples of programming for educational learning, not merely as a skill acquisition approach. Individual programs and researchers have been bringing in elements of coding and computation into mathematics classrooms and curricula since Papert (Ahmed et al., 2020; Attallah et al., 2019; Buteau et al., 2016, 2020; Buteau & Muller, 2017; diSessa, 2018; Hart et al., 2008; Israel & Lash, 2020; Krause et al., 2020; Lockwood et al., 2019; Lovric, 2018; Sangwin & O'Toole, 2017; Sysło & Kwiatkowska, 2014; Tall & Thomas, 1991; Weintrop et al., 2016). Nonetheless, the research in undergraduate mathematics education (RUME) community had not fervently embraced this integration until recently, as grants and individual researchers have spurred this work forward. Lockwood & Mørken (2021) called on the RUME community to engage in research that explores the relationship between computing and mathematical thinking and activity. This work builds on previous work to situate computing as a legitimate mathematical practice rather than a potential application for mathematics (Lockwood et al., 2019). The 2021 call specifically invited a focus on research regarding student thinking and learning, research on teaching, and research on issues of equity while exploring the relationships between computing and mathematical thinking and activity. More detail will be provided with respect to the integration of computation and mathematics in the literature review. This brief overview situates the reader within the established history of the integration of computation and mathematics while also highlighting how this combination is a relatively new area of research, even if there have already been some integrations within mathematics classrooms.

RESEARCH PROBLEM

Mathematics is a field filled with beauty, creativity, and innovation. Yet what is echoed through the halls of schools is not how mathematics allows students to explore, but rather the ways in which it constrains and does not require creativity (Silver, 1997) - that mathematics is "a dead subject [with] hundreds of methods and procedures to memorize that [students] will never use, and hundreds of answers to questions that they have never asked" (Boaler, 2018, p. 31). Countering this notion, many organizations push for reform to reinvigorate the curriculum with creativity (i.e., NCTM, AACU) and employers yearn for more creative individuals (Riling, 2020). Simply put, there is a jarring disconnect between mathematicians who use creativity in their profession and a lack of creativity within schools (Boaler, 2018; Fetterly, 2020; Selbach-Allen et al., 2020; Silver, 1997).

Simultaneously, the desire for computing in mathematics education is repeatedly expressed (Buteau et al., 2015; Cobb, 2015; Feurzeig et al., 2011; Marshall & Buteau, 2014; Veaux et al., 2017). Most calls for computing focused on preparing students for future careers given the rise of coding or on the co-construction of mathematical and computational knowledge and the potential benefits. There is nothing inherently wrong with either of these foci, as STEM fields are developing a computational element on par with theory and experimentation, but what if computing could offer more to mathematics? What has been less explored in the literature is how computing, situated in the context of coding, has the potential to develop mathematical creativity. I argue that computation has the potential to be a natural environment to revive mathematical creativity within the classroom. Most studies on the integration of mathematics and coding focus on teaching coding skills to students within a mathematics course, but what has not been explored is how computation enacted through coding can be leveraged to teach and explore mathematics. This switches the paradigm and is a novel approach to the integration of a discipline and computing.

GUIDING RESEARCH QUESTION

This research aims to understand the ways in which computation enacted through coding can provide opportunities for mathematical creativity and rich mathematical sense making while engaging with the mathematics and computation simultaneously. For this study, I organized the opportunities for mathematical creativity and understanding as actions within Cultural-historical Activity Theory. This will be discussed within the literature review, and operationalized within the methodology section, but there are a few key components to highlight and describe to provide sufficient context for the ways in which I am viewing the study. This theory helps delineate the hierarchy between the actions the students take and the overall activity and the objective. Further, this theory places the learners within a larger system. When considering a student coding and learning mathematics, it is possible to use an approach that situates the student in relation to the objective of learning mathematics as having a mediated relation through the coding. However, activity theory brings about the community, rules, and the division of labor. This helps highlight that the ways students engage with mathematical creativity are not solely mediated by computational tools, but still allows for a focus on the computational tool and the experience as a whole by the student. With this contextualization, I now present my guiding dissertation research question:

How can computation enacted through coding provide opportunities for students to develop and express their mathematical creativity specifically in the context of learning linear algebra?

Although prior discussions of computation centered mathematics in a broader sense, for this specific study I chose to focus on linear algebra. Linear algebra provides the computational and theoretical foundation for a multitude of STEM disciplines, and besides Calculus I, introductory linear algebra is one of the most common mathematical prerequisite STEM courses

(Stewart et al., 2022). Therefore, when considering the areas of mathematics that would benefit a large population of students, linear algebra was a natural choice. Further, linear algebra underpins numerous computational applications including but not limited to data storage, machine learning, computer graphics, statistical methods, and computational modeling. Computation and linear algebra coexist within a disciplinary perspective, and therefore it would be advantageous for students to engage in computational linear algebra as an authentic practice. Finally, consider NumPy arrays within Python, a key data structure that is used heavily within data science. Two dimensional arrays are used heavily, and this data structure is simply representing a mathematical matrix. As students would be used to having the entries within the two-dimensional array encompass numbers, strings, and other data types then there is a natural extension to how matrices are not solely an array of numbers, as often seen within high school, but also that matrices can contain symbols or expressions and represent both mathematical objects and properties (Selinski et al., 2014; Stewart & Thomas, 2009). For all of these reasons, linear algebra was chosen as the focal mathematical area of concentration.

SIGNIFICANCE

The scope of this research is of use to the research community within mathematics education and computational education, as well as to practitioners. First, this study serves as a proof of existence case for computation enabling mathematical creativity. The pilot study already demonstrated some of the potential avenues for this occurring, but after refinement with regards to both methodology and pedagogy, this study is a richer case of the ways in which mathematical creativity can be supported. Secondly, this study provides an alternate pathway for the integration of computation and mathematics. Although many studies frame the work as bringing computation into a mathematics classroom, this study reverses that flow. Students were recruited from an introduction to computational modeling course, being proficient in basic coding syntax, and introduced to novel mathematical concepts. This shift in potential ways for mathematics and computing to be integrated is especially of interest to the computational education community and the computer science education community. This study aids in the understanding of bringing disciplinary context to computation, but also sheds light on potential methods for teaching mathematics. There are studies engaging in how computer science and mathematics overlap (Baldwin et al., 2013; Ralston, 2005), the relation of computational thinking and mathematical thinking (Gries et al., 2001; Sysło & Kwiatkowska, 2014), as well as computational and computer science students' 'misconceptions' and perceptions about mathematics (Attallah et al., 2019; Benadé & Liebenberg, 2019; Sigurdson & Petersen, 2017). This study reframes the narrative by leveraging a domain in which the students are more comfortable, as reported by this group of students, and uses this domain to introduce them to the mathematics, with which many students may have a complicated history with. This work thereby also counters some of the deficit narratives of mathematical understanding by taking a student-asset approach and leveraging students' strengths in computing for learning mathematics. Finally, this study paves the way for thinking about how to integrate computation and mathematics not solely from a content perspective but also from a pedagogical standpoint. The basis of this study is that the act of coding for computation has the potential to bring about actions that enable mathematical creativity and a deeper understanding of mathematics. This differs from some prior calls that computation should be integrated because of the need for coding in careers. Therefore, this study enables for richer implementation of the computation as a way of reforming the undergraduate mathematics experience. It is important that this will be a case of one method of enaction and is not the sole method, but the emphasis is still on leveraging computation for mathematical creativity and understanding.

DISSERTATION STRUCTURE

The remainder of the dissertation is structured as followed. In the following chapter, a detailed literature review will explore the ways in which mathematical creativity have been conceptualized, implemented, and assessed, as well as the prior work focused on the integration of computation and mathematics. This will be followed by an introduction to cultural-historical activity theory, which serves as the guiding theoretical framing for this work, as well as an introduction to the design frameworks leveraged to develop the computational materials. Chapter 3 details the methodology used within this study, in conjunction with the introduction of participants and the study materials. It is important to note that there was already a pilot study of the computational materials and the study's design. Therefore, there will be examples of the methodology grounded in the pilot data and linkages to the pilot study (Castle, 2023a, 2023b) while also using the pilot study data within the final results. Chapter 4 details the results from this study through five distinct claims. Finally, Chapter 5 centers on the discussion of the claims through the lens of cultural-historical activity theory as well as the implications of this work and potential limitations, barriers, and future steps.

CHAPTER 2 - LITERATURE REVIEW

The merging of computation and mathematics is not novel. For example, one of Papert's most known creations, LOGO, was a child-friendly software and coding language designed to engage students in mathematics and science. In a famous excerpt from his book Mindstorms, Papert (1993) states:

> In many schools today, the phrase "computer-aided instruction" means making the computer teach the child. One might say the computer is being used to program the child. In my vision, the child programs the computer and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building. (p. 5)

Within this quote, what is evident is that the programming was not solely a task, but rather a mediating tool that promoted students developing a unique conceptualization of the mathematical ideas. Within this dissertation, Papert's work motivated the conceptualization of computing being more than a syntactical tool for students and providing space for the fostering of mathematical creativity. Other researchers followed his push for mathematics and coding and as a result there have been multiple pushes for reform based on this overlap. However, most of Papert's work and his fellow researchers' work centered the child's experience and was designed for school-age children. This is one of the key delineations of my work as my work focuses undergraduate education. This is not to say that work from K-12 cannot be useful, but the contextual factors influencing my study are vastly different from both Papert and K-12, in addition to the content matter. In this context, I will review the related literature as this research aims to understand the ways in which computation enacted through coding at the undergraduate level has the potential to provide opportunities to foster mathematical creativity, influence students' understandings of mathematical content, and influence students' relationships with mathematics.

For this study I employed cultural-historical activity theory (CHAT) (Engeström, 2014) to map relationships and mediating elements when students engaged in learning mathematics through computation. When considering the interactions between students, mathematics, computation, and other elements, this activity system can become complex, and CHAT provides a way of organizing the relationships. This theory and its relation to activity theory (AT) will be discussed following the introduction (see Figure 2-6 for a visual model); however, a short overview of the critical parts of the framework will aid in the understanding of the literature review. Broadly speaking, we can consider an activity system where participants are motivated towards an object which leads to an objective. This interaction can be mediated by a tool or artifact. Further, there is a broader community that the subject interacts with, and the subject's interaction is governed by specific rules or social norms. The division of labor refers to how tasks are broken up among the community during the activity. There are three key relationships where it is pertinent to dive into the literature for the purposes of this study: the relation between the subject and object as mediated by the tool, the relationship between the subject and objective, and the relation of the subject that is mediated through the rules. Therefore, following an introduction to AT and CHAT, a review of the literature in the three foundational relationships for this study will be done. This includes the ways in which computation and coding have mediated students' interactions with mathematics, the ways in which mathematical creativity has been enacted for student learning, and finally the ways in which students have conceptualized mathematics, and their attitudes towards the field.

ENACTING MATHEMATICAL CREATIVITY

To discuss the relationship of students with mathematical creativity, especially in computational contexts, it is imperative to be explicit about how I am conceptualizing creativity. Recalling from the introduction, the definition that I am using for mathematical creativity is as follows:

> *Mathematical creativity is a process that can be fostered, is context-dependent, and manifests in fluency, flexibility, and originality with respect to the mathematics which results in the connection and abstraction of ideas that bring a new perspective that is of use to the community and is many times the result of prolonged work and reflection.*

This definition results from the combination and synthesis of research centering mathematical creativity, as there is not a universal, established definition – which many authors point to (Haylock, 1997; Savic, 2016; Savic et al., 2017; Silver, 1997; Sriraman et al., 2013). In this definition, *fluency* refers to being able to use a mathematical approach in a variety of situations and understanding the limitations and constraints. *Flexibility* refers to the ability to apply multiple mathematical approaches to a given situation. *Originality* can be thought of as novelty of ideas and approaches. The community mentioned with regards to this originality is not one isolated to solely the mathematical community at large, but rather is dependent upon the situation that the student is in. For example, consider when students work in small groups, the community would be situated to that subset of students. When they go to share out their work, then we could consider the entire classroom as the community. If a student is working on their own project, then the community could be considered that student and perhaps those with whom they share their work.

This conceptualization of mathematical creativity and originality allows for novelty to be found at all stages of students' engagement with mathematics.

A key point within the definition is the conceptualization of mathematical creativity as a process. This necessitates some historical context to highlight the importance of this conceptualization. The conversation about mathematical creativity originated in 1910 when Poincaré published his work entitled Mathematical Creation. Within this article, Poincaré raised questions about how individuals can(not) invent by drawing on his own personal mathematical experience to emphasize that discernment is a key facet as "to create consists precisely in not making useless combinations and in making those which are useful" (1910, p. 325) among other claims. He claimed from his experience that personal discernment is key to creativity. This element of discernment continued within scholarly writing; however, this also gave rise to a 'genius view' of mathematical creativity (Helson, 1983). The issue was not that discernment was bad but rather creativity was now an attribute of students, reserved solely for the geniuses or gifted students (Aiken, 1973; Kattou et al., 2013). This view narrowed who could be a mathematician and presented mathematical creativity as a flash of insight, rather than honoring the long periods of work and reflection. Not only did this limit whose thinking was privileged in the classroom, but it specifically limits the potential careers of non-white, non-male students, or those that have not been traditionally upheld within the mathematics tradition. If only a small subset of creativity is being recruited, then mathematics is depending on innovation from a small sample of individuals and therefore lacks the diversity of approaches and thought (Riling, 2020). This genius view still persists today, but the notion of mathematical creativity as a process, as set in motion by Hadamard, has risen in popularity and focus (Karakok et al., 2018; Savic, 2016; Savic et al., 2017; Sriraman, 2009; Yaftian, 2015). By viewing creativity as a process, this broadens the opportunity for participation within mathematics and honors the experience of students, rather than hiding behind a flash of insight. Within my study, I specifically conceptualize and implement this idea of creativity as a process. If creativity was only for a set group of students, then there would be no point in studying the ways that a computational context could generally give rise to opportunities for mathematical creativity. This specific view is nested within the guiding research question, and it is important to understand the historical origins in order to dispel some of the commonly associated connections. While this view of creativity does provide a start for the operationalization of creativity as a process, a crucial step is determining, developing, and using evidence for the manifestation of creativity.

Assessment of Mathematical Creativity

When detailing ways to promote the mathematical creativity process, a question that arises is how to determine if there is evidence of success. What are criteria that can be used to determine whether opportunities for mathematical creativity have been fostered? Current research has focused on the operationalization of rubrics to highlight the key manifestations of creativity (Blyman et al., 2020; Henriksen et al., 2015; Savic et al., 2017). The creation of these rubrics allowed for the operationalization of key features of creativity in conjunction with the development of examples of the corresponding manifestations of mathematical creativity. The categories from these rubrics were used within this study in order to provide an analytical framework coding system. The rubrics focus on the degree of mathematical creativity, many times with the goal of providing ways to grade for creativity or to look for creativity across a course. However, this is not the goal of this specific study. Rather, the goal is to understand how computation can promote mathematical creativity. Therefore, the question at hand centers on identifying moments of mathematical creativity and then looking at these different moments together to identify the role that computation played in fostering mathematical creativity. These rubrics provide dimensions of creativity with examples of how creativity may manifest in addition to operationalized definitions specifically focused on the mathematical creativity and provide a starting point.

This study used Blyman et al.'s (2020) categories as a starting point for operationalization, with definitions slightly adjusted based on the work of additional work and insights (Blyman et al., 2020; Savic et al., 2017; Silver, 1997), where the five dimensions of creativity are: originality, flexibility, visualization, elaboration, and risk. Originality is the ability to extend knowledge to new situations. Flexibility focuses on the use of multiple or interdisciplinary approaches whereas elaboration is establishing meaningful conceptual connections (Gadanidis et al., 2016; Leikin, 2019; Leung, 1997; Silver, 1997). Visualization is defined as developing and using illustrations to clarify concepts (Vale et al., 2018). Finally, risk refers to 'responsible' risks where an action is taken with an unknown result to advance the problem-solving process (Arney et al., 2020; Blyman et al., 2020; Craft et al., 2013). Each of these dimensions are consistent with the literature. However, one of the pieces that is missing from this conceptualization is fluency, or the ability to apply a mathematical approach to multiple situations (Kattou et al., 2013; Silver, 1997). This dimension is found across literature and therefore was added as another potential manifestation alongside the original five within this study. Further, these definitions leave ambiguity surrounding visualization. A student need not create a physical illustration, but a mental illustration is also a potential key insight into a manifestation of the mathematical creativity process. Therefore, for the purposes of this study, the definitions are adjusted accordingly and given some additional nuanced understandings. As an overview, these conceptualizations of the manifestations of mathematical creativity can be seen in Table 2-1. While these dimensions are to be used in data analysis, specifically for coding instances of mathematical creativity, they can also serve as a potential starting point for the design of activities to promote mathematical creativity. However, what needs to be considered is where within the overlap of computing and mathematics creativity might lie.

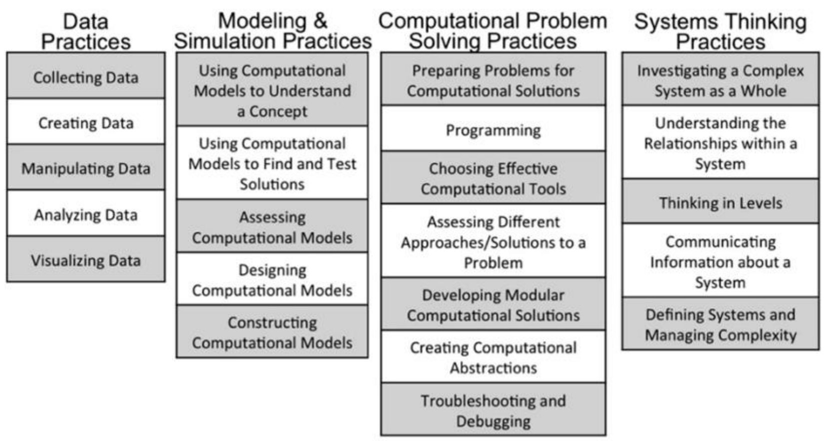
Table 2-1: The six dimensions of manifestation of mathematical creativity with their respective definitions.

Code	**Definition**
Fluency	The ability to apply the same mathematical idea, concept, or procedure, to a variety of problems and situations
Originality	The ability to try novel or unusual approaches towards a problem. This is contextually dependent, as a novel approach is caveated by what material the students had encountered previously and the current solution path.
Flexibility	Ability to use multiple methods (either from mathematics or drawing on an alternative discipline) for solving the same problem
Visualization	Development and use of illustrations (either physical or mental) to clarify or present concepts
Elaboration	The ability to establish meaningful connections between concepts typically by explaining a thought process in words. The validity of the solution is independent of the meaning made – but the student is engaging in sense making.
Risk	Taking responsible risk in the problem-solving process, willing to take an action where the result is unknown, or is a novel approach, in order to advance problem-solving process. Note that this can entail social risk

Enacting Creativity Through the Design of Computational Mathematics Experiences

When considering the potential affordances of computation for creativity, an acceptable approach would be to consider how computational thinking within mathematics manifests and what this allows with regards to creativity. Although computational thinking is not guaranteed, as students still can engage in the process of ‘pushing buttons without thinking’ (Tedre & Denning, 2016), one of the goals within computing is to bring about computational thinking. Computational thinking depends on the enactment of computation and is typically associated with computational environments, just like the tenants of mathematical thinking are hoped to be seen when engaging in mathematical exploration. Further, research supports the notion that students can develop both mathematical habits of mind simultaneously with computational thinking practices (Pei et al., 2018) and that mathematical and computational thinking have potential synergy (Rich et al., 2019). Within computational thinking, there is an element of choice, in that the student must decide which of all the different potential routes to take seems to be the 'best' implementation of the method for that context. This element of choice mirrors the focus of Poincaré (1910) in that mathematical creativity depends on choosing between a multitude of approaches rather than trying every potential combination of approaches.

Figure 2-1: Weintrop et al.'s (2016) taxonomy of computational thinking within mathematics and science.



The taxonomy proposed by Weintrop et al. (2016), as seen in Figure 2-1, delineates computational thinking in mathematical and science practices; there are four general broad categories: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices. Using this framework rather than viewing how computational thinking arises within mathematics, I used the identified categories to think about how to use situations that bring about computational thinking in mathematics for developing opportunities for mathematical creativity. Specifically, I focused on the modeling and simulation practices and computational problem-solving practices. This is not to say that other practices would not aid in the development of opportunities for mathematical creativity, but these are the areas in which computation had the richest theoretical potential.

These practices have a theoretical basis for why they could develop opportunities for mathematical creativity. A key practice within modeling and simulation is assessing computational models. During this process, students refine their justification for not only the choice of model, but also what assumptions affect the model's behavior. Through the understanding of the model parameters and validity threats, a student is able to potentially know when a model can be applied to a different problem in a novel context. This thereby has the potential to develop a student's fluency and broaden their understanding of further uses and additional assumptions for the model. Another practice focuses on the construction of computational models, as students who have mastered this practice should be able to extend existing models, which thereby develops students' flexibility once again as they create associations between the tools and applications. These are simply two examples of ways in which the activities presented within Weintrop et al. (2016) can be extended for mathematical creativity.

This framework offers guidance when designing curricular materials to promote mathematical creativity in computational experiences. There are multiple types of activities within each of the practices mentioned, and these give different computational exercises. Therefore, when designing computational tasks for mathematical creativity, this framework can serve as a source of tasks to leverage. This approach will be highlighted later within methodology and module design.

Avoiding Automation Within Enactment of Creativity

Within computation and coding in general, a concern is whether students are just 'pushing buttons' without engaging in computational thinking (Broley et al., 2018) and this is amplified for the ramifications of computing for mathematical creativity. If poorly enacted, then there is potential that the recommendation for computation for mathematical creativity could lead to its antithesis, namely simply following a set of commands, implementing a program, and going through an automized routine. However, this concern is not solely pertinent to computation, as within mathematics there are concerns that even if students demonstrate procedural and conceptual proficiency, this does not guarantee mathematical creativity (Tularam & Hulsman, 2015). The underlying premise is how to enact the computation, and the solution rests with the design. This is where computation paired with ill-defined and open problems (Savic et al., 2017; Selbach-Allen et al., 2020) has the potential to safeguard against 'button pushing'. By leveraging previously discussed activities and using Weintrop et al.'s (2016) taxonomy to develop mathematically creative experiences, this has the potential to help safeguard against button pushing, as will hopefully become explicit in the methodology.

Valuing Creativity and Encouraging Risk

One of the challenges that underlies the entire discussion on creativity and computing is developing a space in which students feel safe, encouraged, and rewarded to pursue creativity. Within academia, procedural can be valued over creativity, but at a detriment to student growth (Selbach-Allen et al., 2020). This is why creativity needs to be explicitly valued to counter messages from previous mathematical courses. If grades and assessment are focused on procedural fluency, then the implicit message to students is that ultimately creativity is not what matters, and it is more important to "play it safe" and follow a standard solution (Lew et al., 2016). To further the goal of creativity, it would be wise to not penalize in the same traditional manner and to have tasks in which the students are free to explore without judging 'correctness' (Savic et al., 2017). As risk taking is a key component of the manifestation of creativity, it is imperative to consider the ways in which instructors and the classrooms can encourage this risk taking and atypical thinking (Isomottonen et al., 2020). Therefore, within this study, the goal was towards exploration. This necessitated explicit messaging to students about the focus on the process over the product. Further, both creativity and programming are personal processes where a student is tied to their creation as well as achievement emotions are tied to the creative process, necessitating positive value appraisals for constructive problem-solving (Moore-Russo & Demler, 2018). Not only did students need to have a space in which creativity is explicitly valued, but there needs to be acknowledgment within the study that students' personal emotions are enmeshed within creation (Karwowski & Lebuda, 2017). The study provided opportunities for students to pursue their own lines of inquiry where they felt safe to do so. Therefore, consideration of the explicit messaging given to students needed to be considered.

THE LEARNING OF MATHEMATICS THROUGH COMPUTATION

This section will detail how computation has the potential to serve as a pedagogy for mathematics, the challenges posed, and the potential benefit of exploration within computation.

Computation as a Pedagogy for Mathematics

One of the recent shifts within the research community, especially at the undergraduate level, is the focus of using computation not only as an application of mathematics but specifically as a way to bring about unique mathematical understanding (Lockwood, 2022; Lockwood & Chenne, 2020, 2021; Odden et al., 2019; Sand et al., 2022). Further, the act of coding enables students to not only wrestle with mathematical ideas but reinforce key conceptual understandings. Specifically, Lockwood (2022) demonstrated the ways in which student combinatorial thinking is enhanced through the use of coding and computing. Students were able to link the counting processes, formulas/expressions, and sets of outcomes (Lockwood, 2013). The computational element of developing the sets of outcomes is enriched by students developing multiple conceptions of how to create the sets of outcomes. It is through this act that students were able to investigate larger sets, something that would be tedious and impractical to do with pen and paper. Much of this work highlighted the unique ways in which the actual process of computing expands the size of problem with which students are able to work. Further, that the act of engaging in the code through computation elicited a unique understanding due to students' use of coding syntax and the logic used to solve the problems (Lockwood & Chenne, 2020). This reinforces the notion that computation can be utilized as a pedagogical approach. The applications, and the ways in which problem size can scale are affordances of this environment, but a larger piece is the ways that the act of engaging in computation and computational thinking brings about key insights into mathematical ideas. The question remains of how this can be expanded for other mathematical domains.

The Challenge of the Language of Mathematics and the Syntax of Coding

A key consideration surrounding mathematics and computation revolves around language and level of detail. When integrating these disciplines for mathematics learning, students must coordinate between the two distinct but technical languages of mathematics and computation (DeJarnette, 2019). This requires students to combine the two domains which brings in new assumptions and different rules. Especially if students are newer to computation, or the particular coding environment, there is the potential for students to miss the bigger picture and get lost within the details of coding (Miller et al., 2013). Debugging within this context would not be solely syntax, but could be any combination of the coding syntax, computational approaches, or underlying mathematics.

While many studies introduce the coding through an introductory module or ask students to learn the syntax while engaging in new mathematics, what is less explored is the extent to which these domains should be cointegrated and in what ways (Ahmed et al., 2020; Israel & Lash, 2020). Concerns arise surrounding whether this integration could be detrimental to students. Specifically, if students are forced to use computational thinking skills outside of their knowledge, then they will become frustrated and this will in turn result in reduced learning for students (Engelman et al., 2017; Miller et al., 2013). There is an underlying deficit assumption in that students who are not in computer science tracks will have a limited understanding of computing concepts. This not only assigns them a lesser value, but there is no room for student growth in the argument. Further, as computation is becoming far more relevant in multiple fields, the assumption also does not stand the test. Therefore, the question centers on how to introduce both mathematics and computation.

Within this study, rather than trying to co-construct mathematics and coding knowledge, a potential path is to leverage students who already have computational experience. In doing this, the goal focused on leveraging where students were successful and already had experience. This allowed for the novel element to be the mathematical content and the way that it was incorporated into computing, rather than the syntax. This also opened up a key area of research as there is already some work done in relation to computer science and computational students learning mathematics (Benadé & Liebenberg, 2019; Hart et al., 2008; Pérez, 2018) but there is a dearth of assets-based approaches, especially within post-secondary educational research.

Computing Bringing Exploration into Mathematics Learning

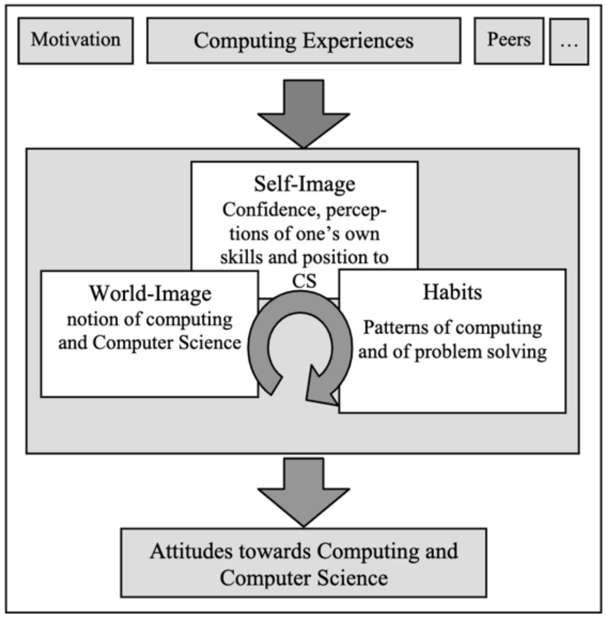
A key feature of a computational approach within mathematics that both has affordances and constraints is the ability to thoughtfully explore and experiment. When harnessed in a beneficial manner, this can lead to students exploring different mathematical concepts and engaging in conjecture and discovery. Singer & Voica (2015) noted that the steps students tended to engage in during problem posing was starting with a base model and then making a series of small changes. Even though the final model may diverge significantly from the original model, the way in which students make the changes is through the building up of small changes and testing components. This concept of allowing students to develop their own models, test them, and then come to a solution is an important portion of creative problem-solving (Gontijo, 2018) and computation is a natural place for this to occur, as learning math through programming and simulation leads to students being able to test and make their own conjectures (Kaufmann & Stenseth, 2020). This thereby has the potential to lead to novel understandings and approaches and developing specific insights into a number of key mathematical concepts (Feurzeig et al., 2011; García-Perales & Palomares-Ruiz, 2020).

Despite the potential promise in this area, the trial-and-error method may evolve from this original design which some argue can have a negative impact on students' mathematical arguments (Kaufmann & Stenseth, 2020). Furter concern surrounds the potential of experimentation devolving to a 'pushing buttons' strategy (Tedre & Denning, 2016). To potentially avoid adverse effects, one could have students to tinker with the computational environment first, prior to engaging in the specific mathematics within the environment. Tinkering allows a student to take advantage of the unexpected, draw on personal experiences, and then use the familiar materials in unexpected ways (Resnick, 2017). This approach not only allows students to become increasingly comfortable with the computational environment, but it also provides a beneficial environment for trial and error. The ability to tinker and test conjectures is beneficial for engaging in the initial exploration of a conjecture, or determining its validity, before engaging in proof, or exploring mathematical concepts (Barichello, 2016; Pei et al., 2018). Although a key point is that this is not of an ad hoc nature, rather this testing is a systematic and well thought out approach. Therefore, the exploratory nature can be beneficial, but as all educational pursuits, it does depend on the enactment and the goals for the lesson. This method of reasoning allows students to explore different avenues and efficiently engage in the conjecturing process which can thereby lead to original ideas.

STUDENTS' RELATIONSHIPS WITH MATHEMATICS AND COMPUTATION

The very act of bringing computation into students' mathematical learning experiences changes the overall activity system. Specifically, there is potential for the students' relationships with the mathematics community and the corresponding rules of engagement to change. To express the students' relationships with mathematics and understand how computational experiences might shift student attitudes towards mathematics and computation, I modified a framework presented in Schulte & Knobelsdorf (2007). Their original framework provides a model for how student's computing experience impacts their attitudes towards computing and computer science, as can be seen in Figure 2-2. The model states computational experiences are one of the factors that influence attitudes towards computing and computer science. More specifically this is because these experiences affect students' self-image (*confidence and perception of one's own skills and position to CS),* world-image *(notion of computing and computer science),* and habits *(patterns of computing and of problem-solving).*

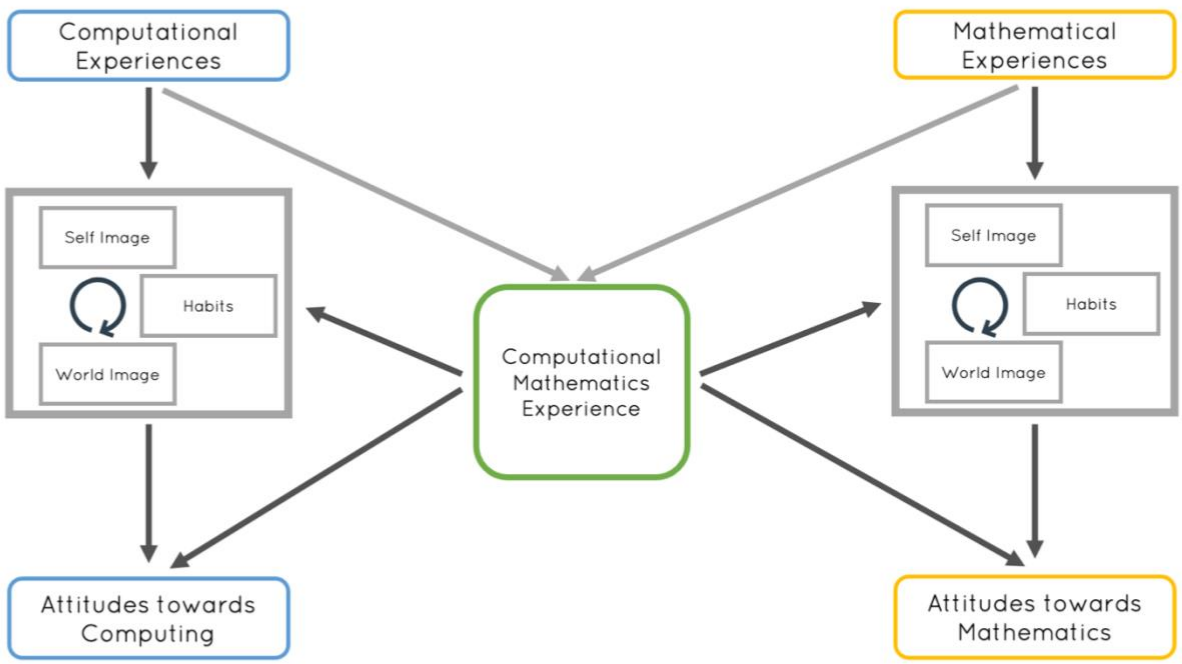
Figure 2-2: Schulte & Knobelsdorf (2007) model of biographical effects of computing experiences on attitudes towards computing and computer science (CS).



This model was chosen due to the theorization that the computing experience has an effect on the attitudes and self-images of students. One of the key underpinnings of why computation and coding have the potential benefit to mathematics is due to the computational practice of debugging. Debugging allows students to develop an attitude of resilience, and the notion that there will be errors along the way but that is part of the process (Pérez, 2018). This contrasts with many mathematics classrooms where students are rewarded for speed and accuracy in the problem-solving process. Therefore, if properly leveraged a computational experience with debugging could normalize mathematical errors for students. This change in student view of mathematics would be rooted in their experience.

The original model had the potential to be adapted to mathematics, as mathematical experiences influence students' attitudes towards mathematics and problem-solving through their self-image, world-image, and habits. Having two distinct models for students' relationships with mathematics and computation would be acceptable if students both experienced mathematics and computation separately and viewed the disciplines as separate without meaningful connections between the disciplines, as many do. However, the interplay between computation and mathematics is what students experienced during this study. Therefore, when we consider students engaging in mathematics through computation enacted through coding, then this assumption is no longer true as these models now have a linkage that needs to be studied.

Figure 2-3: Adapted framework linking computational mathematics experiences and students' respective attitudes. The computational and mathematics experiences are not the sole factors that affect students' attitudes, but the framework highlights the ways in which the experiences can affect their attitudes. The arrows do not indicate a hierarchy between students' self-image, habits, and world-image in the disciplines, but rather represent an ongoing interaction.



To capture the linkage between computation and mathematics, I proposed the following framework as seen in Figure 2-3. The original framework (Schulte & Knobelsdorf, 2007) is still visible as seen with the computational portion of the framework on the left. What has been extended is the introduction of a new type of experience - computational mathematical experiences, as well as the corresponding explicit relationship between the computational experiences and mathematical experiences. In this case, this new experience has the potential to influence students' self-image, world-image, and habits in both disciplines. As many students have already had mathematics courses or computing courses when they experience computational mathematics, the framework also reflects that students' attitudes toward mathematics and computing also influence the computational mathematics experience.

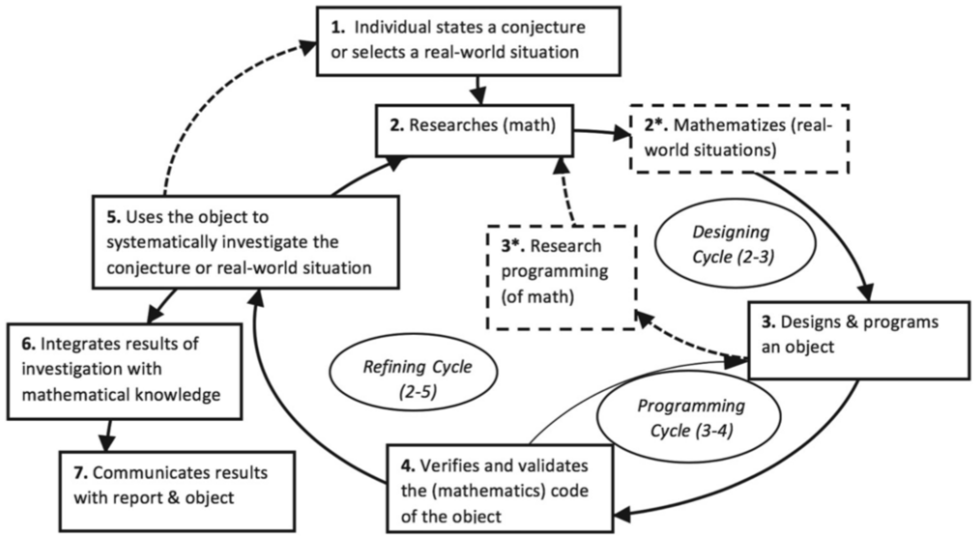
Potential Ways Computation May Change Students' Relationships with Mathematics

Based on the framing shown in Figure 2-3, the key underlying conceptualization is that ultimately computational mathematical experiences have the potential to influence students' self-image, habits, and world-image of mathematics and computation, thereby shifting their attitudes towards these disciplines. Although there is not as deep of literature surrounding this mediated effect, there are multiple theorized ways in which literature from both educational domains can be used to inform potential effects.

Programming Cycle in Mathematics and Dispelling Notion of Linear Progression

When students are engaging in programming for a mathematical model or investigation, there is a cyclical nature to their work (Buteau et al., 2020; Marshall & Buteau, 2014). Using the framework from Buteau et al. (2020), the processes that students are engaged in require multiple iterations across the design, the actual programming, and the refinement of the model itself when students are engaging in 'authentic' mathematical work. These nested cycles can be seen in Figure 2-4. Within the design cycle, there are multiple areas for fluency, flexibility, and originality. However, one of the greatest messages within the overall conceptualization of this framework is that the act of authentically engaging in mathematics is cyclical.

Figure 2-4: Development process model of a student engaging in programming for a mathematical investigation or application from Buteau et al. (2020).



A common myth surrounding mathematics is that mathematical ability is innate (Hurst & Cordes, 2017; Leslie et al., 2015). This notion then brings about the connotation in mathematics that either a student gets it - or simply will not. This also brings about an understanding that any initial struggle, or deviation from the 'correct solution path' is an indicator that this student does not possess this innate mathematical ability. Further, when considering lectures within the undergraduate mathematics context, the 'messiness' of the mathematics is often hidden. Students may be presented with a problem and how to solve it from start to end. Or they might receive a proof in complete form with all formal language and none of the scratch work that went into it. Presenting mathematics in these ways produces a logical argument that mathematics is innate, and those who have authority in mathematics have this innate ability and follow a linear problem-solving strategy. Therefore, an indicator of mathematical ability is being able to immediately jump to the solution. In using the framework of Buteau et al. (2020), what becomes immediately obvious is the fact that the problem-solving process is not linear, making space for mathematical creativity to occur. Computation normalizes the problem-solving process not being linear, thereby enabling students to take risks and follow their own line of thinking, because it is simply part of the process.

Coding Offering a Novel Environment for Mathematical Experiences to Occur

Because there is a link between general creativity and mathematical creativity, as previously discussed, then pedagogical approaches that foster creativity in general have the potential to foster mathematical creativity. Coding itself offers a general connotation of creativity, which is something that mathematics often lacks, as it many times is associated with rules and memorization (Boaler, 2018; Riling, 2020; Silver, 1997). When undergraduate students in an introduction to computer science course were asked to reflect on programming, Isomottonen et al. (2020) found that the top category student responses focused on was the freedom to create and express. These students were from a variety of majors and the freedom they identified included the freedom to create, to resolve unforeseen problems, as well as the freedom to express oneself. Although this study was at a single institution, the data consisted of approximately 2,000 students over an eight-year period. This indicates that there is a perception of potential creativity associated with the computational content. That is, within this study the world-image of computation carries an understanding of computation necessitating creativity. In comparison, the world-image of mathematics many times has a connotation of being devoid of creativity and requiring innate ability. Through computational mathematics experiences, students have the potential to lean into the computational world-image and then view mathematics as creative by providing an alternate experience that may be in contrast with their previous mathematical experiences. Therefore, there is the possibility that student experiences in computation can be leveraged to not only promote creativity but potentially change students' relation with mathematics itself. This conceptualization of the world-image of computation and mathematics offers a basis for how computation can provide opportunities for creativity and draws attention to student experiences with computation. Computational mathematics experiences have the potential to influence student's perceptions of mathematics and their relationship with the mathematics.

ACTIVITY THEORY

Throughout this study, I used activity theory (AT) as a conceptual framework that bridges the gap between motivation and action (Kaptelinin & Nardi, 1997, 2006) as it provides a coherent account for processes at various levels of acting in the world (Nardi, 1996). This section will proceed as follows: an introduction to the two key ideas of activity theory will be presented, followed by sections on the individual principles of activity theory. This structure mirrors that of Kaptelinin & Nardi (2006) as this was one of the most well-structured overviews and aids in the understanding of the conceptual ideas within activity theory. It is important to highlight the tenants of activity theory in order to understand the structure of cultural-historical activity theory (CHAT) and its underlying assumptions.

Two Key Ideas of Activity Theory

Activity theory has its origins in Russian psychology, specifically Vygotsky and Leontiev, and as such, many of the key tenants can be traced back to their work. Two of the main ideas underlying activity theory are the unity of consciousness and activity, as well as the social nature of the human mind. The first key idea, the unity of consciousness and activity, comes out of the desire to have studies explore the context of interactions rather than solely focus on the mind. If one is interested in the mind, then it is imperative to understand it in the context of a subject-object relationship. The mind is in fact embedded in the world and therefore the relationship between humans and the world around them needs to be included.

The second idea focuses on the social nature of the human mind, and is a profoundly different approach, especially when compared to the work of other psychologists at the time of Vygotsky and Leontiev. This idea draws on the first idea, in that the human mind is embedded within the world and cannot be understood on its own. However, this key idea goes further in that society and culture are not external influences, but rather they are present for the creation of the mind. A maxim that embodied this notion was "social being determines consciousness" (Marx and Engels, 1976 as quoted in Kaptelinin & Nardi, 2006). Not only are human beings social, but also the world itself is social. Therefore, when compared to other ideas, such as Piaget's constructivism, culture cannot be relegated to a secondary component or a footnote. Rather, the culture and social interactions must be intertwined into analysis. It is not enough to solely look at an individual and their relationship to the object. Rather, the relationships and culture must be integrated.

Object-Oriented

In activity theory, the underlying assumption is that human activity is always directed towards something – their objects (Kaptelinin & Nardi, 2006). Objects represent the intention that motivates the activity (Jonassen & Rohrer-Murphy, 1999). It is worth noting that the object is not a static entity but can be transformed and shaped over the course of activities. Leontiev expresses the object of activity is twofold:

> First, in its independent existence as subordinating to itself and transforming the activity of the subject; second, as an image of the object, as a product of its property of psychological reflection that is realized as an activity of the subject. *(1978, p. 52)*

In this definition, it brings together the notion of the object of both practical activity and of thought – therefore the object can be either internal or external (Roth, 2004). When considering student intentions, this means that they are shaped both internally and externally. This highlights the need to contextualize all interactions and ensure that any claims of the individual are nested in the larger system. Kaptelinin & Nardi (2006) give the example that "a way to understand objects of activities is to think of them as *objectives* that give meaning to what people do" (p. 66). In this way, we can evaluate actions that people take as whether or not they help facilitate the objective, but they do not unilaterally determine the fate of actions. Hence the malleability.

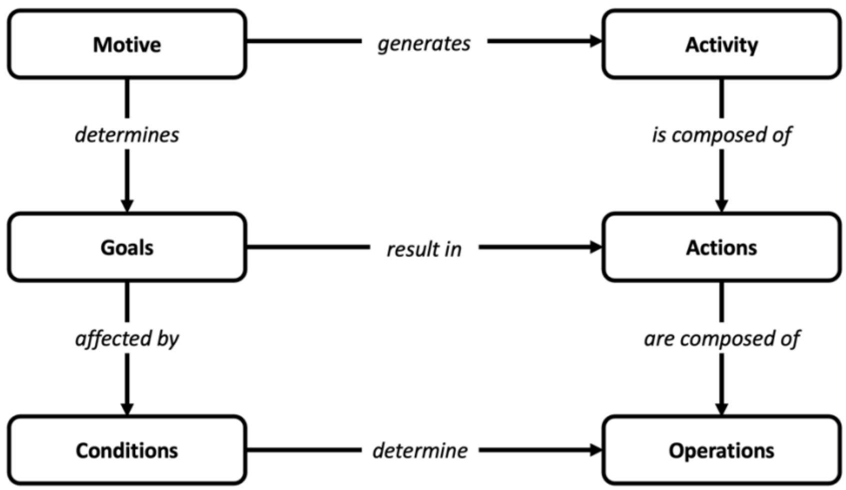
> Human beings live in a reality that is objective in a broad sense: the things that constitute this reality have not only the properties that are considered objective according to natural sciences but socially/culturally defined properties as well. *(Kaptelinin & Nardi, 1997)*

Within this study, it is important to not solely focus on the individual actions, but rather consider how the student objective can shift, especially with regards to social pieces as the students are working within groups.

Hierarchical Structure of Activity

Within activity theory, there are different levels at which you can analyze the relationship between subject and object: *activities, actions,* and *operations* (Leontiev, 1974).The hierarchy between these levels can be thought of as activity is composed of actions, and actions are composed of operations (Batiibwe, 2019). Operations are automatized or routinized behaviors and many times do not require conscious effort (Jonassen & Rohrer-Murphy, 1999). The conditions determine the operations. At the next level up, the goal an individual holds results in actions, but the goal is also affected by the conditions. At this level, it is the functional level (Linnard, 1995) that uses planning and problem-solving to complete the activities (Jonassen & Rohrer-Murphy, 1999). Finally, it is motive that generates activity, which is composed of actions, and the motive determines the goal. This hierarchy can be seen in Figure 2-5 as shown in Batiibwe (2019). This hierarchical structure was used to operationalize activity theory within observations, as there are multiple units of analysis and can be used to build up the overall activity from the operation or action level.

Figure 2-5: Hierarchical structure of activity according to activity theory as shown in Batiibwe (2019).



Internalization and Externalization

Activity theory emphasizes the differences between internal and external activities. This can be directly traced to Vygotsky and his conceptualization of internalization, which argued that internal things cannot exist without initially existing externally. Therefore, in activity theory the internal activities cannot be analyzed or even understood independent from the external activities. Further, internalization is not the eradication of all external activities, but rather the redistribution of resources between the internal and external. Externalization is simply the opposite in that the internal activities are transformed into external activities (Jonassen & Rohrer-Murphy, 1999; Kaptelinin & Nardi, 1997; Roth, 2004). This is critical for group work where activities must be coordinated externally, or for example, if a student uses the command line to modify a variable because the calculation is too complex to do mentally. Therefore, there is a constant shifting of resources between the internal and external which underlies all human activity.

The internal and external is also applied to the development of mental abilities and the relation between the individual and community. There are two stages in this development: the interpyschological and then the intrapyschological (Vygotsky, 1986). Initially functions have to be distributed between others and the individual; however, the switch occurs when there is no need for social distribution. The previously mentioned internalization in this context would be when an individual is able to take and abstract a function that has been socially distributed, whereas externalization is when there is a reallocation of a process' activities. Within groupwork, this is a helpful model to consider how a student internalizes the learning that took place in a cooperative context. The individual cannot be isolated from the community, as there is a constant shifting of internal and external resources. Therefore, the context and community should be accounted for when utilizing this theory.

Mediation

As there is a shifting between the external and internal, artifacts and tools are how individuals interact with reality. Activity will always involve artifacts (Jonassen & Rohrer-Murphy, 1999). When tools are internalized, this impacts individuals' development. Kaptelinen maintains that all "human experience is shaped by the tools and sign systems we use" (1996, p. 10). Just as the mind cannot be understood without the relation of the world, the tools need to be understood in terms of human activity. It is not enough to simply observe the tool, but how the tool is used, the needs it serves, and its evolution all aid in the understanding of the tool itself (Jonassen & Rohrer-Murphy, 1999; Kaptelinin & Nardi, 1997; Kuutti, 1996; Roth, 2004). The use of coding or computation for the learning a different discipline is rich in these mediated interactions. Specifically, this notion of mediation has been studied through the use of instrumental genesis and has been the framing for multiple studies examining computationally mediated interactions (Castle, 2021; Lonchamp, 2012; Maschietto, 2015; Wagh et al., 2017). However, what is critical is that this mediation takes place within culture and context – something that instrumental genesis does not necessarily capture.

Development

One of the key underlying principles that originates in Vygotsky's zone of proximal development is that reality should be analyzed in the context of development. Activity theory is a "formative experiment which combines active participation with monitoring of the developmental changes of the study participants" (Kaptelinin & Nardi, 1997, p. 159). What is important within this principle is that there is always a focus on progressing forward. The student never has a static ability, which is crucial when we discuss the conceptualization of mathematical creativity.

All these principles work together to form activity theory. It is not enough to simply take a sole principle. Rather, it is by the nature of the theory and principles, that when applying any of the principles, the others are engaged as this is an integrated system. These tenants inform not only the CHAT framework that will be used as an analytic framework but also provide some key features to consider when designing and analyzing during a study.

Cultural-Historical Activity Theory Framework

The general conceptualization of activity theory was discussed previously. This theory was utilized for this study through the framework provided by Engeström (1987). The notion of mediation is the underlying conceptualization of his configuration of activity theory, leading to the first generation of CHAT by explicitly bringing the relationship between the individual, community, history, content and interaction of the situation and activity to light (Batiibwe, 2019).

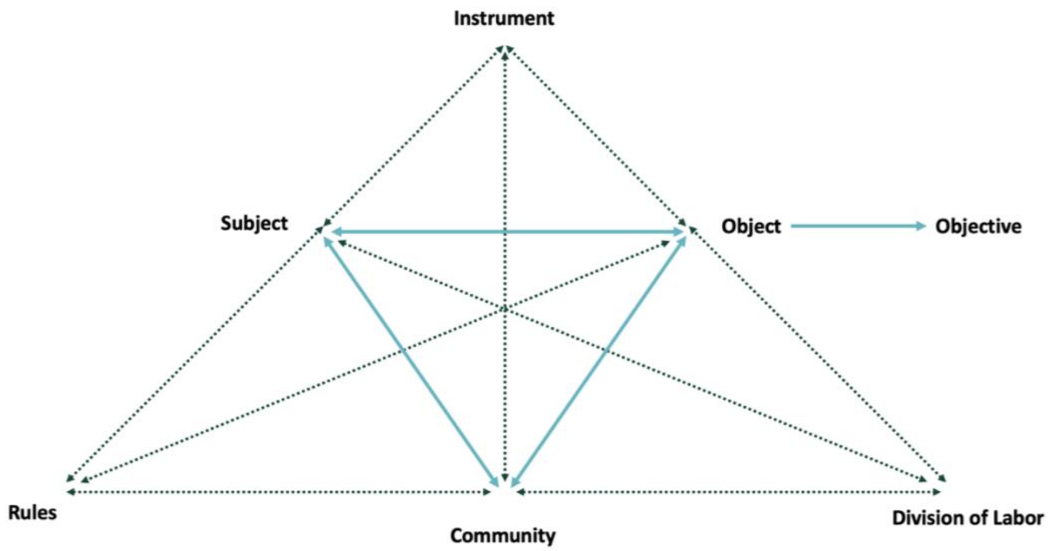
Just as in activity theory, learning can only be understood by attending to the purpose or goals that the participants hold and the context in which it occurs (Battista, 2015). The extensions that CHAT brings are the rules, community, and division of labor, as detailed by Battista (2015) citing Cole & Engeström (1993):

> A community is the social and cultural group that subjects are a part of, with explicit rules or social norms that regulate and influence behavior. The division of labor defines how tasks and responsibilities are shared among system participants as they engage in an activity.

A visualization of the relationship between all the factors can be seen in Figure 2-6. Within this diagram, the original subject and object relation can be seen with the mediation by the instrument and tools, as proposed by Vygotsky. However, where this experience is expanded is within the bottom half of the triangle. The community is also connected to both the subject and object. Further there are rules that mediate the relation of the subject and community. Additionally, the division of labor mediates the relationship between the community and object. All these dimensions bring a nuanced understanding of the activity system to understand the relation of activities, actions, operations, and actions; motive and goals of the subject; and the contextual factors in which the subject operates. This framing has been used extensively in physics education and computational education research and to some extent mathematics education (Roth, 2012).

This representation of CHAT is particularly useful for my study as not only is the subject-object relation mediated through the artifact, but the system includes the community, rules, and division of labor. All these features are key in understanding the activity in which the subjects are engaging. This can be especially important during enactment of curriculum as the responsibility of the completion of the modules is not guaranteed to be equally distributed among all group members, which may result in very different lived experiences for students and alter objects and outcomes. This framework is able to represent a complex interaction and provides scaffolding for the analysis; pilot data will be used to highlight the powerful potential of this framework within Chapter 3, which details the methodology used.

Figure 2-6: Visualization of the cultural-historical activity theory framework (Engeström, 1987) where the light blue solid lines represent the direct interactions whereas the dashed lines represent the mediated interactions.



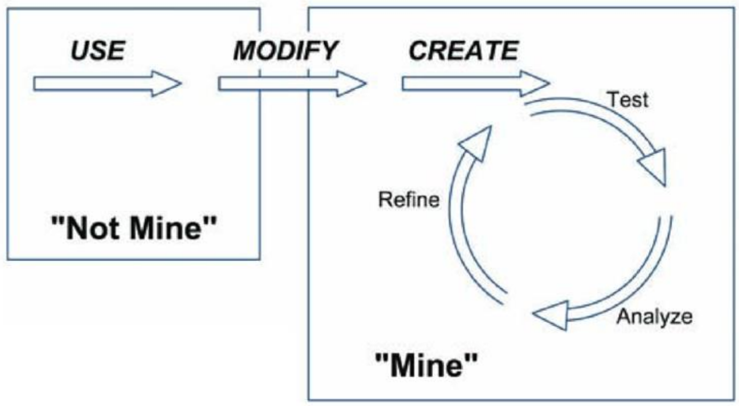
DESIGN FRAMEWORKS

One of the key points of this study is the design of the computational experiences. There are multiple potential frameworks that would be beneficial, but due to my goals of computational creativity and my focus on mathematical understanding, I will review the Use-Modify-Create cycle as well as the understanding by design framework. Both of these frameworks will be operationalized in the methodology chapter.

Use-Modify-Create Cycle

Initially proposed by Lee et al. (2011), the use-modify-create framework is utilized within computer science education to engage students in computational thinking within computational environments (Robins et al., 2020; Scharlau et al., 2019). The use phase of the cycle is when students are presented with a computational artifact where students are expected to be a consumer. The modify phase is when they begin to alter the artifact, and finally the create phase is when students create their own computational artifact that supports their own interest and thinking. A visualization of this cycle is presented within Figure 2-7.

Figure 2-7: The Use-Modify-Create Learning Progression Framework Diagram as proposed by Lee et al. (2011).

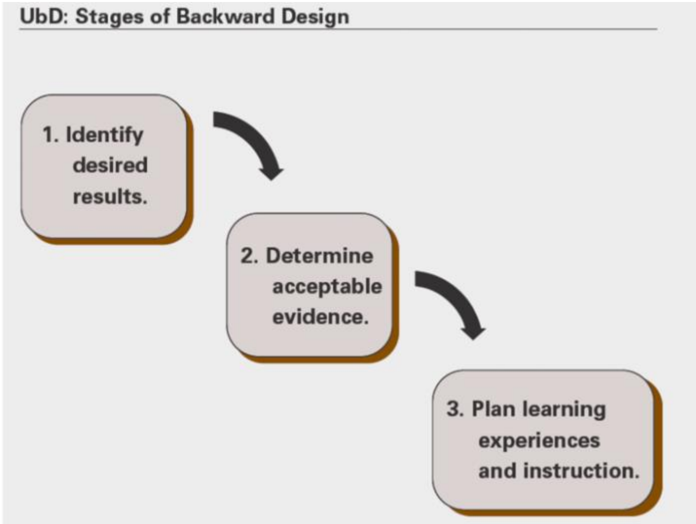


This cycle provides insight into potential design structures, specifically relating to the guiding research question. Specifically, the create portion enables students to engage in abstraction (Lee et al., 2011) while pursuing their own ideas, thereby providing opportunities for both risk and originality. The scaffolding allows multiple points for students to make connections and develop their mathematical understanding, and many times this cycle is coupled with visualization. In addition to the potential for mathematical creativity opportunities, the cycle was also designed as a way of easing student anxieties and developing "appropriate and incrementally challenging experiences" (p. 35). While this is no guarantee of changing student relationship with mathematics, if the mathematics is taught within this computational design structure, then there is a possibility that students may not only be learning in a novel environment but in one that is designed to ease anxieties.

The Understanding by Design Framework

The understanding by design framework is a backwards design process for curriculum planning that utilizes three stages, as can be seen in Figure 2-8. The first stage focuses on identifying desired results. The goal of this phase is to consider goals, content standards, and curriculum expectations to prioritize what students should know, understand, and be able to do. During the second stage of determining acceptable evidence, the focus is to think about a unit or a course in terms of what assessment data is needed to validate that the desired results have been achieved. After these two stages have been completed, then instructional activities can be designed. These activities are meant to support and equip students with the desired knowledge and skills. It is during this phase where frameworks such as Weintrop et al.'s (2016) taxonomy of computational thinking in mathematics can aid in the generation of types of activities to meet the learning goals. The goal of this process is to avoid both aimless coverage of content as well as isolated activities that might be engaging but are not connected to established intellectual goals.

Figure 2-8: The stages of backwards design visualization presented within the understanding by design framework (Wiggins & McTighe, 2005).



A key assumption is that within the first and second step, there must be a way in which you can establish whether a student understands something. Building on the previous theoretical frameworks of activity theory (Vygotsky, 1986), the student would need to externalize their thoughts in order for any assessment to be made. In doing this there cannot be direct evidence of understanding, but rather the focus is on the manifestation of understanding. Therefore, I draw on Wiggings & McTighe's (2005) work on the facets of understanding. These aspects are overlapping and ideally integrated, but they point to how a 'true' understanding can manifest. These facets are displayed in Table 2-2. The definitions given are general, as the goal is to be able to apply to multiple disciplines and contexts. This allows for the adaptation to mathematical content taught in a computational setting enacted through coding.

Table 2-2: The six dimensions of understanding from Wiggins & McTighe (2005).

Dimension of Understanding	**Understanding Dimension Definition**
Can explain	Via generalizations or principles, providing justified and systematic accounts of phenomena, facts, and data; make insightful connections and provide illuminating examples or illustrations
Can interpret	Tell meaningful stories; offer apt translations; provide a revealing historical or personal dimension to ideas and events; make the object of understanding personal or accessible through images, anecdotes, analogies, and models.
Can apply	Effectively use and adapt what we know in diverse and real contexts—we can "do" the subject
Have perspective	See and hear points of view through critical eyes and ears; see the big picture.
Can empathize	Find value in what others might find odd, alien, or implausible; perceive sensitively on the basis of prior direct experience
Have self-knowledge	Show metacognitive awareness; perceive the personal style, prejudices, projections, and habits of mind that both shape and impede our own understanding; are aware of what we do not understand; reflect on the meaning of learning and experience.

Each of these dimensions has the potential to be expanded beyond the design phase and used as an analytic framework to develop codes for actions that demonstrate understanding. These dimensions will be further expounded upon in the methodology specifically with the incorporation of how I have operationalized these definitions using data from my pilot study. This will serve as a rich source of examples but also serves as an initial validation of the theoretical framing.

CHAPTER 3 - METHODOLOGY

THE STUDY DESIGN

The main methodology that this study was designed upon was case study. This choice was for multiple reasons. First, there is a lack of the overlap between mathematical creativity and computation within the literature as previously discussed. Therefore, this case study serves as a proof of existence that students are in fact able to engage in mathematical creativity during computation enacted through coding. The pilot study pointed to this, but a thick description of this was valuable to myself and the research community as a whole. Further, as this interaction and course of study is relatively new, this study serves to extend theory and frameworks to this interaction of mathematics and computation. In doing this, theories can be substantiated and tested to bring about a deeper understanding of the relation between mathematical creativity and computation. Therefore, case study allows for the substantiation, testing, and modification of mathematical creativity theory in the context of computation. Finally, one of the most valuable roles that case study serves for this study is to increase the repertoire of stories of computation enacted in mathematics, specifically with an eye towards mathematical creativity. Case studies allow for a thick description that aids in related phenomena where there are overlapping characteristics (Guba & Lincoln, 1981). Within this thick description of the case, case study methodology allows for studying the interactions that occur between the different elements present within the study and illuminates meaning. It provides insight into the questions of *how* computation can bring about mathematical creativity. For all these reasons, case study was the chosen methodology, and specifically from the interpretivist tradition. Within this tradition, the focus is on an in-depth look at a particular case with human interpretation.

Stake – Constructivist/Interpretivist Approach

The work of Stake (1995) is aligned both with a constructivist and interpretivist orientation. This approach to case study will be used as a key portion of the approach is to be able to understand and discover meaning of experiences within a certain context. As the study centers on computation enacted through coding, with specific design features for mathematical creativity, and enacted through groupwork, the contextualization of student experiences is critical. For example, the lived experiences of students in the pilot study highly differed between two groups. Although the groups had the same gender makeup and the same number of group members, how students experienced the mathematical modules was highly contextualized based on their group work. It is also worth noting that Stake emphasizes the role of the researcher in the study along multiple dimensions and some of the deliberate and intuitive choices a researcher may make. Further, the underlying epistemological approach is relativistic, as "each researcher contributes uniquely to the study of the case; each reader derives unique meanings" (p. 103). Therefore, the reality has multiple interpretations and is subjective based on meaning and understanding. This study did not attempt to lay out an immutable argument, but rather aimed to begin shaping an understanding of the relation of mathematics, computation, and creativity. Note that the data collection mirrors this interpretivist approach as the bulk of the data consists of observations and interviews. While there are additional data sources, as will be detailed in Table 3-1, these data sources are complementary to the interviews and observations, rather than stand-alone measures.

Instrumental Case Study Approach

Continuing to draw from Stake (1995), the study methodology implemented was an instrumental case study. An instrumental case study allows for a thick description of a case while also facilitating understanding of the phenomenon and testing and modifying existing theory. As the broader goal of the research was to investigate the relationship of computation, mathematical content, and mathematical creativity, this methodology allowed for the in-depth analysis of a case while then being able to compare it to other cases, or instances of mathematics enacted through computation, in an effort to understand the relationship and the potential for mathematical creativity. This delineation can be thought of as follows:

> *Issue:* Can computation enhance mathematical experiences for students in a meaningful way and foster mathematical creativity?
>
> *Case:* Students working in small groups engaging with Linear Algebra modules enacted in Jupyter notebooks with the goal of fostering mathematical creativity

Within an instrumental case study, the issue is of highest importance, rather than the case. While the cases to be presented within this study are important, the specific cases are selected with the intention of addressing the larger issue of computation, mathematics, and creativity. Therefore, the need for qualitative categorical data and measurement is highly important rather than direct interpretation. The discussion of the corresponding approach to analysis is discussed later in the analysis section, with excerpts from pilot data, but the focus of the section is on the coding and coordination of data. This is not dispelling any singular instances, but rather it shapes the data analysis portion.

As it was important to get rich descriptions of the case, each group of students working through these modules, the data sources are summarized in Table 3-1, with ways in which they addressed the guiding research question. Once again, the guiding research questions was as follows:

> *How can computation enacted through coding provide opportunities for students to develop and express their mathematical creativity specifically in the context of learning linear algebra?*

The information in Table 3-1 is solely to provide the reader with an initial idea of the data sources and what they aimed to capture. The coordination of these sources with their analysis will be further discussed within the following sections.

Table 3-1: An overview of this study's data sources along with their purpose in answering the guiding research question.

Data Source	How it Addresses the Guiding Research Question
Initial Interview	*Gained an understanding of students' view of the relationship between mathematics and coding, and the relationship of both disciplines with creativity and ability to explore*
	Documented students' previous mathematical experiences and how they typically engaged in mathematics vs. computing
	Developed a profile of student's initial relationship to and conceptualization of computation and mathematics
Initial Survey	*Established a baseline of how students rated elements of mathematical creativity and computation*
Final Interview	*Obtained student narrative of how they experienced the modules and their perception of mathematics, computing, and their relation to creativity*
	Obtained how students conceptualized the different topics introduced in the study. Used the student voice of their mathematical understanding to look at the trends across observations
	Developed a profile of student's relationship to and conceptualization of computation and mathematics after they had completed the modules and potential areas where these were switched
Final Survey	*Provided a general overview of student perspective on how they viewed mathematics and computation and relation to creativity and looked to see if there were any changes*
Observations x6 weeks x3 groups	*Observed how students engaged with the modules and mapped the different features of the computational experience that provided opportunities for fostering mathematical creativity*
	Used student manifestations of understanding during the study to track features that enabled mathematical understanding creation or the constraints
Reflections x6 weeks x8 students	*Used as a check against the observational data analysis to obtain the student voice about their perceptions of exploration and the challenges they faced*
	Used student writing to identify mathematical areas of understanding and highlight the student determined import module features that led to understanding

To provide a thick description of this case, the data collection was across multiple sources and there were reflections at different points in time during the study. The data collection methods implemented are detailed below.

MODULE DESIGN

An underlying portion of this study centered around the design of the computational modules, as the theoretical framework emphasized their role in the mediation of mathematical understanding, creativity, and students' relationships with mathematics. These modules were designed and deployed during the pilot study, and then through an iterative design process were adjusted for the study. This section will highlight the methods used in their construction.

Understanding By Design

A key methodology used as the basis for the construction of the modules was the method of understanding by design, as discussed in the literature review section. This process employs backwards design where the learning goals are established, then potential evidence for meeting the goals, and finally the activity design itself using the prior work. Therefore, to establish the key learning goals for the linear algebra modules, I reviewed multiple textbooks, syllabi from Linear Algebra I (MTH 309) and Matrix Algebra with Computational Applications (CMSE 314), as well as materials from CMSE 314. These specific courses were selected as these are the most common introductory linear algebra courses at MSU and are prerequisites for multiple later courses across disciplines. In reviewing MSU-specific materials, the goal was to gain an overview of what might be beneficial for students later on. As this study and module design was not a prerequisite for anything, nor a required portion of students' educational plans, I had latitude in selecting what I felt was most appropriate for this study. After compiling the potential topics for inclusion within the modules, I centered on the desire for students to walk away with a broad understanding of how matrices can be used to represent various systems as well as for students to develop a geometrical understanding of vectors and their properties. These learning goals were in addition to the broader goal of students engaging in mathematical creativity. These goals were then used to group topics and learning goals, producing a set of 10 potential modules with coordinated learning goals, but was trimmed down to six for the pilot study. Once the learning goals were established, evidence for meeting these goals were written out and used for construction of the modules. A detailed list of each module, the topic, and evidence for meeting goals can be seen in Appendix C. For the sake of space and conciseness, I will use Week 4 as an exemplar which focuses on *Matrices as Transformations*. The full module can be seen in Appendix D, showing the full flow of content, scaffolding, and materials that students received in the Jupyter Notebook. However, I will supply key portions to highlight the process used.

Table 3-2: An example of how key understanding and evidence for understanding were mapped through learning objectives and skills. This was week four which focused on the idea of matrices as transformations.

Week Number	**Linear Algebra Topic**	**Mathematical Learning Objectives and Skills**	**Computational Learning Objectives and Coding Skills**
4	**Matrices as Transformations**	Relate various matrix transformations to geometric illustrations	Reflecting on their thinking and learning in order to transfer to new challenges
		Recognize common types of transformations	Engage in conjecture and computational experimentation
		Interpret a matrix product as a composition of linear transformations	
		Interpret the inverse matrix as representing the inverse linear transformation	Translate mathematical equations into code
		Distinguish between a matrix as a table of numbers and a linear transformation as a function	
		Define the image of a linear transformation	Demonstrate ability to break apart a problem into smaller parts

For the fourth module, the key understanding was matrices as transformations, and the evidence for understanding is listed in Table 3-2 under the mathematical learning objectives and skills. This evidence was based on the potential manifestations of understanding presented earlier in Table 2-1, such as visualization or elaboration. This provided the structure and goals for each activity, but what was left was how to actually implement this design.

Design of Activities

In order to go from the learning objectives and key ideas to an actual Jupyter notebook for students to engage with, there were three main sources that I coordinated between when thinking through potential ideas for activities. These were: the potential manifestation of mathematical creativity within computation, based on Weintrop et al.'s (2006) taxonomy of computational thinking within STEM, the use-modify-create cycle highlighted in Figure 2-7, and the potential manifestations of creativity, presented within Table 2-1.

As mentioned in the literature review, I used the categories of modeling and simulation practices and computational problem-solving practices. Within each of the categories, there were multiple computational activities that I had linked to potential sources of creativity. These were compared against the evidence of understanding to determine the potential enactments of the mathematics within computation. For example, one of the activities presented within the taxonomy was creating computational abstractions nested under computational problem-solving practices. Additionally, one of the mathematical learning objectives was to relate various matrix transformations to geometric illustrations as well as recognize common types of transformations. Therefore, in using this practice to achieve the desired learning objective, students were asked to abstract the matrix transformations they had experimented with in the earlier module where they predicted the effect of a transformation on an image of a cat. Specifically, they had to abstract how the elements within a 2x2 matrix translated to the image of an input matrix (note the terminology was added towards the end of the module). Two of the prompts from the module can be seen in Figure 3-1. This is an example where ultimately students created a computational abstraction. Note that there is no requirement that computation is enacted through coding, as the focus is on designing solutions that will be carried out by an information-processing agent. Therefore, the development of a general matrix transformation that performs certain desired properties would be a computational abstraction. Therefore, the elements within the taxonomy were used as a basis for designing potential activities.

Figure 3-1: Module 4 excerpt focused on students abstracting different types of transformations based on their prior work.

Reflections

There are multipleways that we can reflect our image. What are the matrices that would correspond to the following reflections?

- Reflection across the x_1 -axis?
- Reflection across the x_2 -axis?
- Reflection through the line $x_2 = -x_1$?
- Reflection through the origin

```
In [ ]: # Code to check
```

Contraction and Expansion

- What are the matrices that would correspond to the being able to horizontally contract and expand? *shrink/stretch*
- What are the matrices that would correspond to the being able to vertically contract and expand? *shrink/stretch*

```
In [ ]: # Code to check
```

Another facet used within the construction of the modules was the use-modify-create cycle. This framework was discussed within the literature review, but the general idea is for students to first use an artifact, then modify it, and then create their own. This cycle was used within the structuring of the notebook as a way of scaffolding some of the materials, especially when all mathematical concepts were new. For example, in Module 4 students were given a base example chunk of code with a matrix of points that defined a cat and a function to plot a matrix transformation. From there, they modified the piece of code and experimented with different transformations and abstracted the different types of transformations. Finally, the students created their own transformations and code chunks designed to modify an original image they created. This cycle was used throughout the notebooks as a way of breaking down a learning goal or big idea and providing scaffolding for the students then to be able to create and explore.

Finally, the other source of design ideas came from the dimensions of creativity: fluency, originality, flexibility, visualization, elaboration, and risk. These dimensions were used in the middle and end of module creation to ensure that students had opportunities for engagement in the mathematical creative process. If students never had these opportunities within the module, then there was a potential that the experience would be ‘pushing buttons’ and not only would this lead to a lack of opportunity for mathematical creativity, but it also would constrain students’ opportunities to develop their mathematical understanding. Therefore, I ensured that there were opportunities for a variety of dimensions of mathematical creativity within the final product. Within the fourth module, originality is present within the modify and create portion as students were designing and testing their own transformations and following their own line of thinking, which also coincided with risk. Elaboration opportunities were within how students connect their understanding of functions to transformations and explaining transformation. Visualization opportunities were also within the construction of these analogies and the creation of plots to clarify and present matrix transformations.

THE DATA COLLECTION PROCESS

The data collection process took place over the course of the Spring 2022 semester for the pilot study, followed by the Fall 2022 semester. The timeline is presented in Table 3-3.

Table 3-3: The data collection timeline across pilot and final dissertation study.

Study Phase	Dates	Data Collection Description
Pilot	03/03/22 – 03/18/22	Student Recruitment from CMSE 201
Pilot	03/14/22 – 03/22/22	Initial Student Interviews & Surveys
Pilot	03/23/22 – 04/29/22	Observations of Modules #1 - #6
Pilot	05/01/22 – 05/20/22	Final Student Interviews & Surveys
Modification	Summer 22	Revision of Modules based on initial analysis and student feedback from pilot study, resulting in final eight modules
Dissertation	09/12/22 – 09/28/22	Student Recruitment from CMSE 201

Table 3-3 (cont'd)

Dissertation	09/29/22 – 10/11/22	Initial Student Interviews & Surveys
Dissertation	10/11/22 – 12/09/22	Observations of Modules #1 - #8
Dissertation	12/08/22 – 12/23/22	Final Student Interviews & Surveys

The Research Context

I recruited students from CMSE 201 - *Introduction to Computational Modeling*. This is a multidisciplinary course for students from majors across the university. The core underlying tenant of this course is to use computational models to solve disciplinary problems. Incoming students are not required to have any prior experience with coding or computing and the only prerequisite is a semester of calculus. Over the course of the semester, students are introduced to coding in Python, and developing computational modules - through a flipped classroom employing Jupyter notebooks. CMSE faculty have spent considerable time and resources overhauling this course and ensuring that it is employs a learner-centered approach (Silvia et al., 2019).

I selected this course for multiple reasons including access, student composition, and coding experience. I had connections with the CMSE department that make this class easily accessible as a researcher, as well as I had departmental support from their computational education group. This aided not only in recruitment access but also with material gathering, structural questions, and research support. With regards to the student demographics, this course employed students from multiple disciplines, and anecdotally, many students in the course have expressed mathematics anxiety or hatred. Finally, all students in the course were exposed to key computational concepts and coding syntaxes by the time they began the linear algebra modules. This differed from multiple other studies in which students are either mathematics majors, or in a specific mathematics course. Rather than recruiting students from a mathematics course, teaching coding, and then approaching modules, this design used coding, a familiar language by the middle of the semester, and then introduced the mathematics. The rationale behind this choice is detailed as follows.

It has been documented that students many times have to translate back and forth between the languages of mathematics, coding, as well as their own vernacular (DeJarnette, 2019). Although this can be a rich site for instrumentation to occur, it often leads to a sole focus on one of the areas at the neglect of the other two. Further, it can be overwhelming to students who are struggling with imposter syndrome and can cause the exodus of students from a discipline. Recruiting students from CMSE 201 enables a focus on the mathematics and computation that would best support the learning of linear algebra instead of having to also provide materials for the introduction to coding such as for loops, lists, arrays, etc.… For students who either have faced mathematics anxiety, or a general dislike of the subject, the learning of mathematics in this context potentially enables students to have a safe ground from which they were entering in the mathematics realm. Even if they were unsure of their mathematical abilities, they could be able to feel more confident in their computational abilities, as noted during the pilot study.

Participant Selection

Students had the option of participating in this research study for an honors designation for their CMSE 201 course. To get the honors designation, students had to maintain a 3.0 or above and either complete this series of research modules or do an additional final project for the course. According to the course coordinator, students would spend approximately 20 hours on their final projects, therefore the goal was to mirror this time commitment for students within the study.

It is important to highlight that this sampling did focus on a subset of students from the course, namely those who are in the honors program or who wish to be. It is known that these types of programs can perpetuate inequities with who is accepted, and who is recruited. Further, those students that are choosing the honors program in this course anticipated earning a 3.0 or higher and most likely had more confidence surrounding their coding and computational abilities. The self-selection was exemplified during the pilot study by a student who chose to participate and do the honors credit because he felt that he was doing well in the course and could therefore seek this option. Although many of these students felt confident in their computing capabilities, this did not necessarily reflect their mathematics confidence. As noted in the pilot study, students' mathematics comfort levels were varied, and many noted they were glad to be done with their mathematics courses. Therefore, selecting students from this subgroup allowed for the investigation of non-mathematics majors exploring mathematics in a computational context. Further, by having comfort with the coding, it allowed students to be presented with new mathematical ideas and have the underlying coding syntax abilities.

In the pilot study, the CMSE 201 course coordinator sent an email to all students soliciting participants and directing them to a google form. Students expressed their interest in the project via this form and then were contacted to set up an initial interview. Due to time constraints, the form was turned off after 8 participants had signed up and scheduled their initial interview. This was solely a sample of convenience, but there was a diverse set of student majors and individuals. Three groups were created with 2-3 students in each based on availability.

For the dissertation study, a similar process was followed as an email was sent out during the second week of classes with ongoing recruitment until the end of September. A total of 11 students expressed interest, with nine students scheduling and completing the introductory interview and one student only showing up for the first week. Due to scheduling constraints, students were initially placed into three groups, each with three students. However, due to one student dropping out of the study and another student having an unavoidable scheduling conflict, the final configuration was two groups of four students and a student who would be completing the modules individually. The modules were designed to facilitate discussion, and so while the situation was not ideal, this participant was excited to learn, and I wanted to ensure that she still had the opportunity to explore and earn her honor's designation. Therefore, this student experienced more of a think-out-loud style interaction.

Interviews

I conducted an initial interview with each student for approximately 30 minutes. It was important to establish a rapport with the student as well as answer any questions they had prior to the observations. I wanted to ensure that there was an opportunity for them to ask any questions one-on-one prior to the group work. This interview was semi-structured as it was important to capture participants' thoughts about their comfort with coding and mathematics, their views about the nature of these two subjects, and the degree of creativity needed for computation and mathematics. Further, general information such as their degree, backgrounds in mathematics and computation, and educational plans was also captured within this interview. However, within these interviews it was important that I had the freedom to follow lines of questioning that are either interesting or seem relevant to the context. This interview served as the core for the development of an initial participant profile. The interview protocol can be seen in Appendix A.

Observations

For a total of 8 weeks, students met with their group for two hours per week. All modules were posted on a google drive folder to which participants had constant access. The choice was made not to use GitHub as not all participants may have encountered this yet as well as each individual had their own Jupyter notebook and there was not a need to commit changes to a common repository.

Participants downloaded the module and then began working through the notebook with their groups. During this time, my engagement with the group was minimal. I let them follow their own path and ideas, allowing for the engagement in a hopefully productive mathematical struggle. Nonetheless, there were some key points where I intervened. If the group bypassed a ‘check in with your group’ stop sign, then I ensured that they discussed their answers. Further, based on the pilot study, I jumped in if they had truly engaged in the mathematics struggle, but hit a point where they are asking for help. However, I let the students initiate the ask. Finally, if there was a point of confusion that was simply based on the wording of a question or a mistake that I made (such as stating something was in $\mathbb{R}^2$ when I meant $\mathbb{R}^3$ in a question), I provided a quick directional note.

I collected each individual’s screen recording. This was done so that during data analysis I could see whether or not participants tried something within the code that they had not shared with the group yet. For example, consider the varying levels of social risk when suggesting a novel approach versus first testing the novel approach on a private script and then sharing the approach with the group. In addition, there was a video recording of the whole group to capture if participants moved to look at a screen (such as leaning over to look at their neighbor’s screen) or ignored other participants with their body language. To ensure that the audio was captured, I utilized an omnidirectional microphone that sat in the center of the participants.

A key portion of both the research and experience for the participants was the ability for students to share their screens with one another. Due to room constraints, there was not an easily accessible individual screen students could airplay to or connect via HDMI. There was a projector within the room which students were free to use, but due to the size of the group, students often simply turned their laptop to face one another and then used their laptop to share their results. This was in contrast with the pilot study where students would share their screen to Zoom for others to be able to see. Both of these methods allowed for highlighting an individual approach, but also enabled there to be a shared common space where they could think together, rather than trying to ‘keep up’ with the typing or methods being tried.

Reflections

After each observation, there was a small portion of time set aside for students to complete a reflection document. This allowed them to document their in-the-moment thoughts and emotions, rather than having to recall sometime after. However, the choice was left to the student as some students preferred to write their reflection after having some time to process the new content. Students responded to a series of prompts that asked them to describe what they thought the purpose of the lesson was, challenges they faced, what they felt was rewarding, opportunities for exploring, and what the most helpful parts were. There were also tailored questions from each of the modules to probe the students’ understandings and their conceptualization. This provided variation within the reflections which students in the second phase of the study appreciated, as students within the pilot study wanted some variation in question prompts.

Final Interviews

After the final module, students scheduled a one-hour interview with me. This was once again a semi-structured interview, similar to the initial interview. The interview protocol can be found in Appendix B. The purpose of this style of interview was to have a core set of questions that all participants responded to, especially due to the content questions and student conceptualizations. However, I needed to maintain the freedom to follow the participant’s lead, especially when there were important features of the student’s experience that were not captured in the protocol.

The first section of the final interview focused on the students reflecting on their experience. This included questions to capture both challenge and reward. This was important during the pilot study, as it helped shed light on what the students really focused on – the mathematics, computation, or their intersection. It was also important since this brought to the forefront what the salient features of their experience were. Also, it was through the challenges that many of the features of mathematical creativity could manifest. Students were also asked about whether they felt they were able to explore during the modules. Based on the pilot study, exploration seemed to be an underlying component in the mathematical creativity process, and therefore it was important to have the students identify when they felt they could explore. This also brought about some of the most salient explorations. The student weekly reflections captured the in-the-moment conceptualizations as well as the immediate reflections while it was fresh in their mind. This allowed for the reflection to bring about some of the most notable features for the students to begin to see some of the longer impacts.

As the guiding research question specifically focused on the context of linear algebra, the second portion of the interview focused on the mathematical content. As one of the overarching themes of the backward design linear algebra concepts was multiple representations, students were asked to describe the different ways that matrices could be used for representation. The goal was to allow the student to parse through the different topics that they remembered to see what stood out to them. It also identifies what ways a matrix can be used for representations. I then asked students to explain the concepts of matrices, linear systems of equations, linear independence, span, basis, eigenproblems and Markov chains. These were some of the larger topics in which students engaged. Further, this also provided students with the opportunity to articulate elements of mathematical creativity, such as elaboration and visualizations. Students had the opportunity to describe unique mental visualizations that they developed during these modules.

During this content-based portion, students could look at their previous notebooks that contain their responses to the modules. Although there may be the fear that this would bias student's response towards computation, these notebooks have the formal mathematical definition as well as the computational portions. Based on the pilot interviews, students sometimes pulled up their old notebooks, but no student read the definition verbatim or any portion of the text. Many times, they would take a quick scroll though for a reminder, and then articulated their understanding. As I was not as concerned with memorization, but rather the ways in which they explained the mathematical concepts, this activity targeted the concepts that I am interested, without creating a testing environment that promoted mathematical anxiety.

Surveys

Before the initial and final interview, participants were asked to complete a survey, which is shown in Appendix E. The goal of the introduction of the survey questions was for students to reflect on the nature of mathematics and computing, how they perceived their abilities within the disciplines, and what was needed to be successful in each of these areas. The purpose of having these surveys prior to the interviews was that it enabled me to ask follow-up questions surrounding salient features of their surveys and to gain clarification on their answers. These surveys were used as a potential entry point to begin a conversation with the student.

Student Notebooks

Students maintained their own individual Jupyter notebooks. This allowed students to synthesize the information, and ensured that they could document their current understanding, or key ideas, in their own words. The generation of these notebooks will also be helpful for them if they take MTH 309, CMSE 314, or another course that depends on linear algebra. The generation of individual notebooks also aided me as the facilitator because the notebook captured their written down explanations, giving me insight into their current conceptualizations. In light of the guiding research question and the desire to promote mathematical creativity, these notebooks in combination with the reflections served an additional purpose. Reflective writing activities have helped students develop their metacognitive awareness, self-regulated learning behaviors, as well as their problem-solving and critical thinking skills (Zarestky et al., 2022). As problem-solving and critical thinking skills have been linked in the literature to mathematical creativity, this activity should have aided students in expressing their conceptualization, and in also developing key skills for mathematical creativity.

COORDINATION OF DATA SOURCES AND ANALYSIS IN LIGHT OF GUIDING RESEARCH QUESTION

As there are a multitude of different research sources and sub-analyses, I have given an overview of the analysis in Table 3-4. Each data source is linked to what part of the guiding research question it addressed and how it was used to answer the research question. The final column highlights the analysis across all phases of analysis. Prior to a more thorough discussion of the planned analyses, there will be a discussion of how the guiding research question was answered using different research sources. Further, results from the pilot study will be incorporated to shed light on the nature of the claims, as this was the first phase of the study.

Table 3-4: Overview of the data analysis within the study. Each data source is linked to how the analysis addressed the research question, then the steps of the analysis are given.

Data Source	How does this address the guiding research question?	Data Analysis
Initial Interview	*Gained an understanding of students' view of the relationship between mathematics and coding, and the relationship of both disciplines with creativity and ability to explore*	1. Used in vivo coding to code the transcripts with the goal of identifying salient features of those participants relating to mathematics, coding, and creativity 2. Developed key common codes based on the in vivo codes to create a common base coding system in relation to participants 3. Constructed analytical memos based on the student individual in vivo codes and common code system 4. Developed profiles of each participant
	Documented students' previous mathematical experiences and how they typically engaged in mathematics vs. computing	1. Coded interview for instances of statements where students identify their process for working on mathematics and computational problems 2. Used codes to develop section in student profile that details their reported problem-solving approaches in each discipline prior to the study
	Developed a profile of student's initial relationship to and conceptualization of computation and mathematics	1. Coded for categorization within adapted mathematical and computational experience and attitude framework (shown in Figure 2-3) 2. Developed initial model with student quotes related to each dimension
Initial Survey	*Established a baseline of how students would rate elements of mathematical creativity and computation*	Used the student profile to compare against the initial survey results to check if there were any contradictions or reinforcement of the profile
Final Interview	*Obtained student narrative of how they experienced the modules and their perception of mathematics, computing, and their relation to creativity*	1. Used mathematical creativity codes (shown in Table 2-1) to code transcript in order to identify experiences salient to the student surrounding mathematical creativity 2. Use codes to determine common themes between computing and mathematical creativity.
	Obtained how students conceptualized the different topics introduced in the study. Used the student voice of their mathematical understanding to look at the trends across observations	1. Looked across participants for each question and note any responses that are mediated by elements of the modules and the experience 2. Recorded each student's response to the conceptual questions, and bundle responses by group 3. Compared against the themes developed during observational analysis to determine how the experiences shaped the given responses

Table 3-4 (cont'd)

	Developed a profile of student's relationship to and conceptualization of computation and mathematics after they had completed the modules and potential areas where these switched	1. Coded for categorization within adapted mathematical and computational experience and attitude framework (shown in Figure 2-3) 2. Placed student quotes related to each dimension into final profile 3. Compared against students' initial profiles and note any changes in conceptualization 4. Grouped changes in conceptualization across participants to identify the different ways in which student relationship to discipline had changed 5. Used observations at action level to reinforce these changes, specifically if there were changes in ways of engaging with the material
Final Survey	*Provided a general overview of student perspective on how they viewed mathematics and computation and relation to creativity and see if there was any change*	1. Compared the pre and post survey responses to determine if there were any changes for each individual and check against student profiles to determine if there is a contradiction or reinforcement 2. Aggregated all students to look at trends across students and check if there were any notable change along any of the questions
Observations (x6 weeks x3 groups)	*Observed how students engaged with the modules and mapped the different features of the computational experience that provide opportunities for fostering mathematical creativity*	1. Used initial theoretical mathematical creativity codes (shown in Table 2-1) for each observation at the action level 2. Noted if action mediated by the Jupyter notebook and attach subcodes for each action 3. Constructed analytical memos detailing initial themes of creativity that were seen for each observation or instances that were notable and why 4. Grouped codes along dimensions of creativity and determined if there were common elements among the actions taken in each dimension 5. Grouped codes along whether action is mediated and looked for whether there were common elements among the actions taken in mediated actions 6. Developed a framing of what elements of these notebooks enabled mathematical creativity and how
	Used student manifestations of understanding during the study to track features that enabled mathematical understanding creation or the constraints	1. Used initial theoretical mathematical understanding codes (shown in Table 2-2) for each observation at the action level 2. Noted if action mediated by the Jupyter notebook 3. Constructed analytical memos detailing initial understandings that appear to highlight unique ways of conceptualizing ideas 4. Grouped codes along dimensions of understanding and determined if there were common elements among the actions taken in each dimension 5. Based on the groupings, construct a framing of the features of the modules that provided for student's mathematical understanding and areas that students were constrained in understanding

Table 3-4 (cont'd)

Reflections (x6 weeks 8 students)	*Used as a check against the observational data analysis to obtain the student voice about their perceptions of exploration and the challenges they faced*	1. Coded using mathematical creativity codes and mark if mediated by module experience as determined by student self-report 2. Compared against the themes developed in observations to either modify observational framework or validate
	Used student writing to identify mathematical areas of understanding and highlight the student determined import module features leading to understanding	1. Coded using mathematical understanding codes and marked if mediated by module experience 2. Compared against the affordances and constraints developed in observations to either modify or validate findings

Guiding Research Question

The guiding research question for this study asked *how can computation enacted through coding provide opportunities for students to develop and express their mathematical creativity specifically in the context of learning linear algebra?* To be able to answer this question, the first phase of analysis was to understand student's relationship with both mathematics and computing, as well as identify if students have had prior opportunities to foster their mathematical creativity, whether computational or not. This was accomplished through initial in vivo coding to highlight salient student features followed by another round of coding on all student initial interviews based on the codes generated during the first phase of analysis. This served to identify key features of the student's prior experiences alongside their 'typical' approach to mathematics and computation. From there, analytic memos were constructed to introduce the students and describe how they have previously experienced mathematics, computing, and mathematical creativity. Upon completion of student surveys, responses were disaggregated by student, allowing for the comparison of answers with their profile developed by the initial interview. Any reinforcements or contradictions were be noted within the student profile.

As the research focuses on the computation providing opportunities for mathematical creativity, a large portion of the supporting data came from the observations. Each observation was coded at the action level using the mathematical creativity codes initially presented within Table 2-1. Each instance of mathematical creativity was tagged with whether it was mediated by the Jupyter notebook, as determined by both student self-report and observation. After the initial analysis, I constructed an analytical memo detailing potential initial themes of creativity that were salient within the observation, or any other notable instances that relate to mathematical creativity. Upon completion of all observation coding, the codes were grouped along the dimensions of creativity (i.e., all codes from flexibility together, visualization together, …) to then thematically group and determine if there were common elements within each dimension of creativity. The codes were then regrouped along whether the action was mediated by the Jupyter notebook to determine if there were common features of the computing experience within the mediated interactions that promoted mathematical creativity and whether there were elements of mathematical creativity that did not manifest specifically through mediated actions. Upon completion of both of these groupings, I utilized MaxQDA's questions, themes, & theories (QTT) qualitative analytical tool to investigate the relationship of computational elements and their relation to the dimensions of creativity. This mapped the opportunities for students within the computing experience to the promotion of mathematical creativity. For example, during the pilot study, one of the findings that came to light after coding along the dimensions of mathematical creativity was that the act of experimentation within computation enabled students to engage in risk taking, a dimension of mathematical creativity. This was supported many times with the use-modify-create cycle and gave students opportunities to experiment in ways that were completely new to them, as a student remarked that he knew this was math but that it was different. The analysis section will provide further detail surrounding the codes and examples.

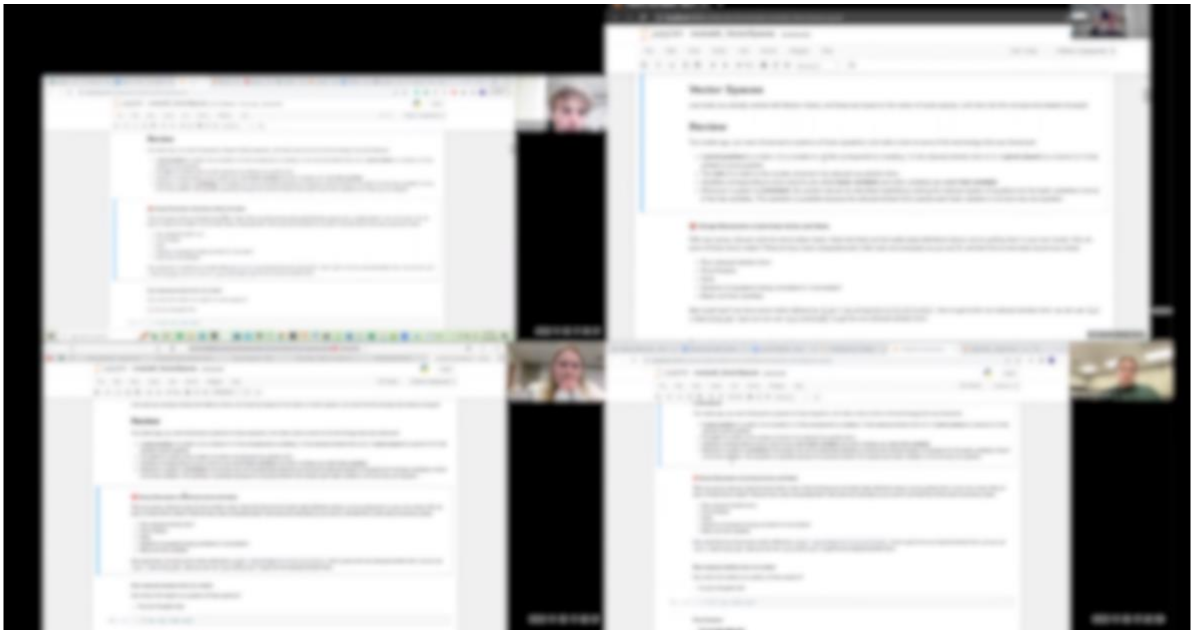
The final interviews with students were used to validate or modify the claims focused on connecting computing and mathematical creativity, alongside with their reflections. This was done by coding each interview with the mathematical creativity codes initially presented within Table 2-1. Then each coded experience was compared against the claims. If the claims did not support the connection of computing and mathematical creativity within the particular experiences, then a modification was made to the claim to incorporate this experience and the process continued on. The final product from this process was a series of claims that link affordances and experiences within mathematics taught using computing and the impact upon the opportunities for students to engage in mathematical creativity.

To answer the guiding research question, it was important to understand how computation, enacted through coding, has the potential to influence student's perceptions of mathematics and their relationships to mathematics. This was informed both by literature and the initial pilot study as potential changes in student's relationship with mathematics thereby influenced the potential for mathematical creativity (Castle, 2023b). Students' initial interviews were coded using the adapted framework presented in Figure 2-3. This framework specifically focused on the linking of students computational and mathematical experiences to their attitudes in each discipline. For each student an initial filled-in framework profile was generated based on the codes, and this provided an overview prior to their experience in the computational mathematical modules. Each observation was coded based on whether students vocalized statements that related to their self-image, habits, or world-image, as governed by the framework. Both the final interview and the weekly reflections were coded using this framework to construct a new profile that included student's new computational mathematics experiences and their relation to the attitudes within computation and mathematics. The initial and final profile were then compared noting the changes that occurred for each student, accompanied with the student self-report of the shifts. These changes were then grouped together across students and thematically analyzed to develop themes in the ways that students' experience with the modules shifted their attitudes towards mathematics. For example, claims from the pilot study utilizing this methodology were how computing provided opportunities to counter student's prior mathematical experiences that negatively impacted their confidence and ultimately generated excitement for the mathematics. This was specifically linked to their mathematical experiences, and how the new experience shifted their self-view in mathematics, leading to a change in their attitude towards mathematics.

DATA ANALYSIS PHASE

From the pilot study, one of the key lessons that arose in terms of the data analysis was the richness of data that a transcript alone cannot capture. Students tried different approaches on their computer, and this was not always verbally expressed, but was documented within their Jupyter notebooks. Further, being able to view students code in live time was a key element that was necessary for the research question. Therefore, MAXQDA was used for analysis. Both the transcription and the video were pulled up at once to fluidly switch back and forth between both verbal and non-verbal communication. Further, to understand the approaches students took in the public and private spaces, students individual screencasts were synchronized to form a grid of student videos, as shown in Figure 3-2. All transcripts were initially processed using Rev's AI transcription and then either edited by myself or sent off for human transcription.

Figure 3-2: Example concatenation of student screens used for the analysis phase. Each student's video was synced with the main audio track providing a live time viewing of different tests students ran. This image is blurred for student privacy.

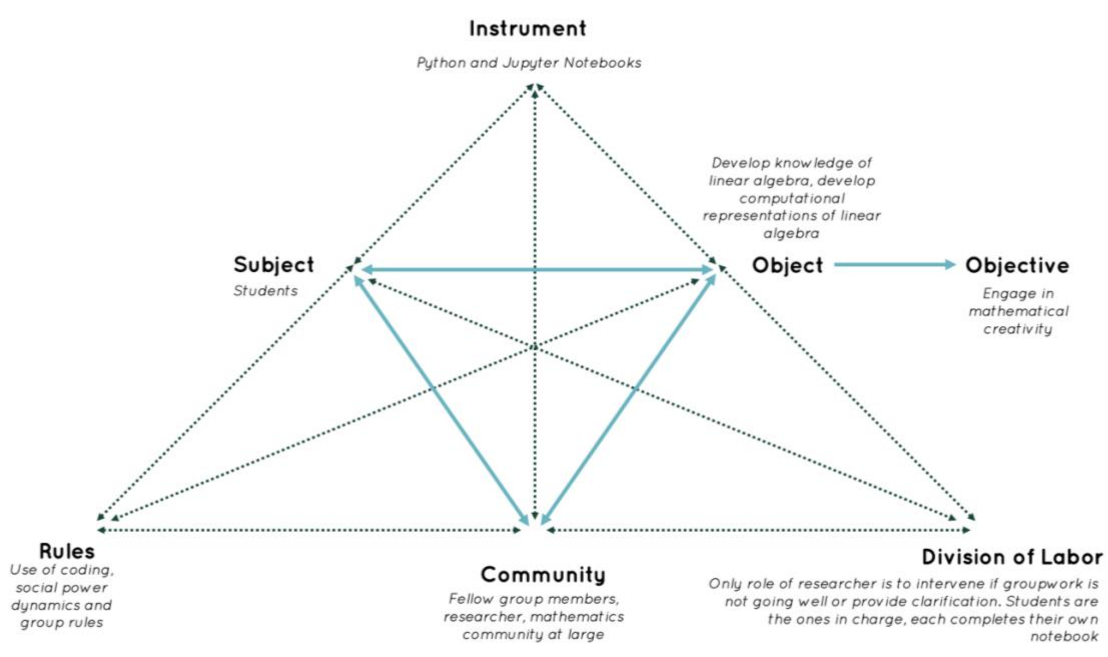


Operationalization of Activity Theory Framework

As previously mentioned, the theoretical framework for this study was cultural-historical activity theory. Figure 2-6 provides the visualization between the elements and relationships within the system. To apply this to the study, Figure 3-3 provides an example of how the activity system was analyzed for the purposes of this study. Due to the work in the pilot study, the sociocultural practices and rules, which mediate students' interaction with the object and community, can be quite broad. Therefore, the framework presented in Figure 2-3 was used to provide a more nuanced understanding of the ways in which these rules can manifest. Namely that within the sociocultural practices I considered, how do students perceive the role of mathematics in the world and what it entailed, how did they see themselves in relation to the mathematics, and what were the resulting ways or habits in which they were allowed to interact. The world-image, self-image, and habits were used to refine the understanding of the sociocultural rules, in addition to other elements such as rules and norms within the groups and social power dynamics.

In the context of the research questions, the key relationships that are of interest within this study are the subject to object relation, mediated by the Jupyter notebook as well as the relation between the participants and the community, which is mediated by different rules. The guiding research question focuses on how the instrument mediates the relationship between subject and object/objective. To operationalize this relationship, there are two foci that will be used to understand the relation. The first of which is using a mathematical creativity framework to code for the action level instances of creativity, and then elevating to the activity level through thematic grouping. The second is using an understanding framework to identify substantiations of the student's understanding of the objective, or potential creative visualizations and elaboration. These are at the action level as well, and then coding and interpretivism was used. These processes will be detailed in the following sections.

Figure 3-3: Application of Engeström's activity theory scheme where the blue solid lines are direct interactions whereas the dashed lines are mediated interactions.



Initial Coding

Engeström's activity theory visualized in Figure 3-3 and the hierarchy of activity systems presented in Figure 2-5 informed the initial levels of coding. Within this hierarchy, the activity was oriented towards a motive, which corresponded with a specific need. Within this study, the need of the students can be thought of as completing the modules while learning linear algebra concepts and learning how to use it in computation. This relation of need and activity was a multi-layer system that has multiple subunits to unpack. During each of the modules that a student worked to complete, there were multiple actions that took place to meet the students' need. This was where the heart of the guiding research question come in, and it was at this level that coding for mathematical creativity and understanding came into play. Within this the hierarchical structure in cultural-historical activity theory framing as shown in Figure 2-5, actions were the conscious processes motivated by goals that then resulted in the attainment of the object. These actions were implemented through lower-level units of activity: operations, which were the routine processes, or could be thought of as the actions you take without thinking. Therefore, when asking the question of how computation could provide opportunities for students to develop and express their mathematical creativity, this mediated interaction manifested through the actions that were taken by students. To identify these actions, the non-mutually exclusive coding scheme presented in the next section aided in this identification.

Mathematical Creativity Coding

As previously stated, the actions corresponding to mathematical creativity were coded. The codes themselves were an operationalization of the definition of mathematical creativity and were not mutually exclusive. As discussed in the literature, these dimensions had not been utilized within a computational and coding context. Therefore, there was an *'other'* category that was used to flag any instances of what seems to be mathematical creativity that was not captured in the current scheme. When using this code, an attached memo was inserted into the transcript describing why I believed this would be a case of mathematical creativity and why the current codes were insufficient in capturing this instance.

All other codes are below in Table 3-5, where the code, definition, example, and subcodes are indicated. The subcodes were based on the pilot study, as these were some of the common attributes that arose, and it aided in the delineation of the role that the tool played in mediating the subject-object relationship.

Table 3-5: The six dimensions of the manifestation of mathematical creativity accompanied by examples of each code from the pilot study, and the subcodes.

Code	**Definition**	**Examples**	**Sub Codes**
Fluency	The ability to apply the same mathematical idea, concept, or procedure, to a variety of problems and situations	"Yeah. If you're, if you're on the same plane in three dimensions, it's kind of like being on the same line in two dimensions."	Idea Generation Pursued Idea
Originality	The ability to try novel or unusual approaches towards a problem. This is contextually dependent, as a novel approach is caveated by what material the students had encountered previously and the current solution path.	"Um, I pretty much, I was trying to do it a diff, I had one idea that I wanted to see, but I wasn't sure. So, I didn't wanna lead you guys down the wrong path." *Note the approach she tried was different than group and from module*	Deviation in solution attempt No prior content *(Note that this solely is based upon what is covered in the modules)*
Flexibility	Ability to use multiple methods (either from mathematics or drawing on an alternative discipline) for solving the same problem	"You can think of it like that. Each of your vertical columns in your, matrix, are like a step you can take. So, like in, in physics where you have i-hat, j-hat, and k-hat, or like 1 0, 0 1." Kylie: I had one idea that I wanted to see, but I wasn't sure. So, I didn't wanna lead you guys down the wrong path. Alex: Can try to do it that way too? Kylie: It pretty much ended up almost exactly the same as, as yours. Um, wait, I'm just gonna, for the sake of knowing how we're doing it, um, there's two, that makes sense.	LA method Computation Physics Other

Table 3-5 (cont'd)

Visualization	Development and use of illustrations (either physical or mental) to clarify or present concepts	"And then I always think of it like, which is like, not really, I guess how you supposed to think about it, but like if it's like a vertical or horizontal line, like how many ticks are on the line is like how many rows are in like, like a table basically, but like the, then the numbers are like how many? I don't know like how many takes there are, but I, that doesn't make any sense. I think it's just how my brain thinks that."	Physical representation Computation Based Mental Illustration
Elaboration	The ability to establish meaningful connections between concepts typically by explaining a thought process in words. The validity of the solution is independent of the meaning made – but the student is engaging in sense making.	"Yeah. *If we multiply anything by zero, we get zero*, but if A inverse doesn't exist, then we're not gonna get the zero matrix back basically. Um, because it's a, it's a thing where it's, it doesn't exist… Um, and if our columns are, uh, linearly dependent in A, then we can't take the inverse basically. And then it breaks."	Previous material Other discipline Other mathematics
Risk	Taking responsible risk in the problem-solving process, willing to take an action where the result is unknown, or is a novel approach, in order to advance problem-solving process. Note that this can entail social risk	Casey: We, we have to experiment. Mobius: Oh yeah. Uh, the experiment. Do we just copy and paste that one? I think we can just copy and paste [the matrix], then calculate [row reduced echelon form] and then change that example. Casey: Um, yeah. We could change them to be the exact same values and see if it changes anything. Mobius: So, can we just pick, we can just pick whatever numbers we want. I guess. Casey: I'm doing the same exact values to see if it comes out the same. Okay. So, if they're the same exact values, it looks a little weird.	Social Risk Unknown result Experimentation

When coding each action of mathematical creativity, an additional tag of whether this action was mediated by the tool – the Jupyter notebook, was documented. The determination of whether the action was mediated by the Jupyter notebook begs the question of if the notebook was no longer there and rather was replaced with a textbook or a pen and paper problem, would the opportunity still have existed? This was a somewhat crude definition, but the main purpose was identifying what opportunities these enacted computational notebooks had. In the pilot study, there were instances where a mathematical definition was provided, but then the students relate it to previous work they did while working on the module and created a mental model. In this case, the action was mediated, and was also an example of elaboration as the student was engaged in sense making and connection of the ideas.

Once this coding was complete, the codes were grouped along dimensions of creativity to see if there were key elements of the computational modules that better mediated mathematical creativity. Similarly, the codes were then grouped by whether actions were mediated by the tools to highlight if there were trends in the ways that mediation (or lack thereof) was elicited. One of the goals during this stage was to identify linkages between the mathematical creativity and the context it was enacted in, or the actions taken prior. For example, from the pilot study, a lot of the risk was taken in a coding/computation setting. During this phase, analytical memos were also constructed to heighten theoretical sensitivity and sense making. These memos also aided in the clustering of data. These were then used to answer the guiding research question surrounding mathematical creativity and computing.

Mathematical Understanding Coding

The process of coding for mathematical understanding was similar to that of mathematical creativity since both were analyzed at the action level. However, during this process the initial coding phase was along the dimensions of understanding. Recalling from the literature review, each of these dimensions came from the framing of understanding by design. In this framing:

An understanding is a mental construct, an abstraction made by the human mind to make sense of many distinct pieces of knowledge. The standard further suggests that if students understand, then they can provide evidence of that understanding by showing that they know and can do certain specific things (Wiggins & McTighe, 2005, Chapter 2).

Therefore, as we do not know what was occurring in students' minds, as well as from activity theory we cannot analyze the mind without the social element, the result is a set of operationalized manifestations of understandings. The six dimensions of understandings are presented within Table 3-6. Each of the explanations are the definitions within Understanding by Design, and there are also examples from the pilot study that help illuminate how this can occur within the mathematical and computational context. The absence of different dimensions did not mean that a student did not understand, as when considering the activity theory framing, there could be other rules or division of labor that are influencing what actions they took towards the completion of these modules with their group. Further, if a student stated "I do not know" in the context of a concept then this did show metacognitive awareness, as they identified where their struggle was, or where they did not understand.

Table 3-6: Dimensions of understanding from Understanding by Design (Wiggins & McTighe, 2005).

Dimension of Understanding	**Understanding Dimension Definition from (Wiggins & McTighe, 2005)**	**Examples from Pilot Study**
Can explain	Via generalizations or principles, providing justified and systematic accounts of phenomena, facts, and data; make insightful connections and provide illuminating examples or illustrations	"Yeah. If we multiply anything by zero, we get zero, but if A inverse doesn't exist, then we're not gonna get the zero matrix back basically. Um, because it's a, it's a thing where it's, it doesn't exist… Um, and if our columns are, uh, linearly dependent in A, then we can't take the inverse basically. And then it breaks."

Table 3-6 (cont'd)

Can interpret	Tell meaningful stories; offer apt translations; provide a revealing historical or personal dimension to ideas and events; make the object of understanding personal or accessible through images, anecdotes, analogies, and models.	"It just says is the system consistent. Does at least one solution exist? … So, it means like the lines aren't right on top of each."
Can apply	Effectively use and adapt what we know in diverse and real contexts—we can "do" the subject	"It kind of made me think like, what else can I do with math and computing at the same time? And like, can I use this on previous math? Like I can totally code a function to do it for like derivative of this!"
Have perspective	See and hear points of view through critical eyes and ears; see the big picture.	(After talking about very detailed code pieces) "This probably would've been easier if I just talked through it for a second, but instead of just vomiting things. Okay but so we want that. So, we're wanting all our squared values?"
Can empathize	Find value in what others might find odd, alien, or implausible; perceive sensitively on the basis of prior direct experience	Kylie: Um, I pretty much, I was trying to do it a diff, I had one idea that I wanted to see, but I wasn't sure. So, I didn't wanna lead you guys down the wrong path. Um, we Alex: Can try to do it that way too? Kylie: It pretty much ended up almost exactly the same as, as yours. Um, so I we're good. Um, wait, I'm just gonna, for the sake of knowing how we're doing it, um, there's two, that makes sense. Discussion about approaches followed
Have self-knowledge	Show metacognitive awareness; perceive the personal style, prejudices, projections, and habits of mind that both shape and impede our own understanding; are aware of what we do not understand; reflect on the meaning of learning and experience.	"I don't understand the last line. So, from the example, one above V1 and V2 are linearly independent. However, if we look at the set of V1 and V2..."

The goal of this process is to be able to highlight ways in which students were able to express their understandings during the mediation of the module with linear algebra, and specifically whether there were any creative mental constructs or visualizations. These were used as tags to probe potentially interesting conceptualizations and to compare with ways that students demonstrated their understanding during the final interview. An important element to note is that each of these dimensions do not incorporate any element of 'correctness'. Students were able to demonstrate their understanding (whether validated by their communities) in a multitude of ways.

This process was used to be able to begin to assemble a narrative of the affordances and constraints that computation provides for student's mathematical understanding of linear algebra.

Student Relationship to Mathematics Analysis

The guiding research question focused on mathematical creativity, and yet this implicitly brought about a focus on students' perceptions about their relationship to mathematics, because the ways in which students were enabled or constrained by computation were also dependent on their relationship with the mathematics. The framework represented in Figure 2-3 was operationalized in order to answer this question. As a reminder, this framework connected students' mathematical and computational experiences to their attitudes towards each respective discipline. The original hypothesis was that the computational mathematics experience could influence changes in their attitudes in each discipline by impacting their self-image, world-image, and habits in both mathematics and computation. Therefore, after the initial interview I coded for instances of student's previous experiences, self-image, habits, and world-image towards mathematics and computing. These were then used to develop an initial profile with student quotes placed within each of the categories presented Table 3-7. The same process of coding and then developing the profile for students after the experience was done after the final interview. I then compared this against students' initial profiles and noted changes along any of the dimensions. Changes were grouped across participants to identify the different ways in which student relationship to discipline had changed, and general themes were developed using MAXQDA's QTT feature to combine codes, memos, and other analytical tools. Finally, the observations at the action level were used to reinforce these changes, specifically if students demonstrated changes in the ways they engaged with the discipline. As the focus was on computation enacted through coding, any coding experiences were documented within the computing experience, even though engaging in coding may not necessarily have been engaging in computation.

Table 3-7: Experience and relationship codes based on Figure 2-3 with definitions for each code and examples based on pilot data.

Code	Definition	Examples from Pilot Study
Mathematical Experience	Any engagement with mathematical ideas, content, or classes	"A lot of my math lectures were just, you know, someone was standing at a whiteboard writing things down and you were writing the same thing down on, um, a notes page that they printed out for you, and you were not absorbing any of the information cause you were just focused on writing what they were saying down. And then, you know, you'd go home, and you'd do homework, and it would just be like, oh, I like, I'm just gonna copy what these notes say."
Computational Experience	Any engagement with computational ideas, content, classes, or coding	"And I took an AP computer class in high school, and I really liked it. So that's why it kind of led me in the direction."
Self- Image *(Computing)*	Confidence and perception of one's own skills and position to computing	"I have like a pretty high level of confidence [in computing] that whatever I have to learn, I will be able to understand, I have a good understanding of the basics of a lot of those structures."
World-Image *(Computing)*	Notion of computing and computer science	"I think about like my cell phone and how the apps are coded in my cell phone and how like technology and everything is all intertwined with coding today."
Habits *(Computing)*	Patterns of computing and of problem-solving	"In my mind, I don't exactly type it in my mind. I just think of the words that translate into code. So, the words of how to solve the problem"
Self- Image *(Math)*	Confidence and perception of one's own skills and position to mathematics	"I wouldn't say math is not ever really been my like strong suit. … I never really had a lot of confidence about like solving mathematical problems and like being able to kind of conceptually master them."
World-Image *(Math)*	Notion of mathematics	"[Math] was honestly more of straight and narrow. I think my experience with math has been getting equations, someone telling me an equation and memorizing it and then memorizing where to use it and where to plug numbers in. I think obviously like it's problem-solving. But I think it was more of a if you have the right equation and you put the numbers in, it's gonna work and you really just have to keep doing that over and over again. And there's no amount of, I don't know, at least in the math, the level of math I got, there's not a lot of like creativity that you can use to solve problems. You're pretty much, there's one way to do it. And if you don't know it, you're not going to, you're not gonna find the answer. And also, if you do know it, then you're gonna get the answer and it's gonna be right. And there's really no other, there's nothing else you're really gonna run into. I mean, you plug your numbers in, and you get an answer and it's either right. Or it's wrong."

Table 3-7 (cont'd)

Habits *(Math)*	Patterns of mathematics and of problem-solving	"That's where I struggle with the math cause I'm the type that goes back and I double check all my answers on my test. I, if I have time, I go back and I give it all over again. So that's when I, that's why I think I struggle with it so much is cause I second guess, and there's nothing telling me if it's wrong or not."

PARTICIPANTS

One of the key portions of CHAT is to understand the history of a participant and a given activity system. Therefore, it is important to introduce the reader to the participants from both the pilot study and the second phase of the dissertation study. An overview of the participants can be seen in Table 3-8. All names presented are pseudonyms and the students' majors, status, and prior mathematical and computational background are given within the table. Each of these students had unique backgrounds and experiences that will be thread throughout the results section. However, this is simply to serve as an initial point to familiarize the reader with some of the participants and highlight the range of experiences.

Table 3-8: This table provides an overview of all participants both within the pilot and the full study. Note P1, P2, and P3 all refer to groups within the pilot study and D1, D2, and D3 refer to groups within the main dissertation study.

Name	Major	Minor	Year	Group	Prior Mathematical Experiences	Prior Computational Experiences
Alex	Physics	-	1	P1	Through Calculus III	Introduction to Programming Logic (Python)
Kylie	Psychology	Quantitative Data Analytics	3	P1	Through Calculus I	Computer Science I (Python)
Ivy	Data Science	-	1	P1	Through Calculus I	AP Computer Science Principles
Lee	Data Science	-	2	P2	Through Calculus II	Scratch in Middle and High School
Versha	Data Science	-	1	P2	Through Calculus II	Computer Science I & II (Python and C++)
Ezra	Physics	Mathematics and Data Science	3	P2	Through Calculus IV, Linear Algebra, Transitions, & Analysis	Brief introduction to Python within Physics course
Colton	Statistics	Pre-Med	1	P3	Through Calculus II	Java & Python in high school
Micah	Biochemistry	Japanese	2	P3	Through Calculus BC	Python in high school

Table 3-8 (cont'd)

Allison	Actuarial Science	Data Science	4	D1	Through Calculus IV	Computer Science I & II (Python and C++)
David	Actuarial Science	Mathematics and Data Science	5	D1	Through Calculus IV, Linear Algebra, Transitions, Abstract, & Analysis	Computer Science I (Python)
Nate	Physics	Computer Science	1	D1	Through Calculus III	Physics seminar introducing Python
Ash	Astrophysics	Philosophy		D1	Through Calculus III	AP Computer Science Principles
Theo	Biochemistry & Molecular Biology		4	D2	Through Calculus II	Statistics course which utilized R
Jack	Chemistry	Data Science	3	D2	Through Calculus IV	Introduction to visual basic & HTML coding
Izzie	Astrophysics	Data Science & Classical Ancient Mediterranean Studies	2	D2	Through Calculus IV	Introduced to Python in astronomy course but focus was on data visualization
Harper	Applied Engineering	Data Science	1	D2	Through Calculus III	Statistics course which utilized R & Introduction to Engineering course which introduced MATLAB
Olivia	Biochemistry & Molecular Biology	French	2	D3	Through Calculus I	None

CHAPTER 4 - RESULTS

This research aims to add to the growing body of literature focused on investigating how computing can be used as a pedagogical approach within a mathematical context. This work is centered on the research question: H*ow can computation enacted through coding provide opportunities for students to develop and express their mathematical creativity specifically in the context of learning linear algebra?* This chapter will serve to answer this question based on the methodology laid out in the prior chapter. From the analysis, five main claims emerged:

1. Computation, enacted in a creative environment, enabled opportunities for experimentation within mathematics through prediction and reflection cycles.
2. Computation, enacted through coding, aided in the facilitation of connecting multiple representations.
3. Computation provided new opportunities for students to expand their views of the nature of mathematics.
4. Computation provided a novel environment that challenged previous negative mathematical experiences and allowed for shifts in students' mathematical self-image.
5. Computation enacted through coding provided the opportunity for students to develop new mathematical habits and strategies.

RESULTING CLAIMS

Each of these claims will be discussed in detail by first providing a more thorough explanation of the claim followed by evidence compiled across sources according to the analysis plan dictated in the prior chapter. Then these claims will be used to answer the guiding research question and highlight the affordances of computation for mathematical creativity. However, prior to the discussion it is important to acknowledge that these claims are bound to the activity system that the computation occurred. These claims are not generalizable for all of computation, as specifically will be discussed. Computation on its own does not guarantee anything about the enactment of the modules, the commitment of the instructor, the norms of the classroom. All of these are critical elements of the system that impact students lived experiences. However, these claims serve to highlight some of the affordances of computation within this study, and cause reflection for some of the more general potential affordances.

Claim 1: Computation, Enacted in a Creative Environment, Enabled Opportunities for Experimentation Within Mathematics Via Prediction and Reflection Cycles

This first claim centers on how the modules provided opportunities to engage in experimentation and try new ideas within mathematics. This was accomplished through prediction and reflection cycles in which students engaged. These cycles were both built into the modules through various prompts and design choices as well as were instigated by students' own curiosities, such as when students asked "what if" questions. It is important to note that the claim does not mean that general coding enabled these cycles, rather it was through the modules within the Jupyter notebooks where this occurred. Further, there were other ways in which computation enabled experimentation and exploration of mathematical ideas. This exploration spanned individuals, groups, and modules, and arose in a multitude of ways. As will be detailed later in this section, during the matrix transformation module students were able to experiment with different matrices of their own choosing to predict and observe the effect on an image and abstract the different types of linear transformation through reflection. The experimentation allowed for risk taking and originality in their choices. The students guided their own exploration with respect to the matrix transformations and were supported through the computational environment. Other examples included students wondering if they could extend a mathematical concept in the code, such as developing a non-cube parallelepiped where the code enabled them to try different approaches to test their idea. Within the examples for this claim, the students were not ‘pushing buttons’ as some fear, but rather they were making claims, or positing potential explorations, and then using the coding to put their ideas into action. The prediction and reflection cycle that was present deviated from the students’ previous mathematical experiences, as they said the idea of play or creativity had not been present in their prior classes. Additionally, students were able to explore and follow their own inquires which also differed from prior experiences. Coding alone did not guarantee this exploration through prediction and reflection, but it was one of the key structural components of the computational environment that enabled this pattern.

During the fifth module, students were introduced to Markov chains. They were asked to write a function that would take in a state vector, a stochastic matrix, and the number of observations, and return the final state vector. They used the function to predict the population levels of a related suburb and city after 1, 2, and 5 years. The following exchange occurred within one group after writing their functions with the code seen in Figure 4-1.

> Kylie: I'm just like running them all. Uh, it basically just keeps going in the same direction.
> Alex: I think eventually it would, I don't know if we did, like, a thousand that would probably make our computers freeze, but um, would eventually it balance out somewhere.
> Ivy: Maybe we can try like 10 and see what happens.
> Alex: Maybe balance out, balance out isn’t the right word. But eventually it would like 50-50, the directions would've reversed. Yeah.
> Ivy: Let's try this. Yeah. It reversed at 10. So, the city is less than the suburb.
> Alex: I mean like eventually they would get like eventually a year would go by and the city would increase while the other one would decrease.

Within this example, a key point is that students were never explicitly prompted to explore beyond the 5-year mark. Rather, they were designing a function to be able to easily model the Markov chains, and then testing that function to check predictions for a model. After completing the task at hand, the group demonstrated both originality and risk by asking a question that was of interest with a prediction about their question. Note that they had not specifically encountered the concept of steady state yet. After this prediction they tested their hypothesis using their function, and then reflected on the output itself. Within this example, the computational environment enacted through coding enabled a cycle of prediction and reflection, which ultimately later led to a concept of steady state vectors. The coding enabled function creation, which allowed for an efficient way to engage in experimentation. The students were able to create a prediction and then check their result.

Figure 4-1: Alex's code (a) and Ivy's code (b) when testing the population levels of suburbs and cities using Markov chains after 2, 5, and 10 years.

```
# Put Code Here
M= np.array([[.95, .03],[.05, .97]])
x2M = linalg.matrix_power(M,2)@x0
#x2M = linalg.matrix_power(M,5)@x0
#x2M = linalg.matrix_power(M,10)@x0
print(x2M)
```

```
[[0.56544]
 [0.43456]]
```

(a)

```
x0 = np.array([[.6],[.4]])
M = np.array([[0.95, 0.03], [0.05, 0.97]])
x2 = linalg.matrix_power(M,2)@x0
x1 = linalg.matrix_power(M,1)@x0
x5 = linalg.matrix_power(M,5)@x0
print(x1, x2, x5)
```

```
[[0.582]
 [0.418]] [[0.56544]
 [0.43456]] [[0.52329334]
 [0.47670666]]
```

(b)

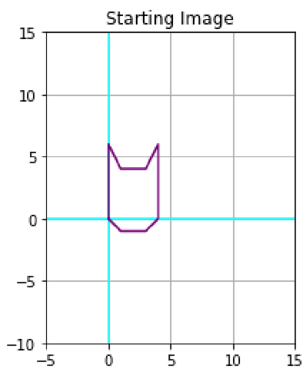
As seen previously, students were willing to experiment without being explicitly prompted; however, there were instances within the modules where they had explicit guidance to do so. For example, during the fourth module which can be seen in Appendix D: Example Module, which focused on linear transformations and matrices as linear transformations, students were asked to experiment with the visualization of matrix transformations. During this task, multiple groups engaged in predicting the different transformations and reflecting on the visual result, which then allowed for the abstraction of general matrix transformation structures such as shear and reflection. After each of their experiments, students reflected on the visual results, which then informed their next iteration of testing. The code both prompted the prediction and reflection cycle through errors that appeared as well as validated students' predictions during the abstraction phase. This set of tasks within the module allowed students to determine the changes to make to figure out the corresponding effect, thereby developing a deeper understanding of the system. Students noted how different this was from their prior experiences in the mathematics course and how this module was a salient memory due to them being able to choose what to test and to then be able to abstract from that process. The following excerpts will follow different groups' engagement with the tasks as well as student reflections on their experience.

At the beginning of the previously mentioned fourth module, students were provided with an initial set of points and the plot of those points which formed the shape of a cat as can be seen in Figure 4-2.

Figure 4-2: A portion of Module 4 which focused on image transformations. Students were presented with a set of points that formed a cat shape and then asked to make it double the width.

In [25]:

```
# Define some points
x = np.array([0.0, 1, 3, 4, 4, 3, 1, 0,0 ])
y = np.array([0.0, -1,-1,0,6,4,4,6,0])


p = np.matrix([x,y])


sm = p.copy()

#Plot Points
plt.plot([-20,20],[0,0], color='cyan')
plt.plot([0,0], [-20,20], color='cyan')
plt.plot(sm[0,:].tolist()[0],sm[1,:].tolist()[0], color='purple');
plt.grid()
plt.axis('scaled');
plt.axis([-5,15,-10,15]);
plt.title('Starting Image');
```



Students were then given the following prompt: *"Let's say that you now wanted to make the cat face slightly wider but want to keep the same height. How might you want to modify your points matrix? Give it a try below!"* Most students first began by multiplying the x component by some factor – such as three – resulting in the cat being three times as wide. From here students were asked to consider how we could use matrices to transform the cat and make it wider. They were asked, *"If we represent our points vector as p what size of matrix should we use? Let's call this matrix A. Would we want to calculate Ap or pA and why? Talk with your group about what each would mean."* There was a space for a response followed by the following section prompting students with *"Let's say that I wanted to double the size of the cat outline. Discuss with your groupmates how you could accomplish this, write out the mathematical expression, and then implement this below. You can use the code above as scaffolding."* The following vignette will follow the group with Harper, Jack, Izzie, and Theo as they engaged with the beginning of this fourth module.

> Theo: What'd you guys say A was? I think A is a factor when you're multiplying the matrix by.
> Izzie: Yeah, through a number in this case.
> Harper: The vectors are P, yeah.
> Theo: It's confusing.
> Jack: If we just multiply the whole point vector by scaling, you'd be scaling X and Y, and you just want to scale one dimension of it. Not both. So, we need to have a transformation matrix, not a scalar.
> Izzie: How annoying.
> Theo: So, that means that whole matrix would just be full of like, let's say that factor's two. It would just be all twos.
> Izzie: Wait that wouldn't work because it is the same as multiplying all by two. Well, you need the [inaudible] just multiply by twos.
> Jack: Yeah.
> Izzie: Would it be like two and like that X and then one everywhere else?
> Jack: Maybe…
> (General conversation and joking about whiteboard time)
> Izzie: Yeah, [our p matrix is] two by nine.
> Harper: Noted.
> Izzie: So, wouldn't the two be like the X component?
> Jack: I think it would be, because are we scaling
> Izzie: I guess it depends on what you're trying to change, because if it was the height, it would be the Y.

Jack: Because if we're doing matrix multiplication, we want the columns of the first to match with the rows of the second. I think if we're only wanting to scale X by two, we'd want to do two-zero, because then it would be you'd do across then down from there.
Izzie: But would it be zero though? Because wouldn't that make it zero?
Jack: Two times whatever it is. That's zero, but two, one?
Izzie: Oh yeah, okay. I guess. It'll make sense.
Theo: This is the answer? … (Question about moving tables)
Izzie: We want to calculate Ap right? Not pA?
Theo: Right.

Figure 4-3: Code screenshot from Jack's notebook with the error he received after running code trying to scale the points image of the cat.

```
A = np.matrix([2,1])
p3 = A@p

sm = p3.copy()

#Plot Points
plt.plot([-20,20],[0,0], color='cyan')
plt.plot([0,0], [-20,20], color='cyan')
plt.plot(sm[0,:].tolist()[0],sm[1,:].tolist()[0], color='purple');
plt.grid()
plt.axis('scaled');
plt.axis([-10,15,-10,15]);
plt.title('Starting Image');

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [18], in <cell line: 9>()
      7 plt.plot([-20,20],[0,0], color='cyan')
      8 plt.plot([0,0], [-20,20], color='cyan')
----> 9 plt.plot(sm[0,:].tolist()[0],sm[1,:].tolist()[0], color='purple');
     10 plt.grid()
     11 plt.axis('scaled');

File ~\anaconda3\lib\site-packages\numpy\matrixlib\defmatrix.py:193, in matrix.__getitem__(self, index)
    190 self._getitem = True
    192 try:
--> 193     out = N.ndarray.__getitem__(self, index)
    194 finally:
    195     self._getitem = False

IndexError: index 1 is out of bounds for axis 0 with size 1
```

From here, the group goes and writes the code to match their current prediction of how they think the cat can be scaled using a matrix. However, they encounter an error in their code, as can be seen in Figure 4-3. Adjustments were made to the code based on the error that they had an out of bounds index, and then they tried simplifying it to just the matrix multiplication to see if they could figure out what was going wrong, prompting the following discussion:

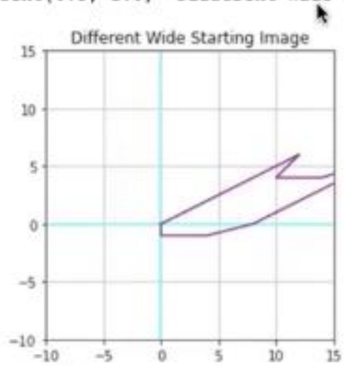
Harper: So, P is two by nine, or is A two by nine? Then other one's one by two, or am I completely off? Should I have not asked?
Theo: I don't know myself, so I can't comment.

Izzie: Well, hmm
…
Sarah: So, your point vector is a two by nine, is what you're saying? Also, you're scaling, meaning you should have the same number of points left over. Then A needs to be something that you multiply by two by nine, so you're left with a two by nine.
Harper: Oh shoot!
Izzie: Why?
Harper: That is a mind [inaudible].
Jack: It needs to be a two by two then, because wouldn't you have the two by two, by two by nine, so it would be kind of...
Harper: Yes.
Theo: So, what would the two by two be then if we want to double the size? It would be two, zero? I don't know. Two, zero, zero, two?

After this point, the group coded their solution which in turn was able to double the cat's width while keeping the height intact using a matrix. Within this excerpt, there is a prediction and reflection cycle where the code errors or unexpected results aid in the reflection. Consider the first trial where the group utilized a 1x2 matrix. They had predicted that this would double the cat outline because the x component would double while the y component remained the same. After coding this they ran into an error, which caused them to reflect on what went wrong and reexamine their original assumptions. As the observer, I did interject after their discussion to clarify that with scaling, the number of points should remain the same. From here, Jack posited the dimensions of the matrix, while Theo supplied what he thought the matrix values should be. The group engaged in small prediction and reflection cycles that were prompted due to an error in the code or an unexpected result. After they had discovered the transformation matrix and how it altered the cat image, Harper began to make changes to the matrix to see what happened, first having one off diagonal entry followed by creating a symmetric matrix with nonzero off diagonal elements, as shown in Figure 4-4. This seemed to trigger the curiosity of her groupmates as Theo exclaimed: "Warped Cat! How does that come about?"

Figure 4-4: Screenshot of Harper's changed transformation matrix and the resulting images.

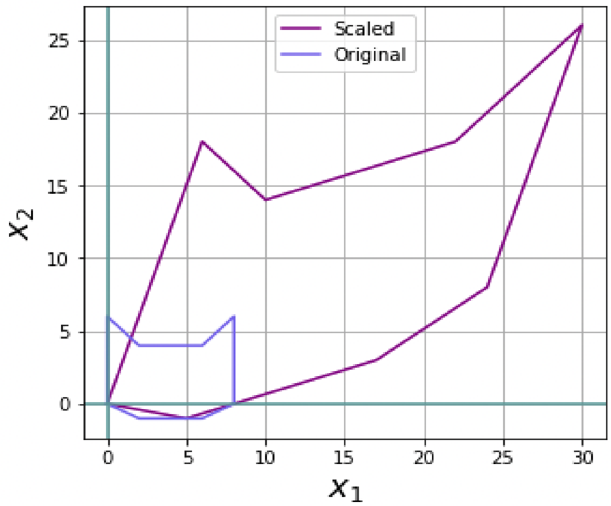
```
In [24]:
# Put code here
A = np.array([[2,2], [0,1]])
s = A@p

sm = s.copy()

#Plot Points
plt.plot([-20,20],[0,0], color='cyan')
plt.plot([0,0], [-20,20], color='cyan')
plt.plot(sm[0,:].tolist()[0],sm[1,:].tolist()[0], color='purple');
plt.grid()
plt.axis('scaled');
plt.axis([-10,15,-10,15]);
plt.title('Different Wide Starting Image')

Out[24]: Text(0.5, 1.0, 'Different Wide Starting Image')
```



There were other changes and explorations that had started such as Theo adjusting his transformation matrix to see if he could make a tiny cat and adjusting the code to turn the cat pink. Throughout these cycles of predictions and reflections, the group could use the computational environment to test their mathematical predictions and they felt comfortable in making changes and exploring what would happen with different linear transformations, especially during the next phase where this was built in.

From this portion, the next phase of the module centered on a structured time where students would be able to experiment just as Harper had begun to. The following prompt within the notebook (following the previous experimentation to determine how to alter the size of the cat) was *"It will now be your turn to explore! See what happens when you try different matrices. Use the cell below to document your thoughts or whatever you find easiest! Try and state what you think will happen before you run the code."* This was followed by multiple cells designated for students to try different matrices. This portion was specifically designed for students to be able to experiment. The following excerpt is from the group consisting of Kylie, Ivy, and Alex.

Figure 4-5: The output from Ivy's Jupyter Notebook when using a matrix transform along a set of points.



Ivy: I wonder if you like had like 3, 1, what would happen like anything that's not a zero.
Alex: Or you wanna do 3, 1, 3, 1. (Referring to a 2x2 matrix)
Kylie: Do you think it will work?
Ivy: Um, we can say, we don't think it will work.
Alex: What do you mean? What do you think?
Kylie: I guess it will just change. Like it'll disproportionately change the width and length.
Alex: Uh, I think the X values will be three times X plus Y and the Y values will be, uh, Y plus three X. So, I think it - the X and Y will turn out to be the same, if that makes sense…
Ivy: So, you think that it's gonna, they're gonna be multiplied by three, but then the value, like a value's gonna be added to them as well…
Kylie: Okay. And then what do we think that's gonna do the graph. It's gonna make it larger and move it?
Alex: I don't think it'll look like a cat anymore.
Kylie: Yeah. It's gonna move him around. So, it's gonna like break up the lines. That is correct. It looks really weird. Oh! *[Output graphic shown in Figure* 4-5.*]*
Ivy: It kind of looks like it just stretched linearly. Like, kind of like on a line.
Kylie: It's it looks like if you took a side of it and just pulled yeah. You like took one of the cat ears and pulled it.
Ivy: So, it's still cat.
Alex: I think it, it kinda looks like you took, like, we took the lines and multiplied them each by like some number, but it looks consistent, you know? You know what I'm saying?

Ivy: Kinda like, it just looks linear to me. I don't know why.
Alex: So, for the next one, what do you guys wanna do?
Kylie: What about if we did it like negative numbers?
Alex: Oh, it'll flip it over. Like the X, Y or something?

This process continued on and the students each completed 6-8 different trials, deriving their own novel transformations, and plotting the visualization. From here, students were then asked to define the matrices that would correspond to the points undergoing: reflection across each axis, lines, and through the origin, as well as contractions and expansions, and shears. During each of these phases students then took what they had learned through experimentation to abstract, make meaningful connections, and create general representations of the types of transformations and the corresponding matrix. If they were usure, then they would experiment, using the insight they gained during the first part. Then they used code to validate their approaches and show the transformation on a set of points.

During this activity, there was specific reference to 'exploring' within the prompt. However, it was previously shown that students engaged in experimentation via prediction and reflection even when not prompted. This new excerpt highlights the potential ways in which the prediction and reflection cycle can take place. Specifically, the students were able to predict the ways that a 2x2 transformation matrix would affect an image and how the corresponding vertices would shift. The experimentation highlighted their initial choice in what matrix to start with and followed by different modifications. This allowed for their thoughts to guide the exploration. Further, during this experimentation, students were not simply "pushing buttons." Rather, they actively engaged in the sense making and making predictions based on their current understanding of the materials. This exploration piece allows for a type of inquiry-based approach to transformations, and the computational environment is something that enabled this to be fostered, especially due to the visualization and the speed in which students could compute transformations.

This was drastically different from previous mathematics experiences for most students. When reflecting on the experience as a whole, Alex stated:

> *I know it was a math, you know, teaching us math, but it didn't feel like math the same way as like solving integral does … it was more so recognizing like what was changing … it felt much more like recognizing patterns... Yeah. Just, just figuring that out. It didn't feel so much like, step by, I move this over here and then simplify this, it felt more like, playing a game, trying to, trying to figure out the way to get to the end goal.*

This quote highlights the fact that the experimentation portion felt different to Alex than previous math classes. Although he does not explicitly name experimentation, he discusses this concept of changing pieces and looking for patterns to get to the end goal, which mirrors the notion of prediction and reflection. Within the previous examples from the observations, the group engaged in making changes to figure out the corresponding effect, thereby developing a deeper understanding of the system. This experimentation fostered via computation countered Alex's prior notions of mathematics, and specifically gave a framing of exploration rather than following a series of given steps, thereby promoting opportunities for creativity. Further, when reflecting on whether modules allowed for creativity, Harper (in the prior group) brought up this module where they were engaging with linear transformations as one that stood out in her memory. Specifically, she said:

> *The little cat graph - the transformations, yep. We had time to experiment and see what different things would lead to different outcomes. And then later you asked us to, you put, I think you put a name to it or something or I can't remember exactly, but you were like, okay, now what do we do to get a transformation over the x axis or something? And we were like - well it just so happens we know exactly that. So, I liked that - a bunch of*

creativity. You gave us time to experiment before and see what each thing would do and how different values inputting in different places would react. And then we went into applying it, which was good.

During this module, Harper actually associated the portion that was prediction and reflection within the module to creativity, as this experimentation prior to application enabled creativity and was prompted by these cycles of creation and abstraction. In her group, it was noted that students were engaging in this prediction and reflection, even if not vocalized, as Izzie stated that she had predicted cat transformations but forgot to record them within her notebook. Multiple students echoed these sentiments and reflections about exploration during their final interview after their experience. For example, when asked about her experience, Allison stated:

I think a lot of it was the modules we were able to, it was we were given something to test out and we could look at it different ways and try different things to see what would happen. I think that we were able to explore and see all the different things would happen. Just here, test it out - have fun.

Within this quote, there was an emphasis on the design of the module in that students were given actual code to test out. This is in line with the module creation process and the focus on the use and modification of code. This enabled Allison and her group to test different ideas and see what happened. Interestingly, this was one of the areas that resonated with Allison, and she felt it was also fun – something that is not always expressed within a mathematics classroom. This emphasizes the role that computational experimentation played in the student experience in addition to the ways in which it was facilitated via prediction and reflection. Interestingly, this sentiment was not isolated to those who were new to linear algebra. For David, who had already taken linear algebra, this was one of the features of his experience that was most intriguing.

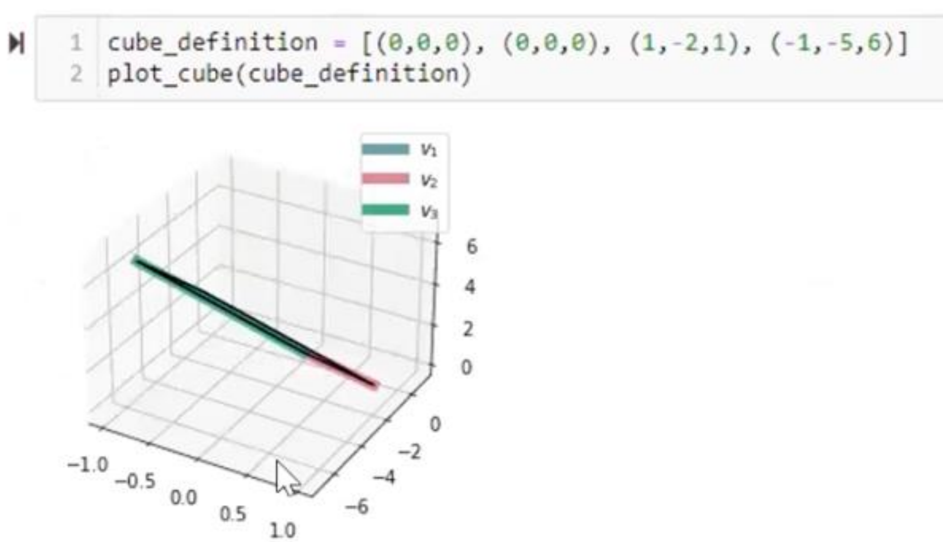
I think the best thing in the modules was when you got to play around with things a little bit - like when we were creating matrix transformations. So, we build the matrix and then get to see the actual transformation on the graph. I never really did that in my linear algebra classes prior, so that was kind of cool, just being able to where there's not even a set right answer, but you get to play around and try new ideas and see how things, if they work the way you thought they would.

Within this excerpt, David specifically calls specific attention to this prediction and reflection cycle in that they would have some sort of transformation matrix and then use the code to test if it affected the image in the way that they thought it would. This idea of exploration, or playing around, is one that he had not experienced before in his prior linear algebra courses. The ability to be able to quickly try different matrices see if the actual transformation on the graph aligned with the prediction was due to this computational environment. Of course, it is worth noting that most of his prior mathematics experiences were lecture-based and without coding, but the focus is on this experience and how computing opened doors that were previously closed.

There were numerous examples of these prediction and reflection cycles, especially when pertaining to visual elements. One of the other modules where this cycle occurred frequently was when students were investigating what a determinant was and how it related to linear independence and dependence. Students were initially provided with the statement that the determinant of a matrix was equivalent to the volume of the parallelepiped defined by the columns of the matrix. Students started with a unit cube to check that this statement holds and then were asked what could cause the determinant to be zero. During this initial phase, Colton had just completed the unit cube task and stated, *Yeah, this is really cool! Yeah, I'm gonna mess with this for a second just because - I dunno it looks neat. [Changes matrix to enlarge the cube 2x2x2 cube and a rectangular prism that is 1.5x2x2] Amazing oh it's actually pretty cool now! We made a rectangle! [Makes a rectangular prism that is 1.5x2.5x2] I want to think about how I could make this a parallelogram, but I don't think that are actually like… Oh wait…*

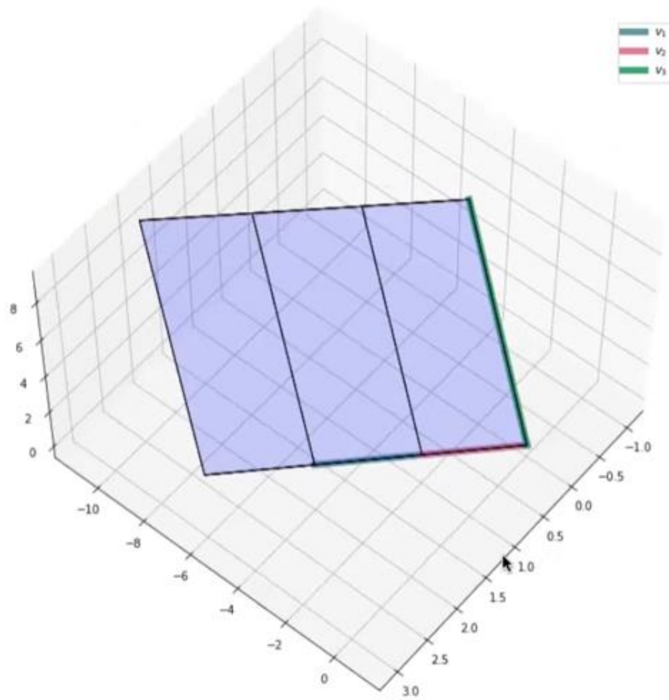
It is at this moment that Colton realized that the next phase allows for the creation of parallelepipeds that were not rectangular prisms. Colton then proceeded to the question asking what would cause the determinant to be zero. He *stated "What would cause the determinant to be zero… Would, wouldn't technically it just mean that it has zero volume - like a zero volume - ohhh so it's a 2D shape."* After this insight he tried setting the first column of the matrix to be zero, checked that the determinant is zero for the matrix that he created, and then plated it as seen in Figure 4-6. In explaining his approach Colton stated, *"So technically I did make the determinant go to zero… it's because the vectors are 2D right?"*

Figure 4-6: Colton's initial plot checking his hypothesis of setting one of the column vectors to be zero would result in a determinant and volume of zero.



I followed up with an additional prompt and asked whether anything else could cause the shape to be two-dimensional (besides setting one component of each vector to be zero). He then exclaimed excitedly *"Oh!! Because these are dependent right?"* while using his mouse to circle the matrix columns of the cube definition. *"So, if we were to put dependent vectors in there"* and then visualizes a matrix where there is not a row or column of zeros, but rather one of the column vectors is a scalar multiple of another column vector, as seen in Figure 4-7.

Figure 4-7: Colton's matrix visualization of a matrix with linearly dependent vectors that are nonzero to check if the shape would be 2-dimensional and have a volume of zero.



During this example, Colton made predictions both in what he thinks the code will do, but also why the mathematics behaves in a certain manner. When initially given the plotting code, he tried multiple different rectangular prisms to see the relationship between the matrix, parallelepiped, and the determinant. After a few of these tests, he decided he wanted to go further and see how he could create a parallelogram. This experience demonstrates that his desired experimentation with the code was built upon his initial prediction and reflection of differing rectangular prisms. After the additional prompt of questioning what could cause the determinant to go to zero, that is when Colton predicted that it is due to the linear (in)dependence of the column vectors and then plotted and reflected to check that this is in fact the case. This demonstrates how his exploration of the mathematical construct of determinant was able to be investigated by predicting what he thought would drive the determinant to zero and being able to then try this in the code.

The previous excerpts and quotes have centered the verbal communication between group members alongside the coding actions taken within the notebook itself. This is not the sole way in which these prediction and reflection cycles occurred; however, the prediction was less obvious and oftentimes placed within inline code comments, quiet utterances that students made to themselves while working, or in the order of their tests. For example, Ash was a part of a group that tended to engage in less full group discussion and performed much of their experimentation on their own. When working with matrices that represented systems of equations, students were asked to generate their own rules about when a system of equations would have one solution, no solution, or any solution. Ash started with an initial matrix, checked the row reduced echelon form, and then came up with another example. Within this cycle, she began to make updates where each row was a scalar multiple of another, or where each variable had scaled in the same manner, as seen in Figure 4-8. It is important to note here that there was not a verbal prediction; however, there was clear evidence surrounding the genesis of these examples to suggest that she was thinking through potential possibilities and the outcome on the reduced echelon form. While all examples were within the same cell in the code, they were individually added after running the prior experiments and each appeared to be crafted in a particular manner.

Figure 4-8: Ash's Screen. Note that calculate_re_reef() was a function that was used to calculate the reduced echelon and row reduced echelon form of a matrix.

```
In [45]:  # Put code here
          ex2 = np.array([[3, 5, 1], [6, 2, 7], [9, 0, 1]])
          ex3 = np.array([[2, 5, 1, 6], [0, 9 ,1 , 4]])
          ex4 = np.array([[1, 2, 3], [2, 4, 6], [3, 6, 9]])
          ex5 = np.array([[1, 1, 1],[2, 2, 2], [3, 3, 3]])
          ex6 = np.array([[2, 4, 6], [3, 6, 9], [5, 10, 15]])
          calculate_re_reef(ex2)
          calculate_re_reef(ex3)
          calculate_re_reef(ex4)
          calculate_re_reef(ex5)
          calculate_re_reef(ex6)

          ------- Echelon form -------
           [[ 1  1  0]
           [ 0  1 -1]
           [ 0  0  1]]
          ------- Reduced Row Echelon form -------
           [[1. 0. 0.]
           [0. 1. 0.]
           [0. 0. 1.]]
          ------- Echelon form -------
           [[1 2 0 3]
           [0 1 0 0]]
          ------- Reduced Row Echelon form -------
           [[1. 0. 0. 3.]
           [0. 1. 0. 0.]]
          ------- Echelon form -------
```

If Ash were to have done this using pencil and paper, it would have been a laborious task. Each of the echelon and reduced row echelon forms would have been calculated by hand and a small arithmetic error would convolute potential pattern recognition and sense-making. In this case, the code provided the space to keep running and testing different matrices and comparing against previous trials. This environment opened the doors to experimentation through prediction and reflection.

The prediction and reflection were not isolated to traditional visual elements with the code as well. For example, when in the module focusing on linear transformations, students were asked whether they thought certain types of transformations were linear, such as an affine transformation. The group with Izzie, Harper, Jack, and Theo decided to code a function that would test the different matrices to determine if all matrix transformations were linear functions. Then they generalized it to an affine transformation and used their function, and they anticipated that the transformation would be linear, since y = mx + b is a linear function. However, they were surprised that this was not a linear transformation despite the likeness to a linear function.

As noted in the initial description of the claim, this claim is not centered on general coding. One of the key affordances of Jupyter notebooks was that students were able to have their predictions and reflections alongside each experimental trial. There was documentation of what they tried, and why they thought the code/mathematics would behave in a certain way. For example, during the second module focused on images and filters, Harper scrolled between her code, inline comments, as well as the visual output across multiple trials. From there she was able to hypothesize what different dimensions of the matrix represented and how to manipulate the code to obtain solely a red image, or solely a blue image. During the matrix transformation module, which was previously discussed with the cat image, students would record their trials of different transformation matrices, just as David referred to previously, and try different variations to understand how the system behaved. For example, Harper alluded to this previously, where they experimented with the system and would try different matrices to see what happened and then were able to abstract from there. Therefore, the claim on computing facilitating mathematical creativity cannot be divorced from the enactment within a creative environment. In all the cases presented, the computational environment facilitated experimentation through prediction and reflection cycles.

Claim 2: Computation, Enacted Through Coding, Aided in the Facilitation of Connecting Multiple Representations

This claim focuses on the ways in which computation was able to support students making connections between different representations of mathematical concepts and ideas. The way in which computation facilitated these connections was twofold. The first of which was the act of coding itself following a prediction and reflection schema, as previously discussed, necessitating the student to view the computational task simultaneously in terms of lines code, but also the output of the code. Secondly, the mathematical prediction and reflection cycles that students engaged in allowed for the development of multiple representations of mathematical concepts. This was due to both the previously mentioned act of coordinating the inputs and outputs as well as coding providing a key affordance of visualization. While students were coordinating their code and output, many times they had multiple representations of a mathematical idea nested within their code and the corresponding output. This enabled students to play with multiple ways to approach a concept or mathematical procedure. The graphical nature and numerous visualization packages present within coding enabled students to create varied representations of how they understood mathematical content. Additionally, the coding environment also provided opportunities for students to engage in the visual prediction and reflection cycle – something that is difficult to replicate using pencil and paper approaches to learning linear algebra. Through the coordination of these multiple acts and affordances, the computational environment that students explored and discovered linear algebra through facilitated connections between multiple representations of key ideas and theorems.

Prediction and Reflection Within Coding as Coordinating Representations

When students coded for computation, they connected multiple resources through prediction and reflection, as they designed a solution to accomplish a specific task. To do this, they had to have a conception of what they would like to produce. Therefore, students had a prediction of what the code would do, in that they anticipated the behavior of the code. This may be evidenced through students stating what they believed would happen prior to and after running the code. If what the code produced differed from their prediction, then often they would comment on this mismatch and enter into another prediction and reflection cycle while debugging. This thereby necessitated student reflection as they examined both the code and the output to compare their prediction to the result. As previously noted, this process could continue to iterate through additional cycles. Within these cycles of prediction and reflection of the code itself, students coordinated the representation of their concept within the code itself as well as the output. Both of these two features did not exist on their own, but rather the code could be thought of as the instructions to make a certain object or execute a task, while the output was the physical manifestation of this process. Students naturally went between these two representations when coding and this is evidenced in a number of ways.

Figure 4-9: Results from cropping task within the second module which focused on how images are stored and edited within Python. The original image (a) was cropped to produce the final image of just one character (b).

(a) (b)

Consider Module 2 which focused on image storage and manipulation within Python. During one of the tasks, students were asked to crop an image to focus on a sole individual within a photo with the results shown in Figure 4-9. When students were attempting to crop the photo, they had a conception of what they wanted the code to do and predicted that their code would generate the desired image. For example, when initially presented with the problem, the group with Harper, Izzie, and Theo (Jack was absent that day) had the following exchange:

> Harper: And then to test it I think - I mean to crop it. I think we can just, instead of having the colons, we could just do whatever it's called. Zero to half point. You know what I mean?
> Izzie: Yeah.
> Theo: So, zero for the first and -
> Izzie: Yeah, because the colons are basically saying the whole -
> Theo: I'm going to do a hundred for the first value and then just a colon for the second and the third.

After writing out the code they thought would produce the desired image, the group ran their code and were met with a black box. This was not what they had expected as Theo stated, *"It's kind of just a back screen and that is the problem."* Izzie then followed up with her new prediction: *"Oh wait, no, because you'll need it. You'll need a value in the first and second because otherwise you are just saying I want this one line of pixels in the y value."* This cycle continued to produce the desired cropped image. This was not in and of itself a mathematical prediction and reflection cycle, as the focus was on the syntax. Rather, the prediction of what the code would do followed by reflection on the output necessitated an understanding of the idea of what they were trying to do both in the form of code and in the output. This positioned students to already be thinking and engaging with multiple representations through the coding environment itself. The group had correctly determined that they would need a submatrix to produce the desired image, but the sticking spot was the syntax. This excerpt highlights the potential ways in which students engage in prediction and reflection cycles while coding, which creates a natural scaffolding to use this method when engaging in experimentation about mathematical concepts.

Mathematical Prediction and Reflection Cycles While Coding Brings About Multiple Representations with Unique Opportunity for Visualization

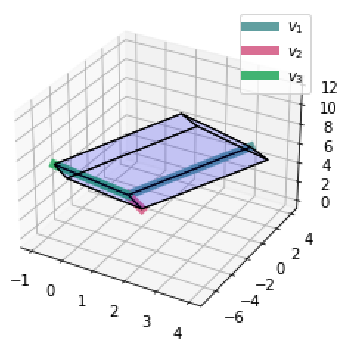
This portion of the claim centers on how when students engaged in prediction and reflection cycles that centered a mathematical prediction, such as those evidenced in the prior claim, the computational environment provided the opportunity for students to make connections between different representations of the mathematical concept. Further, computation enacted through coding brought about the unique affordance of visualization, which was able to be leveraged during prediction and reflection cycles. When students engaged in the coding as previously mentioned, students were already in the process of bridging representations – namely the lines of code and the output from the code. Therefore, within the mathematical prediction and reflection cycles, this provided the scaffolding for a natural way of connecting the tradition of multiple representations, namely students balancing the graphical, numerical, and algebraic representations. Students making connections between these mathematical representations were seen across the observations and facilitated not only flexibility but also the creation of personal visualizations and elaborations. In this study, students represented different linear algebra concepts through coding potential solutions, visualizing the concepts through packages such as matplotlib, or calculating the numerical properties utilizing the code itself, all of which are embedded within their code. In doing this, the previously discussed prediction reflection cycle of the code provided students with the basis to begin to connect their mathematical conceptualizations through multiple representations, as they had done with the code previously. Further, one of the unique affordances of having students code was the access that they had to the suite of visualization and graphical tools available as different packages. The coding environment provided specific opportunities for students to leverage visualization for prediction and reflection, thereby developing a geometrical representation of their mathematical concepts, something that can be difficult to develop through textbook reading and working on paper alone.

The following excerpt follows the group with Kylie, Ivy, and Alex as they engaged with the module on determinants and the connection to the graphical representation and interpretation with respect to the linear algebra concepts. During this interaction, the group members had multiple representations of which to keep track. There were the computational representations of the code and the output, as well as the output itself was both a visual graphic and a numerical output, as seen in Figure 4-10. Within the code the students had to navigate the code description of the parallelepiped as defined by a series of vectors and by columns of a matrix as well as the visualized parallelepiped that was displayed in 3D and was interactive. Further, there was the numerical determinant calculated within the code, as can be seen using the linalg.det() function and the determinant being the volume of the parallelepiped. Students had to coordinate the code itself and the embedded mathematical representations. During this exchange, students predicted and tested their ideas, then reflected on the output and how this connected to the different linear algebra concepts. The act of predicting and reflecting required students to coordinate between the different computational representations, thereby providing a natural environment to structure the connections between multiple mathematical representations. This resulted in the students connecting the ways in which determinants of 3x3 matrices connect to the volume of a parallelepiped and then in turn connecting why a determinant of zero necessitates linear dependence. Therefore, students developed mathematical connections between multiple representations of the invertible matrix theorem as described in the following section.

Figure 4-10: Example of the student code utilized within module focused on determinants. The code for generating the figure was previously defined. The shown cell focuses on the use of that function in conjunction with comparing the numerical determinant value with the visual that is generated.

```
cube_definition = [(0,0,0), (3,4,5), (1,-2,1), (-1,-5,6)]
A = np.array([[3,1,-1], [4, -2, -5], [5, 1, 6]])
plot_cube(cube_definition)
Adet = linalg.det(A)
volume = abs(Adet)
print(volume)

# Complete code here
```

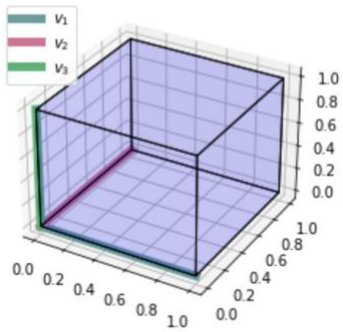
```
83.99999999999999
```



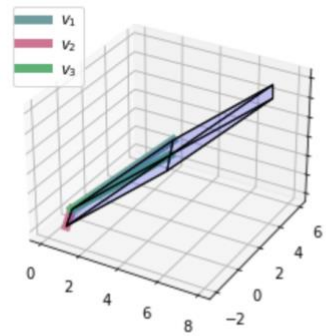
This excerpt begins with Kylie sharing her screen with her group. She began by plotting the identity matrix, which represented a unit cube. The group ensured the determinant was one since the volume of the cube was one and then read through the plotting function noting what was occurring and how to use the function. This is where the following excerpt picks up.

> Alex: So, let's use the columns of A as our vectors. Okay. So, um, I guess that kind of makes sense. ‘Cause the determinant is like, doing the, it's kinda like doing the cross products a couple times. Right. And cross products, like the area of a parallelogram. So anyway, um, that's our parallelepiped… Cool. What would cause the determinant of A, to be zero? Uh, if it were, how do you get a volume of zero?
> Ivy: Would it be like a 2D shape?
> Alex: Oh yeah! Yeah, that makes sense.
> Ivy: But how would we express that?
> Alex: We could try different matrices. So v1, v2, and v3 are probably all gonna be in the same plane is my guess. I guess. Yeah, but how do we –
> Ivy: Should we set one of them to all zeros? No –
> Kylie: We could try it! I guess we don’t need to plot it right now. We’re just looking at the determinant…. Uh, okay. So, we are trying. Yeah. Here is our original -
> Ivy: I’m just gonna change the first value to be zero for all of them. See if that does. Yeah. That is how you can do it… ‘Cause technically you’re saying v1 has like zero value…
> *(Kylie then sets one of the columns of A to be 0 and uses linAlg.det() to check determinant)*
> Alex: Okay so that one is like on the same plane? …
> *(Kylie then runs the plotting code with the new matrix that has a determinant of 0)*
> Kylie: … How does – no. Cause it’s all on the same plane, because one of them is zero. What must be true for one of them to not be zero?
> Alex: They must be on the same plane?
> Ivy: So, if we have three vectors that are all on the same plane or that they’re all, oh wait, like 2D plane…
> Alex: Right. Well, there’s no volume if they’re all on the same.
> Kylie: Plane. Yeah, okay. No, no, okay, got it. Got it. We’re good.
> Ivy: Which means that they must be linearly dependent!
> Alex: … Oh yeah. So, we can say they’re linearly dependent. If the determinant is zero.

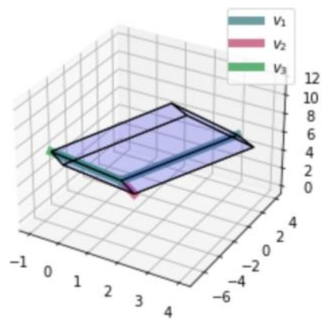
Figure 4-11: Visualizations Kylie created during the exploration of determinants, where she used (a) unit basis vectors, (b) linearly independent vectors, and (c) linearly dependent vectors to check the volume of the parallelepiped and determinant.



(a)



(b)



(c)

The group then went on to check another set of known linearly independent vectors to verify the determinant is not zero. The three visualizations produced through this experimentation and reflection are shown in Figure 4-11. The activity used a series of prediction and reflections which thereby leveraged multiple representations as well. Students then discovered the relation between the determinant being zero and the linear dependence of vectors, providing opportunities for flexibility and visualization.

During the excerpt, the computational representations were the code itself and the output of the code, referring both to the plot of the vectors in three dimensions as well as the numerical value of the determinant. The code plotting the parallelepiped had students decompose the matrix into a set of vectors to calculate the volume to determine the determinant while simultaneously using the definition of a matrix to calculate the determinant. It was up to the students to determine the test matrices, how the matrix was then used to determine the parallelepiped, and then calculate and visualize the determinant. The plotting code was provided but for all other representations of the determinant, students developed the code to test. This necessitated a switching between multiple representations. Further, when they discussed the notion of determinant, they drew on the algebraic notion alongside the graphical interpretation, leading to the discovery of the impact of linear dependence on the determinant. The coding enabled them to engage in these prediction and reflection cycles using visualization tools which developed their multiple mathematical representations of different concepts.

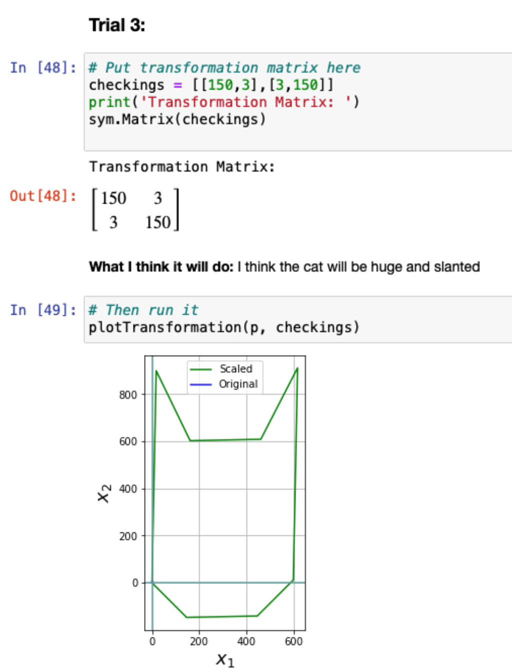
This was not the sole instance of the prediction and reflection cycle within this module, as the next example focuses on the same module but Colton's experience. Specifically, Colton expanded the given code in hope of creating a three-dimensional parallelogram. During this process, he initially coordinated the representation of the 3x3 matrix and the visualization of this matrix within the code and engaged in the prediction and reflection cycles to determine the relationship. He then developed the code and the scenarios to examine what could cause the determinant to go to zero. It is through the code and corresponding visualization that he was able to develop a conceptualization of the determinant being zero, the loss of a dimension when visualizing, and then the effect on linear dependence, as seen in Figure 4-7. Once again, the computation facilitated the prediction and reflection cycles, which brought about Colton connecting multiple mathematical representations.

The exploration of the determinant was not the sole module, as computation being able to support multiple representations was across modules, participants, and groups. For example, consider the aforementioned matrix transformation activity where students were asked to try different matrices acting upon a series of points and then abstract different types of transformations, such as reflection, scaling, and sheer. An example of part of the completed task is shown in

Figure 4-12. Within this activity, students connected the matrix both as an object in and of itself, as well as an action in which they could transform the set of points. It is important to note that the very design of this activity was implementing prediction and reflection cycles, and so connections to multiple representations are rooted in computational prediction and reflection. The conceptualization of a matrix was not a static entity, rather there were multiple representations of the purpose and use of a matrix. For example, in Allison's weekly reflection on the matrix and linear transformation module, she stated that she viewed *"matrices as vectors or a system of linear equations for the most part. For linear transformations, I'm thinking of it as a function that is altering the original value."* Jack stated, *"matrices I mostly see as vectors just because most math is done by either rows or columns, and transformations I see either as matrices or functions depending on the context."* Within these excerpts, students began to view matrices in a variety of ways and how a matrix can have multiple interpretations. Jack highlighted a more algebraic understanding of the matrix in the relationship between transformations and matrices. Whereas Theo stated that he conceptualizes matrices as *"in Python as ([number, number]) and in writing them out they have rows and columns and can be added/multiplied by scalars."* This points to more of a numerical understanding of a matrix. Within the module itself, Izzie posed the question of how to do a vertical stretch on her cat and Theo responded to use the second part of the matrix, bringing about a more geometrical understanding of how the matrix acts upon the points matrix. This idea of matrices as transformations was quite noticeable for Ezra because when he was reflecting on what modules we felt were most beneficial he stated, *I feel like the linear transformation module really sticks out for me for that since thinking of matrices as linear transformations in themselves, I had never previously paid attention to that. I had only applied the ideas of linear transformations to the systems. So, using the matrices themselves as linear transformations to change images and affect plots via shears, which I previously hadn't seen, uh, that really stuck out to me as like a different way that I could apply the coding knowledge and how I could actually use these linear algebra concepts.*

Even though Ezra had already taken a linear algebra course, it was through these computational prediction and reflection cycles and coding the matrix transformations that allowed for a new understanding of matrix. The combination of defining matrices within Python, using them within the code, and also visualizing transformations within this module allowed for the prediction and reflection cycles that students engaged in to naturally support multiple representations.

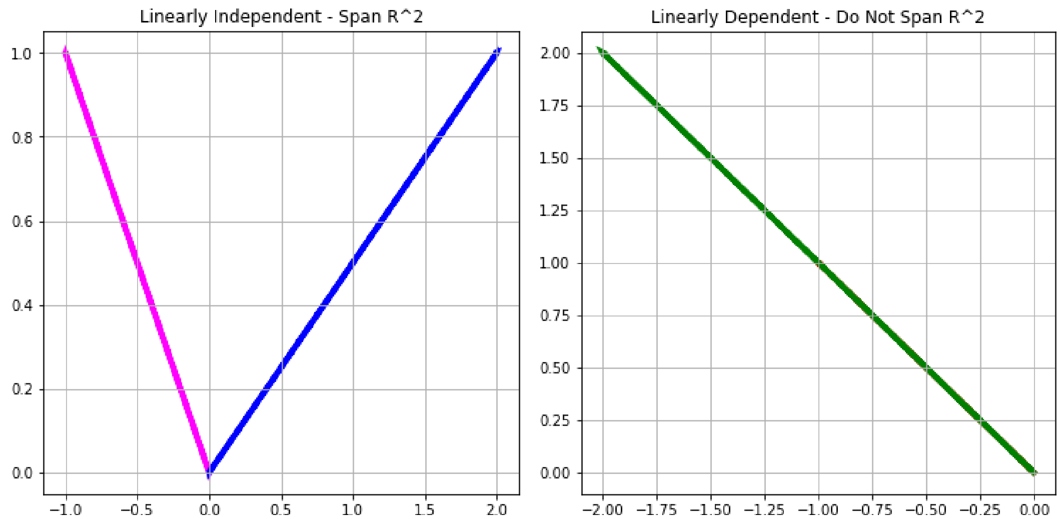
Figure 4-12: An example of a trial taken from Harper's third trial within Module 4. This module examined linear transformations and matrix transformations. Students were asked to generate test matrices, predict the effect, and then plot the result.



Within the modules, students had opportunities both to explain their own conceptualizations of different mathematical ideas and to create visualizations (whether mental or physical) that they felt captured the given mathematical concept or demonstrated their understanding. During the sixth module, students engaged with the concepts of linear independence and span. Students engaged in prediction and reflection cycles surrounding conditions for a set of vectors to span a given subspace and how to use row reduction to determine if a set of vectors were linearly independent. For example, one group tried different matrices composed of vectors they knew were linearly dependent and linearly independent to look for a pattern. From here students were then asked to put span and linear (in)dependence in their own words and then to *"create a visual, simulation, or anything that you feel captures what span is."* Within this, Allison began by stating that *"span is the area that vectors can reach. It is the area that can be covered or reached by linear combinations of certain vectors".* In this example, it is worth noting that both parts of her definition do have a graphical approach in that there is mention of area; however, she also ties in the idea of linear combinations. Allison then created two plots to exemplify her understanding of span. In describing her plots, she wrote,

> *I created two different plots, one with two vectors that are linearly independent and will span all of R^2 and one that is linearly dependent and will not span all of R^2. You can see that in the linearly dependent graph that the two vectors create one line since they are just multiples of one another. With only one line it shows that the vectors will not be able to span the entire R^2 space since they can only go in the one direction. The linearly independent graph shows the vectors going to two separate directions and when you make different linear combinations of the two vectors, they will be able to span the entire R^2 space.*

Figure 4-13: Allison's visualization created to highlight the concepts of span and linearly (in)dependent vectors.



Within this example, it is important to note that the visualization itself would not be difficult to do using pen and paper. However, the speed at which students could try different linear combinations and then visualize the linear combination is something that cannot be matched. Through the computational activities of prediction and reflection, Allison developed multiple representations for span and then used them to summarize her ideas. In the earlier portion, she worked with the idea of linear independence as no vector being a linear combination of the other vectors and how to solve that system algebraically. Further, she was able to extend this to a visual representation of linear (in)dependence in conjunction with the connection with the idea of span.

Summary of Claim

As shown throughout this claim, the prediction and reflection cycles brought about during the coding process were able to aid in the development of multiple representations of mathematical concepts. This was in part also due to the affordances in code because when reflecting on what was helpful, Lee stated, *"Cause I like that, like, I could see the numbers side. Like you can have numbers print out, but like there were also like plots and stuff, so you can visualize it as well. So* *those and like reading through it, you kind of like, it's more like the dictionary way and like get like kind like, and like just learning it for the first time you read through it and then you visualize and stuff. I think those were all helpful."* This quote points to why the coding environment can be so beneficial for prediction and reflection. There are already multiple ways to express the same computational idea, and since Claim 1 established how computation brings about these prediction and reflection cycles, this claim highlights how computation and coding can then be leveraged during these cycles to promote multiple mathematical representations, especially leveraging the different computational representations that Lee discussed.

Claim 3: Computation Fostered New Opportunities to Expand Students' Views of the Nature of Mathematics

The third claim centers on students' perspectives of the nature of mathematics and what doing mathematics entails. Specifically, for many students, mathematics was no longer a single-solution, single-pathway discipline composed of rules and procedures; rather, the process of doing mathematics was opened to exploration, multiple methods, and meaningful discovery. There was a marked shift from doing mathematics by following a set number of steps to centering mathematics on exploration. For example, consider what Kylie said about how she viewed mathematics during her initial interview:

> *[Math] was honestly more of straight and narrow. I think my experience with math has been getting equations, someone telling me an equation and memorizing it and then memorizing where to use it and where to plug numbers in. I think obviously like it's problem-solving. But I think it was more of if you have the right equation and you put the numbers in, it's gonna work and you really just have to keep doing that over and over again. And there's no amount of, I don't know, at least in the math, the level of math I got,*

there's not a lot of like creativity that you can use to solve problems. You're pretty much, there's one way to do it. And if you don't know it, you're not going to, you're not gonna find the answer. And also, if you do know it, then you're gonna get the answer and it's gonna be right. And there's really no other, there's nothing else you're really gonna run into. I mean, you plug your numbers in, and you get an answer and it's either right - or it's wrong.

Within her statement, what becomes evident is this notion that mathematics had a sole answer and that it was simply a case where she either knew how to do it or she did not. She stated that there was problem-solving, but ultimately it boiled down to using equations and plugging in numbers, and this also did not make any affordances for creativity. In contrast, during Kylie's final interview while reflecting on the experience, she reminisced about her previous mathematics experiences and compared them to her computational mathematics experience:

[My math courses] were really just kind of the basic up to calculus. I wouldn't say that I really had an opportunity to [explore]... but that's really something I've only experienced in coding classes so that was, that was definitely new... (Switches to discussing the mathematics in the modules) I think a big part of when we got to be creative is when we were testing things. And I think, I don't know if creative is the word, but I, I definitely think it's like curious of like, okay, well, what if we do this? And what if we do this? Um, you know, how will those things change? What if we wanted to make it look like this? Definitely think with some of the more like visual aspects of what we were looking at... Um, I think there was definitely opportunities for that and pretty much all of them modules, because all of them, we were working through examples and working through, um, kind of bit by bit the concept with lots of like opportunities to visualize all that was possible.

In this quote, one of the pieces that was striking was the lack of comments about memorization or plugging in numbers to get to the final answer. Rather, the focus was on exploration of the mathematical content. This contrasts greatly with her earlier statement where she remarked, *"If you don't know it... you're not gonna find the answer."* Rather than having a complete stop based on whether she could initially solve a mathematics problem, the focus was on following what questions naturally arose and connecting to the visualization. Through the computational environment, she was able to ask questions about the mathematics, such as what would happen if she changed the code to reflect a mathematical principle. One of the affordances of coding was the ease with which she was able to visualize different mathematical concepts, specifically determinants.

This combination of creativity, exploration, computation, and mathematics was best exemplified during her group's exploration of determinants. During this time, they modified code to plot the column vectors of a matrix as vectors and used these to define a parallelepiped in three dimensions. From there, they explored what happened if the columns were the unit basis vectors, three linearly independent vectors, or linearly dependent. These visualizations can be seen in Figure 4-11. During this exploration, a group member brought up that they *"knew you had to multiply a bunch of things"* but that they didn't know why. Rather than simply plugging in numbers to the code or looking for formulas, the group wrestled with the conceptualization and the relationship between determinants, volumes, and linear independence. If she had retained this idea of *"either you get it or you don't,"* then this unique insight into the relationship between concepts would have not come to fruition. These visualizations and abilities to form connections were one of the elements that she highlighted in her quote that were critical to her experience. During her reflection she expressed that the module was *"surprisingly fun"* and that she was able to explore new connections to previous materials. This was a novel way of working with mathematics, and for Kylie, because of this experience, mathematics now had an aspect of exploration that was not present before.

This experience was not isolated to Kylie. During the initial interview, Ivy expressed her frustration because *"Math is based on speed, especially on the tests... The timed sets - you gotta go in fast mode."* She continued by stating that speed is what is valued in mathematics through assessment and connected this to her conception of mathematics:

> *I don't necessarily think that math is a creativity because [the problems] have a set number of steps. You have to do this, this, this, and there's a set value. I definitely think it's very fixed in the, okay, I think that if you don't follow the fixed way, you're not gonna help yourself in the future since everything builds on it. I think that if you don't do it [the instructor's] way, then when they refer to it back in the future, you're gonna be like, oh, what is that again?*

Once again there was a notion of mathematics having a set procedure to follow. Due to her experiences which reified a notion of speed being a marker for mathematical ability, she believed it was better to adhere to the instructor's method of doing something. This was because the fixed way would ultimately aid in understanding in the long run, because the instructor's method was viewed as a building block for the next concept/procedure. When reflecting on her experience with the modules, Ivy revoiced a notion of being curious and wanting to go further:

> *I think that this experience definitely expanded my mathematical thinking with coding. I think that it brought more questions into my mind as to like how far we can go if that makes sense. Like what other things can I do?*

This highlighted that rather than being done after following a set number of steps, there were natural places to go further, and therefore she developed a curiosity and wanted to push her thinking. While speaking about how she viewed mathematics, Ivy noted, *"I still feel like there is like one answer that you've gotta come to. This has gonna be - it's just how you get [there especially with coding] is where the creativity comes in."* For Ivy, mathematics still had a sole answer, but the act of engaging in these modules opened the range of possible solution methods. This was repeatedly demonstrated during observations, as even if her team was trying one method, she would try another method and then compare. She began to value the different approaches and methods within mathematics, and she spoke about how it was tied to this computational experience that was enacted through coding.

These experiences were not isolated to Ivy and Kylie as there was a common pattern of previous mathematical experiences being procedural based with no room for creativity and a focus on a sole procedure. Alex stated that *"in my math class specifically like no creativity at all in my experience,"* and Lee stated, *"I feel like [math is] just numbers in general. Maybe like numbers, like getting answers, like there's usually one answer and so like it's pretty black and white I'd say."* She does continue on to state that there might be more methods, but it boils down to the same solution. Both of these participants expressed opinions at the end of the experience that multiple solution paths to a mathematics problem were possible. Lee added in that individuals can even add in their own style because of the coding and Alex noted,

> *I know it was a math, you know, teaching us math, but it didn't feel like math the same way as like solving integral does or something like that, you know? ... For me in Calc I'll write down like seven steps or something, like I'll look at a, a sample problem and, and write down the steps of how I'd solve it. And I felt like, um, I mean, I could probably do that for*

> *this, but it was more so recognizing a, like what was changing when, when I changed this, what happens if, it felt much more like recognizing patterns... Yeah. Just, just figuring that out. It didn't feel so much like, step by, you know, I move this over here and then simplify this, it felt more like, you know, playing a game, trying to, trying to figure out the way to get to the end goal.*

This quote highlighted that something was different in Alex's experience in this study compared to his previous mathematical experiences which had *"no creativity at all."* Rather than mirroring procedures given in sample problems, Alex centered exploration and the recognition of patterns. This re-framing of mathematics and what it meant to do mathematics was possible due to the space created by these modules. Specifically, through open-ended coding questions and the development of students' own lines of inquiry that they could then test using computing.

For some students, such as Harper, they had positive mathematical experiences overall, but through the computational experience, their appreciation for what mathematics is widened. Harper noted that mathematics was often her favorite subject despite the reliance upon webwork at the undergraduate level. She stated that for mathematics: *"It's just you get the right formula, you get the right numbers, you plug in, and you go."* She enjoyed her mathematics courses, and she noted that this is partly because mathematics just *"clicks."* When engaging in the mathematics through computation, it was much different than her prior mathematical experiences. Most of her prior work centered on individual experiences, which she accredited to the lecture format. Even when working through mathematics homework in a group or with friends, it never had the *"working through it group dynamic"* that was present within this experience. For her, this experience challenged her experiential view of mathematics as an individualistic process and opened doors to a more group explorational approach when learning the mathematics. Further, she had similar experiences to previously mentioned participants with regards to the creativity. Harper commented on how this experience opened her eyes to math in a new way, specifically:

> *I think definitely learning it through a computational, learning this through a computational coding way has had me open my eyes to different ways in math too, as well. Definitely. I think it definitely made me get a more creative view towards math than I had before. So that was nice to see for sure... Before, I guess before with my math classes, obviously was lecture and what you were given is what you were given, you didn't really stray far from it. Whereas the way we learned it here, it kind of lets you - the coding way and the math way lets you, I guess it was a bit broader and you could pull anything. How I connected matrix multiplication, and the dot product were very similar and things like that. I definitely think it led to more exploration.*

Note that in this excerpt from Harper's final interview, she detailed that in mathematics you do not 'stray' away from what the professor has given you. There was a fixed set of things that a student should know, and a specific way of doing things. By incorporating the coding and the mathematics, it allowed Harper to pull multiple concepts together. The example that she gives is matrix multiplication and the dot product. When her group created their matrix multiplication function, there was a notable moment of realization that within their nested for loops they simply needed to take the dot product between the row and column vector of the two matrices. Her connections were not to different fields, or even to different coding methods. Rather, she made a mathematical connection because of the process of coding the matrix multiplication function. In this example, computation provided the environment to see mathematics as full of connections and to explore concepts rather than having a rigid set of understandings. Through this computational experience her view of mathematics and what it means to learn mathematics shifted.

This experiential shift was also echoed by Allison. She stated that she has always enjoyed mathematics and that she has "*always been good at it. [Math] has never been too much of a struggle."* Further, mathematics focused on numbers and how to use it in daily life. Due to her prior mathematical learning experiences, she felt you have to do a problem in a certain way and continue to do that over and over. Further, mathematics has not been a struggle for her. It was procedurally based, and in reference to her mathematics courses she stated, *"I guess I mostly just do what I'm supposed to and move on for the most part."* When reflecting on her experience, Allison noted that she was able to *"come up with different solutions"* and how the goal was not *"replication"* of a procedure, but rather she could explore the mathematics with the given code. With regards to the coding within the experience, Allison reflected,

> *I think overall it allowed you, like you said to, you asked about being more creative. I think it gave us the opportunity to come up with different ways to do it and not always do the same thing over and over again. Whereas in all those other math courses, you're typically taught one way to do it and you're expected to just replicate it over and over again. And you don't often sit and try to find a different way to do something because they're like, you already taught you how to do it, and you just go do your homework, take the tests, and move on. So, I think this had a lot more room for you to come up with different solutions.*

Within this quote, it is worth noting that there were structural differences between her experience coding and her experience within mathematics courses. The role of testing and homework was a key feature that did constrain her experience. Nonetheless, the experience of coding expanded her view of how to learn mathematics, because the coding provided a space in which she could come up with different solutions. As she and other students noted, there was not one way to solve a coding problem, therefore everyone's solution might look different. This affordance of coding could have provided the initial impetus that there are multiple solutions within mathematics.

The allowances that students were making in their conceptualization of mathematics and what the discipline entailed were so salient, that within her interview Izzie actually reflected back on her initial skepticism about whether creativity was required for mathematics.

> *I feel like over time after we've done all the modules, it kind of made me realize, I mean I remember at the beginning and the interview the first one, I was like, oh, I feel like you could be creative in math maybe, whatever. But actually, you do need to be quite creative when you're doing it with coding! Because as we've done on the whiteboard so many times, there's a lot of planning that you need to do to get in. You know can't just have the idea, you've got to plan it, pseudocode it. And there are normally different approaches that you can take in different ways. You can do it to get the same result. So, I feel like you got to be creative in your thinking of maybe, okay, this didn't work for some weird reason. Maybe we can try it a different way and see if, yeah, we can do that...*

Izzie accredited this shift in her view to engaging in mathematics through coding. The coding allowed for creativity and specifically multiple approaches. Rather than mathematics being the *"same set of steps"* done over and over again, there was now an allowance in her view of mathematics for multiple approaches and methods of solving. Further, she noted that there may be times that the code does not work; however, that just meant she needed to try another. This affordance allowed for the development of persistence in mathematics and countered the notion that to do the math well, she had to do it perfectly the first time.

Claim 4: Computation Provided a Novel Environment Challenging Prior Negative Mathematical Experiences and Allowing for Shifts in Students' Mathematical Self-image

The fourth claim centers on the modules providing opportunities to create new experiences in mathematics that could counter students' prior mathematical experiences, especially for those participants whose prior experiences negatively impacted their confidence in their mathematics. Coding was a new environment without the associated negative mathematical experiences. It was simultaneously a completely new way of engaging in mathematics for many, as well as an environment where students were confident and felt they could learn the material. This in turn ultimately generated excitement for the mathematics, and for some it developed a notion of mathematical persistence. For many of the participants, this affordance was made possible by the act of debugging within coding. They did not have the same anticipation for their code to be perfect when they first engaged with a computational problem. Rather, it took time and different debugging cycles, and not only was this okay, but it was part of the established norm within the discipline. Further, allowing the computer to bear the burden of arithmetic allowed students to concentrate on the mathematical concepts rather than be frustrated with errors that are independent of the new content.

For Kylie, mathematics was never her favorite subject, and she had actually avoided it at the start of her undergraduate experience. She felt that she could *"do it in a math class, but it's not gonna be my favorite thing,"* so she never felt mathematics was her strong suit. Reflecting on those prior experiences she stated,

> *I have dyslexia. So sometimes, genuinely, just like the way that the signs look or, or the numbers and letters getting kind of jumbled, like that was always really like confusing for me. And it was really easy to like to miss a negative or write one thing when I mean another*

or not be able to kind of completely track, you know, the order of everything on a like piece of paper. A lot of my math lectures were just, you know, someone was standing at a whiteboard writing things down and you were writing the same thing down on a notes page that they printed out for you, and you were not absorbing any of the information 'cause you were just focused on writing what they were saying down. And then, you know, you'd go home, and you'd do homework, and it would just be like, oh, I like, I'm just gonna copy what these notes say. And I don't, I'm not really like gaining any like understanding of the concepts behind this or like why it works the way it does or where else I can use this.

Kylie's experiences within the mathematics lectures were where she did not feel that she was gaining understanding and, further, the paper and pencil mode in which she had engaged in the mathematics was actually causing confusion and frustration. The ways in which mathematics was being valued and taught isolated Kylie. These attitudes towards mathematics were in contrast with her computational confidence: *"I have like a pretty high level of confidence (in computing) that whatever I have to learn, I will be able to understand, I have a good understanding of the basics of a lot of those structures."* For Kylie, one of the key contrasts between mathematics and coding is that, unlike in mathematics, she was confident that she *would* understand coding and computation. Her mathematics experience centered on not gaining understanding and feeling confused, and yet she was confident in her ability to learn in computational settings. Entering into the project, her confidence towards mathematics and computation were vastly different.

Over the course of the semester both in her course and this project, she began to build up resilience in coding as well as in computational mathematics. In her final interview, she shared,

I think it's more of just like, getting better at trying and failing a bunch and then figuring something out, um, of it not being kind of like a straight and narrow, like you are gonna

> *get from point A to point B. There's not gonna be any problems. And then like learning how to problem solve through that and not get frustrated or too discouraged from kind of the main goal. So, I think more of just building, like, maybe this isn't gonna work a few times and that's okay.*

It is important to note that within this quote, Kylie demonstrated how her ability to problem solve was developing. The larger picture presented is that she recognized that the work is not a linear progression. Specifically, that the work would not go step by step and there would be setbacks. However, her focus is on how to push forward to build understanding rather than ending in frustration. This was in contrast with her previous mathematical experiences of becoming frustrated or feeling that she was not gaining understanding. This experience and the interaction with the modules allowed for her to have the space to explore (as previously mentioned), but it also motivated her and caused excitement. She expressed that, even though her group met on Friday afternoons, they *"want to keep going, want to keep finishing, like, and figuring this out."* These experiences impacted her relationship with the mathematics:

> *I wouldn't say math is not ever really been my like strong suit. I never, would've like even thought like, you know, a year ago that I would've been pursuing a minor in data analytics, just it's pretty stats heavy, and, and has a lot of other elements of math ... I never really had a lot of confidence about like solving mathematical problems and like being able to kind of conceptually master them. Um, so I would say like that it's definitely very intimidating, but I do think that coding is something that actually makes it less intimidating because you are kind of like, you're telling the computer what to do, but also like, you know, there's a lot of built in tools in these packages in all the different software that we use for this that are really helpful for making things a lot more, um, clean and like stream lined ... but it's*

definitely getting better and it's definitely not an overwhelming intimidation now, you know, like I still really want to try and if anything, I think it's pretty motivating

The mediation of coding in her mathematics experience was evident throughout her narrative. The ability to do the mathematics within the Jupyter notebook helped her see the larger picture. The notebook and code format lent themselves to exploration, while providing clean documentation of steps and the ability to incorporate text and calculations. For her, it was a relief not to have to try and do all of these steps on paper and then try and plug multiple things into a calculator. This was in contrast to her previous experience where she felt that in mathematics, she would get small things jumbled, leading to frustration. Specifically, because of her dyslexia, the tracking of mathematics on paper was difficult, which led to her feeling lost and behind. However, the coding helped her organize the procedure, which in turn helped counter her previous interactions with mathematics. She was beginning to gain more confidence, and initially her coding confidence mediated her mathematical confidence. This was also evident during her group work. As time progressed, she began taking more social risk and initiating ideas, asking questions, and providing help to her group mates. While not all of the intimidation of mathematics dissipated, what was evident was that as she was gaining confidence, she was able to explore the mathematics in this novel environment. She wanted to continue to try, and it motivated her. In her final reflection, she stated that she was so *"grateful for this opportunity and it was surprisingly fun"* and that during this final module *"the most rewarding part of this module was reviewing all of the content that I previously was not familiar with and realizing how much I had learned."* Ultimately, she was excited to go on and see how this could apply to her own discipline and incorporate the mathematical computation, especially since this was something she had not been able to do in prior experiences. It was through the coding, resilience, and the overall experience that her relationship to mathematics was affected.

The mediation of mathematical confidence by computational confidence was also seen as Ivy recalled her shift between high school and college. She entered feeling positive about her mathematical abilities but after calculus she struggled and doubted herself asking *"What do, what do I do?"* She added, *"That's where I struggle. Where do I start?... and I don't know why I'm the type of person, like I'll have it all down and then somehow, it'll be wrong."* She discussed how many times her errors would be because of one small mistake, or she would enter a number into the calculator incorrectly.

Ivy expressed her frustration with getting the 'wrong' answer once she started and finished. Ivy's mention of speed, as well as not getting the right answer initially, is indicative of a common mentality among students that to be good at math, you need to be able to do it quickly and correctly. However, Ivy offered a much different conception about computing and coding in that *"errors are a part of the process"* and, when she got stuck or had trouble starting, her process was: *"In my mind, I don't exactly type it in my mind. I just think of the words that translate into code. So, the words of how to solve the problem."* This approach was seen translating over into her mathematical experience after experiencing the modules, specifically during her final interview, she stated,

> *I definitely think that it [the mathematics] was scary to me at first um, it was very overwhelming. But [after a bit] it's like it's okay. We'll work and understand it later. It kind of made me think like, what else can I do with math and computing at the same time? And like, can I use this on previous math? Like I can totally code a function to do it for like derivative of this!*

Within this excerpt, her initial focus was that she had come into this experience apprehensive of the mathematics, but with time she was okay with not immediately getting the solution. She kept persevering, as shown in the observations. If she was stuck with the mathematical concept, she would try to implement it with the code and look at how it behaved under different conditions and was not afraid to ask questions. She was even able to look at previous mathematics concepts with excitement and feel confident in her understanding to be able to implement it computationally. This shift was highlighted when talking about her experience: *"I learned better when experimenting and coding 'cause I kind of learned from my mistakes."* She embraced her mistakes within this context, rather than viewing them as failures like her previous mathematics experiences. Ultimately, through this shift, mathematics became less intimidating, and she felt *"stronger in math"* specifically through encountering the mathematics via computation.

Similar to Ivy and Kylie, the coding environment allowed for Nate's mathematical confidence to be mediated by his computational confidence. Nate entered into the study feeling *"shaky"* about his mathematical abilities. He felt that he did not have the same way with numbers and mathematics that those around him had. He felt that the mathematical concepts presented in the modules were challenging; however, one of the pieces that ameliorated his experience was being able to do the mathematics in Python. The code *"helped [him] get a better understanding of it"* and while he would not call himself a "*linear algebra savant*", he believed that he would be much better prepared for a course in linear algebra. Nate did hedge his learning of the material and stated that he was ultimately limited by his understanding of math, but that in those situations he would take the following approach:

> *When I'd run into something I didn't understand, I'd always try and rely on my knowledge of Python. And so, I feel like the modules where I was able to use by knowledge of Python more, I was able to do more.*

Nate's reflection demonstrated that the coding allowed for another way to be successful in a mathematics context. He felt that even when he did not quite have a grasp on the mathematical concept, he was able to engage with the code to learn the material and be able to contribute. Part of this, Nate noted, was that *"coding really reflects how your brain works, whereas math not so much."* By engaging in the math through coding, Nate was able to leverage his computational abilities, and this resilience and confidence then mediated his mathematical interactions. The use of coding was able to ameliorate some of his views about his mathematical capabilities derived from prior experiences and highlight his strengths.

Although Ivy, Kylie, and Nate began with self-described lower levels of confidence in their mathematical abilities, the computational experiences were still able to counter some of the negative mathematics experiences for those who felt stronger in mathematics. For example, when reflecting on the experience, Izzie stated,

> *Well, okay, this is going to sound weird. I feel like it adds - it takes out the boring part of math and it adds in a kind of fun aspect because I feel like the part, I don't like about math is how to do a problem. But then you've got all these steps and then you forget how to do a step and then you can't do the problem and you get really frustrated. But with coding it - does the actual calculations for you. You just have to know the concept and how to apply it, which is kind of nice because then you can apply it to loads of different scenarios.*

When Izzie described what she disliked in mathematics, and where her prior mathematical frustrations were, she noted that it resided in 'doing' the math problem. However, she was not discussing the act of doing math, but rather the actual calculations that have to be done after grasping the concept or the procedure. This is because she specifically credited the coding with taking away parts of the frustration that come with carrying out the evaluation of a mathematical procedure. In doing this, she was able to center her focus on understanding the concept and how it is applied to different scenarios both inside and outside of mathematics. Further, this quote points to the idea that coding is this novel environment that can counter some of the previously held beliefs related to students' mathematics experiences. Specifically, Izzie mentioned how coding brings this element of fun and joy into mathematics. This was in contrast with the differential equations class she was taking the same semester. During the observations and in her interviews, she referenced how frustrating this course was. Her time was spent doing the same procedure over and over. Even when she felt that she already understood the topic, she had to continue on with the same procedure and only slight changes to the problem. The coding experience not only allowed her to focus on the conceptual understanding rather than the arithmetic, but it also provided the opportunity to see how the mathematics can be used elsewhere. Coding once again brought joy and excitement to Izzie, something that was missing in her other mathematical experiences.

Allison also had a generally positive relationship with mathematics, but not necessarily mathematics courses. As previously noted, Allison remarked about how replication was not the norm within this experience, but rather exploration. Rather than having to do the same procedure over and over again, there was an element of creativity that provided some excitement. While she was in this study, Allison was also taking a matrix algebra course. She noted that in that course *"it was really just like you're taking the course - do as well as you need to, and move on, and don't really understand what's going on"* and because of this, she was not enjoying her course as much as she had hoped. She was thankful that she was a part of this study, as she had a better understanding of concepts and was able to engage in open-ended problem-solving. While her course was not necessarily a negative experience, the contrast in experiences allowed for her to build up her mathematical understanding in a different type of context and feel more confident when engaging with those problems. Specifically, she was able to not only do what she needed to do, but also understand it. The coding environment and the nature of the questions allowed for this open-ended and explorative nature, but the experience was in stark contrast with her other prior mathematical experiences.

Within this claim, the emphasis was students' experiences with mathematics and how by encountering the mathematics within a coding environment, students were able to leverage their confidence in computation to mediate their mathematical confidence. For example, Nate had actually used his knowledge of Python to investigate the mathematics and by engaging with the mathematics in coding, students had new opportunities and more ways to be successful. Further, students found newfound excitement for mathematics and reflected that this experience deepened their understanding of different mathematical concepts.

Claim 5: Computation Enacted Through Coding Provided the Opportunity for Students to Develop New Mathematical Habits and Strategies

As discussed in Claim 3, computation enabled potential shifts in how students viewed mathematics, specifically in that mathematics went from a single solution, single pathway approach where the goal is to follow rules and procedures to a discipline where it was possible to have multiple approaches, and students could learn through meaningful discovery and exploration. Claim 5 will place the focus on how students shifted away from the view that doing mathematics necessitates a linear progression of steps, and the new mathematical habits and problem-solving strategies that can be implemented because of coding and computation. Many of the students' prior mathematical experiences informed their view of mathematics specifically in that linearly solving a mathematics problem meant that if there was a stuck point at a step within the process, no more progress could be made. Students referenced bashing their head against the wall or becoming highly frustrated in those situations. However, computation enacted through coding was able to challenge this notion and students were able to adopt some of their problem-solving strategies from coding to the mathematical context. Some of the different approaches included modularization of the problem, iterative solution methods (where an initial base case is contrived and then expanded), and finally the ability to test their solution along the way, even if it was not fully complete. The difference between the testing described within this claim and experimentation through prediction and reflection cycles described within Claim 1 was that testing did not necessitate a prediction. In fact, sometimes students were unsure of the result but were willing to try a new approach. Through challenging the linear problem-solving progression and being able to modularize code as well as break down the problem, students were able to take a series of smaller risks within the problem-solving process – rather than feeling the weight of having a 'large' risk taken within typical mathematical contexts. Although these areas have been delineated for sake of clarity, they are in fact entwined in nature, hence why they are all under a sole claim. Consider the reflection that Jack provided about mathematics and coding:

> *In mathematics, it's almost like you get to the point where you know what to do and then I - it's a ball falling down a hill. Just once you set everything up the right way, everything just sort of falls into place. But with coding, I think of it as a lot more modular. Or maybe you'll get one part that runs properly, but then this next chunk of code breaks, and so you have to do it in little bits at a time.*

Within this excerpt, Jack spoke about his view of linearity in mathematical problem-solving, modularization of problem, and the ability to break a problem down and try different approaches. To begin with, the analogy of the ball down a hill gives rise to a linear view of the mathematical process. It evokes the imagery of a continuous path as a ball (without any external forces acting on it) will not roll to the middle of the hill, go back a little, then continue to roll down. Rather, it simply rolls down the hill, in the same way that students describe their engagement with mathematics problems as following a series of steps, and if you miss one step you cannot continue to make progress. Additionally, he described the modular nature of code. While this was not given in the context of using the code to solve a mathematics problem, it highlighted the affordances associated with code which gives rise to the potential of this being leveraged for mathematical tasks. Finally, another association with the code is the ability to do it in little bits at a time. He discusses how there are different parts within a larger problem, and he can both build and test his solution in pieces. This sets the stage for a deeper dive into these opportunities for different approaches when solving mathematics problems.

Expressed Linearity of Mathematical Problem-solving

Many students expressed a view of mathematics that was governed by a set number of steps. Olivia stated that when she is solving mathematics problems, *"you have a set number of defined steps that your professor tells you to do, or certain things to look for."* Within this view, mathematics has a rigid structure that is defined by the instructor and students are supposed to follow this progression. Therefore, it makes sense that if you miss a step, you cannot continue to engage in the problem. As shown earlier, Izzie had a similar view of mathematics because *"you've got all these steps and then you forget how to do a step and then you can't do the problem and you get really frustrated."* She described mathematics as a linear progression of steps and highlighted the frustration that she felt when she hit a stuck point. This view was also highlighted in comments such as when Colton stated his frustration with mathematics was *"having to remember all those little rules and tricks"* or when Harper discussed her mathematics course and how she had to learn the *"tricks."* Further, for her to learn it she needs repetition. Nate also emphasized this repetition and learning the steps to avoid *"bashing [his] head into a wall."* Many of these views came about when reflecting on the relationship between computation and mathematics, and specifically when students emphasized different computational and coding practices they leveraged during the experience. When describing mathematical practices that they engage in, the linearity of approaches also arose in terms of following examples. For example, Micah stated that when working on his mathematics homework: *"I base a lot of my stuff on like examples that we've given in class. So, I usually have just my notes, like literally right in front of me, and I try and find whatever is closest to that."* Multiple students expressed similar viewpoints which highlighted this sequence of steps, similar to the prior claims.

With the rules and tricks previously described many times the mathematical process was needing to agree with the mathematics instructor's method. Nate noted how math is *"a bit more procedural"* and how there is *"some sort of set way you're supposed to solve most problems."* While this does not directly indicate a linear progression within mathematics, what is solidified is that there is a certain way mathematics problem should be solved, and drawing on other students' statements this also typically indicates a certain path or number of steps to take. The way in which the subject, the students, interact with mathematics is governed by their view of the nature of mathematics and the perceived rules that are present within the structure of solving mathematics. This also poses potential problems for mathematical creativity. One of the key dimensions is risk, where students take an action where the result is unknown to progress the problem-solving process.

If the view is that mathematics must be done linearly and in a particular way then when they hit a stuck point, that piece must be resolved before progressing. For example, when Jack was describing a homework problem, he mentioned how he continued working on this complex problem up until he hit a certain point. Despite his different attempts, it was *"unsolvable"* for him. From this point he went to his instructor for help to be able to make forward progress because he said that he was unable to progress until that happened. However, within the observations the difference was how students could leverage coding and computational practices within their mathematical approaches.

Computation and coding were something that could challenge this step-by-step and linearity assumption. Consider the following excerpt from Izzie:

> *And because you're going through it a bit, like I was saying, with the maths problems, you have to go at step by step. And even though with code you don't have to do it quite step by step in the problem, the same when you are writing the code, you still have to follow that step by step. And if you don't know what the code is doing, you can't use the code to do a problem in the first place.*

Note that while she does view mathematics as a linear flow, the code is something that you are able to work on in a different order. The caveat is that when reading code, or evaluating the code, it still needs to be viewed in a sequential order. This provides the opportunity to view a problem and potentially break it down into pieces that do not need to be solved in a linear fashion. This is a key opportunity, as this is one of coding strategies that students were able to adopt and bring into their mathematical habits.

New Approaches in Student's Mathematical Habits

Throughout the different modules, students engaged in mathematical habits that differed from those that they expressed during the initial interview. It is not that every student adopted each of these practices, but rather that these were different mathematical habits that students engaged in due to the computational environment. Students engaged in modularization, where a problem was broken up into independent chunks so if a problem was encountered, it was not a dead stop to all progress as previously described. Other habits included testing new ideas within the code, even if a student did not have a prediction, as students were willing to take risks within the code. Additionally, students could test a solution along the way rather than waiting to check the final answer. These habits advanced the students' mathematical problem-solving and also may be built upon in various ways to promote mathematical creativity.

The modularization of the code was something that students could leverage when working through the modules. Specifically, they could break apart a problem into smaller subparts that were independent of one another and could then be integrated to form a solution. Jack had noted that within the computational mathematics:

> *It's easier to just work on one part at a time, individually, get those parts kind of working, and then put them all together and see if anything falls apart then or if you get a final product that you're looking for.*

In doing this, he and his group were able to break apart the problem into different portions, work on those pieces individually, and then combine at the end when they were done. This was seen across numerous observations, as one of the first steps when engaging in a more complicated problem was to start at the board and actually write the pseudo code or the general outline to a problem, an approach that was directly from their introduction to computational modeling course.

From there they would approach the problem in pieces and work together until they felt satisfied with the solution. His other groupmates also reflected on this in their final interviews as Izzie stated that,

> *Actually, you do need to be quite creative when you're doing it [math] with coding because as we've done on the whiteboard so many times, there's a lot of planning that you need to do to get in. You know can't just have the idea, you've got to plan it, pseudocode it.*

Harper and Theo also made similar comments that focused on breaking down a complicated problem on the board, parsing it into different steps, and being able to work on the different pieces. Note that once again this portion of the claim is coupled with the introduction to Claim 5 because there cannot be a modular approach to a problem that must be solved in a linear fashion.

The element of modularization is also reflected in students' ability to break down a problem and work on the problem piece by piece. For example, Alex detailed one of his computational practices during difficult or larger problems was to *"reread the question and kind of try to break it down a bit more into smaller parts."* This approach was noted across observations within mathematical contexts such as when developing matrix multiplication code and starting from the inmost calculation and working his way outwards, adding complexity as he went.

One of the affordances of coding as noted by participants was the ability to test ideas within the code. This meant trying new mathematical approaches within the code, but also something more. Students seemed more willing to take risks in their problem-solving and try novel ideas when working in code than with pen and paper mathematics. For example, consider Nate's statement about this:

> *I feel like at least with my experience, I find coding to be a bit easier because I could - I can solve a problem and then I could think of a way that I could break my code, and then*

> *I go and I try to fix that, and then I can immediately test it. I can just hit run and if it doesn't work, I have to figure out why it didn't work. I feel like the feedback is much, much faster with coding. I, I like, I like doing that, you know? I don't know, sometimes when I get, when I'm doing a math problem and I get to a point, I just feel like I'm bashing my head into a wall. So, I just can't quite seem to figure out what I'm doing wrong and then, yeah. and sometimes it's really hard to, to figure out what you did wrong, and you can look it up and you may not find out what you're doing, which you, what you've been doing.*

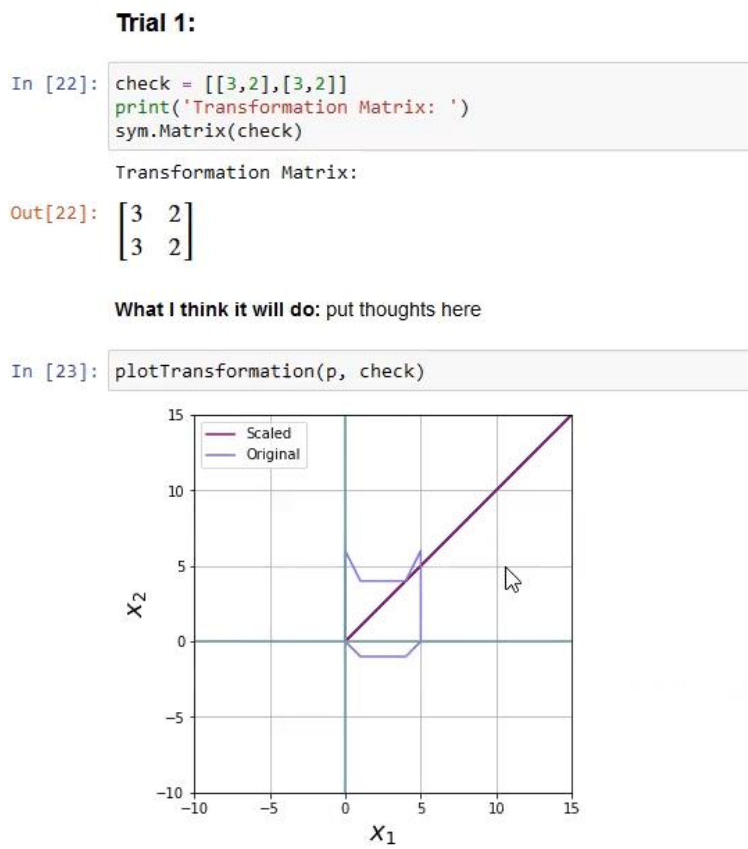
Within this excerpt, Nate specifically highlighted how students were able to get instant feedback from coding. This was due to potential error messages that they receive if some syntax failed or there was a logical error. This is compared with mathematics where he hits a point in the process and cannot advance. Note that this is a different cycle than what was described earlier because it is not mathematical prediction and reflection that allowed for the risk taking but rather is operationalized through the feedback mechanisms built into coding environments. These helped orient Nate to his potential errors and note that he stated, *"I can just hit run."* There is a lower cost implied because as Nate said, it is simply a push of the button where the code will execute and provide feedback. Of course, this feedback will not necessarily indicate whether the mathematical assumptions and problem setup are correct, only whether the code was able to fully execute. However, as seen by many groups when they encountered errors in the code, there was often an underlying mathematical error.

The ability to test within the computational environment, even if not coupled with an explicit prediction, was a key part for some students engaging and learning the mathematics. This is in contrast with Claim 1, where prediction and reflection cycles were one of the structures in which experimentation occurred, especially during the matrix transformation module where students explored how different matrices altered an image of a cat. For example, consider the following excerpt from Micah's final interview when he was asked what helped him push his mathematical thinking:

> *Micah:* Uh, I mean, it's, it was also sort of like the examples helped push it a lot or like just like, um, whatever, when it came to like the cat images or like transforming things or changing those matrices. It was always just like looking at what happened whenever I changed the matrix and then always thinking back on why it happened almost. So, like I'd, I'd always try and like, just play around with matrices until finding something that worked.
> *Sarah:* Mm-hmm. So, how did you go from kind of like the element of like playing around with them to figuring out oh, okay. It's because I had something here that the cat image did this, or now it actually looks like a cat or something like that.
> *Micah:* Yeah. Um, I dunno, it was sort of just like, oh, well, if you change like the value from like, you know, zero to like one and then one to negative one or something, just whatever that did to the image. Just like, I feel like I just like broke it down to the steps about like, okay, well, if I change the number here, what does it do? Like the overall equation. And so, you know, I had to do like several trials in like seven or eight to find out like, or what each value in the matrix changed and like a graph or something mm-hmm. And so, it's like, it came down to a lot of trial and error when I was like first try to figure out matrices and solve them because it's just like, all right. I just gotta like guess a little bit about what's going on and then eventually I'll put it together. So usually, I could piece it together towards the end, but like, it was always just learning matrices. I just gotta like brute force it for a little bit. So hopefully works.

Within Micah's experience, he described a similar pattern to other students when engaging in the experimentation through prediction and reflection; however, he used the terminology of trial and error. This is because during the observation he had started with an initial trial where he was testing a matrix but did not record a prediction and then continued through the cycle of testing different matrices within the same code cell. When reflecting, he noted the ease with which he could change a number and see the result. In doing this, he could test his different ideas – even if there was not an explicit prediction component. The coding element with visualization allowed him to employ a *"brute force trial and error"* approach. This would not have been doable on pencil and paper, as he was able to perform numerous different trials, thereby highlighting how the coding environment supported this method of engaging in the mathematics. He was able to build on this method and engage in reflection for abstraction, and when he was asked why a certain matrix produced a line, as shown in Figure 4-14, he stated, *"I think if I had both the values here (moving mouse across each row of the matrix) it's just like expanding in the same direction. It's the same thing on both sides."* This was not a part of an explicit prediction and reflection cycle, rather it is the result of trial and error, and looking for changes in the system based on changing the inputs. It highlights how Micah was able to test in a computational environment and try things that were unknown at the time. Then he reflected to bring about a mathematical understanding or advance his problem-solving process.

Figure 4-14: Micah's trial where the cat is projected into a line. He reflected on the output, along with prior trials, to determine the mathematics of why this occurred.



Across multiple students there were shifts in mathematical habits in comparison to the habits and world-images of mathematics with which they entered. This is not to say that all prior mathematical habits were abandoned, but rather, there were simply more tools and strategies for students to engage with. As discussed, students now had ways to break a problem down into different parts and therefore not experience the same level of frustration with some of the sticking points within a linear interpretation of mathematical problem-solving.

LIMITATIONS OF COMPUTATION ENACTED THROUGH CODING

The previous claims have highlighted the potential pedagogical power of computing, enacted through coding in a creative environment. However, it is important to highlight some of the limitations that computing has with respect to mathematical creativity. First of all, as previously discussed Izzie reflected,

> *And because you're going through it a bit, like I was saying, with the maths problems, you have to go at step by step. And even though with code you don't have to do it quite step by step in the problem, the same when you are writing the code, you still have to follow that step by step. And if you don't know what the code is doing, you can't use the code to do a problem in the first place.*

Izzie highlights one of the constraints of when students are asked to engage in computation and are given prewritten code. If a student does not understand the code, then they are unable to use the code. This can lead to the aforementioned ‘pushing buttons.’ Further, Izzie highlighted that when solving a problem, you do not have to go step by step, when writing the code, you do have to. The code needs to be written in such a way that a computer can follow the instructions. While there is the possibility of returning a dummy variable or another argument to proceed in the code writing, when working with computation enacted through coding, the computer still follows a serial set of commands. Therefore, there are still limitations about understanding the code and potential linear code writing methods.

Further, computation on its own does not guarantee opportunities for flexibility within mathematical creativity. For example, as Olivia worked by herself to complete the modules, she reflected,

> *So it wasn't, there wasn't a lot of people for me to bounce off ideas from, and I couldn't really hear a lot of different ideas either. And along with that, this is all stuff that I haven't seen before, so trying to apply stuff that I haven't seen before, but I'm trying to understand and then apply it to code, but not really knowing what it should look like. That kind of was a little difficult.*

Within this quote, despite engaging in the computation, Olivia did not have other group members to develop alternate solution pathways. Therefore, even though coding offers multiple ways to code a solution, she did not have the same opportunity as others to see different problem solving strategies. Further, Olivia exposed one of the most critical limitations of computing enacted through coding. When working with mathematics and computational processes enacted through coding, there are multiple key portions of the problem-solving process. It is important to understand the mathematics, the coding syntax, and how to take the mathematics and turn it into code. Oliva noted that she did not know what it *"should look like"* and this is a difficulty. Olivia reported having overall meaningful experiences within this study, but within this quote she highlighted a significant limitation of computation as a pedagogy because there were so many parts to the problem that she had not seen before. Within a classroom context, this could be overwhelming for students and also placed a high cognitive demand on students. While these were not a complete list of the limitations of computing, it is important to understand the nuanced complexities of introducing computing for mathematical creativity. In light of these claims and limitations that arose in the data, it is important to revisit the guiding research question.

ANSWERING THE RESEARCH QUESTION

The findings and claims that resulted from the analysis have been thoroughly expounded upon in the prior sections of this chapter. This next subsection will use the prior claims to answer the guiding research question presented within Chapter 1 and will proceed in the following format. The research question will be presented, followed by a detailed answer based upon the prior claims and supplemented by additional participant reflections.

> *How can computation enacted through coding provide opportunities for students to develop and express their mathematical creativity specifically in the context of learning linear algebra?*

As indicated in the claims, there are multiple ways in which computation enacted through coding can provide opportunities for students' mathematical creativity. The first of which is through engaging students in experimentation through prediction and reflection cycles. As discussed in Claim 1, these cycles encouraged students to ask "what if" questions and to engage within experimentation. Students were able to follow their intuition and curiosities. These cycles thereby support the development of students' originality, risk, and elaboration. Students may have had a prediction, but it was not guaranteed that the prediction was correct. Further, when they made suggestions of potential solutions or ideas to try, they were taking a social risk. However, by having students decide how to approach some of these experimentation pieces and then having them code their solutions, it allowed for the development of meaningful connections between mathematical ideas, such as the matrix transformations. By using code, the students were the ones in charge. It was not that they were just copying down notes, something that many of them associated with mathematics. Rather, they were actively trying different ideas and the process of coding empowered them. For example, Nate felt the code *"helped [him] get a better understanding of it [the mathematical concept]"* and Olivia noted that part of the reason why she enjoyed this experience was the modules did not just focus on content coverage. Rather coding *"lets you learn on your own, experiment with stuff instead of just being told like, oh, this is a rule in mathematics, and you should just follow it and we're not really going to talk about it."* For Olivia, the actual computational environment was what enabled this experimentation and so this is one of the key affordances for mathematical creativity. As noted by other students, this is partially due to the ease in which you can run an experiment. For example, Jack brought up:

> *And so, I think it helped with it being iterable where we could just try this idea and this idea and this idea instead of having to take the five minutes in between to erase everything, write it out again, do all the steps.*

Having the code execute the steps rather than having students erase and do everything by hand, which allowed students to focus on the conceptual pieces and pattern recognition which ties directly into elaboration. The focus could be on establishing meaningful connections rather than solely having to do arithmetic over and over.

The prediction and reflection cycles also supported the development of multiple representations. As visualization (both physical and mental) is a key element of mathematical creativity, this affordance is extremely important and one of the elements that students continually reflected on during their final interview. For many groups, the determinant module had them continually bridging and connecting their code with the output produced. The modification of the code led to the determination of multiple cases of linear (in)dependence where they then pivoted to make connections between the concepts they had already encountered and the code output. This enabled for a natural bridging between the graphical, algebraic, and numerical understandings of a determinant. This is one of the particular strengths of these specific modules but more broadly within computation enacted through coding. Many programming languages have supports built in for visualization which allows students to plot different objects that were previously unattainable by hand. Further, the act of coding coupled with students' control over the representation enable not only originality but also constant maintenance of multiple representations.

The physical visualization capabilities of coding were another affordance that is highly beneficial for mathematical creativity, as detailed in Claim 2. Even students such as David who had already taken a linear algebra course remarked on how helpful and intuitive this was, and something that was missing from their courses. Harper noted how the visualization was really helpful for her, and the examples she gave were linked to different prediction and reflection cycles, such as the following quote:

> *I think I really liked when you included some sort of visual where we could graph it and see what happens. So, you could actually see what putting in, inputting different values. The Roomba one, I think it was on the last one, how you could input different values and it would end up at different spots and stuff like that. So definitely visualizing is better for me.*

Visualization was not limited to just plots but also the way in which intermediate steps can be output within the code, or additional displays can be added. As Nate noted when asked why he felt Python was beneficial for his understanding:

> *Being able to visualize stuff, honestly. Cause being able to, I guess take two vectors and then make some nested for loop to show that it actually can reach basically every point in* $\mathbb{R}^2$ *or* $\mathbb{R}^n$*. Then graph it and see it visually is really interesting.*

Within this quote, Nate is highlighting both the power of being able to graph and develop plots of ideas is helpful, but there is even more. The visualization that the code has and the outputs that are made is something that is unique to the coding process, and yet students found highly beneficial.

The outputting of different parts of the problem allowed for visualization, but coding and computation also enabled new mathematical habits and strategies, as discussed in Claim 5. Many students viewed mathematics as a linear path with a fixed number of steps given by the instructor. On the contrary, coding was something that although it was executable in a sequential order, it was able to be modularized. Students could develop different parts of code and then test them individually and reassemble. Further, during this process coding gave quick feedback through error messages. This enabled students to take a series of smaller risks along the way and then test their code as they created it. This supported students' originality in being willing to follow their ideas, but it also developed some of their mathematical risk taking because it minimized some of the arithmetic work while also allowing for quick inquiry. Once again, the coding environment supported the development of mathematical creativity.

Although many of the structural pieces of coding and computation allowed for different engagement patterns, there was also a relational component to the ways in which computation can bolster mathematical creativity. This relational component was at the core of Claim 3 and Claim 4. Many students had prior mathematical experiences that left them questioning their mathematical capabilities. The computation was a new environment where they could have their confidence in their computational abilities actually mediate their mathematical self-view. For example, both Kylie and Ivy developed a resilience in their mathematical problem-solving due in part to their computational world views. Both articulated a notion of debugging within coding and how the problem-solving process rarely goes perfectly and smoothly. They were able to use these views to then continue on with the mathematics, and they arrived at a point where they were excited to continue and even thought of ways, they could extend what they learned to new domains of mathematics. This is a crucial affordance of computation for mathematical creativity. If a student has a view of mathematics needing to be perfect or in a linear fashion, like the discussion in Claim 5, then it will be highly difficult to take risks or engage in originality. Further, if the student finds the math intimidating, then it is easier to disengage to protect themselves, or follow the path for that specific problem. This then stunts elaboration and other elements of mathematical creativity. Therefore, the norms associated with their coding experiences, and it being a new environment to encounter mathematics, allow for a greater chance of fostering mathematical creativity.

Students' relationship with the discipline was also mediated by coding, as the computational experience had the potential to shift student's mathematical views. Mathematics was no longer perceived as being a single solution, single pathway discipline composed of rules and procedures for students; rather, the process of doing mathematics was opened to exploration, multiple methods, and meaningful discovery. There was a marked shift from doing mathematics by following a specific set of steps to centering mathematics on exploration. Once again, this was mediated by the norms associated with computing, but also the inferred freedom that computation brought. Students reflected on how coding enabled the testing of ideas, and this allowed them to be in control and think through all the possibilities. Differences in coding styles and ideas allowed multiple solution processes that were simultaneously correct. The coding allowed students to take a more active role within their learning, but more specifically the code highlighted the ways in which they could conceptualize a problem and develop an approach. When reflecting on the modules, David stated, *I think it requires more creativity to actually write a code that's going to matrix multiplication. I think it's a lot easier to be given two matrix matrices and multiply them than it is to write a code that can multiply any two matrices it's given, which is what we did.*

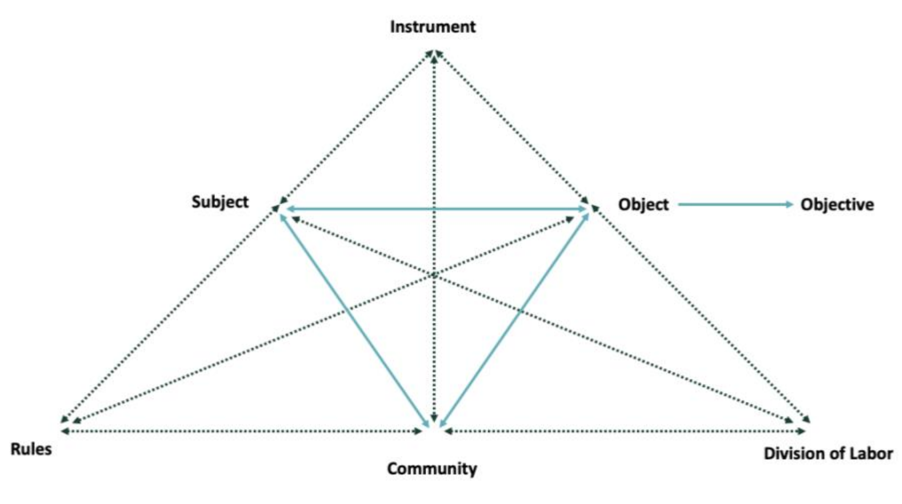
When writing the code for these mathematical concepts, there were multiple ways to approach a problem, and David's statement highlights how the abstraction that is necessary in coding actually requires students to be more creative. Across the different claims, student voices have expressed similar conceptions.

It is important to note that these claims are in the context of computation enacted through coding, and specifically within a creative environment. Nonetheless, together these claims and student voices highlight the potential that computation has for mathematics education. Computation enacted through coding can provide structural mechanisms such as the prediction and reflection cycles that scaffold mathematical creativity while also allowing for new social rules to mediate students' relations with the disciplines.

CHAPTER 5 - DISCUSSION, IMPLICATIONS, AND CONCLUSIONS

The work presented within this study seeks to understand how computation can be integrated into the undergraduate mathematics classroom, and how this has the potential to develop opportunities to foster mathematical creativity. This chapter is focused on developing further nuanced understandings of the claims presented within the results section and highlighting the potential implications for research and pedagogy. The chapter will proceed as follows. Initially, each of the mediating factors from the CHAT framework, including the instrument, sociocultural practices and rules, and the division of labor as seen in Figure 5-1, will be discussed in light of the prior claims and situated within the literature. I will then argue for the multiple research, methodological, and pedagogical implications of this work. This is both for the mathematics education and computational education fields and includes the challenging of deficit narratives, answering the call for the exploration of the relationship between coding and mathematical learning, and novel ways for students to engage in linear algebra that promotes an intuitive view. The chapter will conclude with the limitations of the study and the future directions.

Figure 5-1: Cultural-Historical Activity Theory (CHAT) framework where the solid teal lines indicate a direct interaction and dotted lines represent the mediated interactions.



DISCUSSION

As the CHAT framework provides a nuanced and detailed understanding of the complex inner workings of the activity system of interest, this section will first begin with a discussion of how the computational modules mediated students' mathematical understanding. Specifically, how the prediction and reflection cycles, coupled with the multiple representations, fit into the larger mathematics education discussion. Following this, the mediation of the sociocultural rules within the study will be discussed in terms of student's mathematical world-image, self-image, and mathematical habits, and specifically how these build on and provide new avenues for much of the inquiry-based learning projects. Finally, I will present a discussion of how the division of labor mediated students' mathematical understandings and actions where, in light of the results, I argue that coding is not enough to guarantee creativity, a call that the mathematics and computational education community needs to take seriously when considering the broader scope of this work.

Mediation of the Instrument

Engaging in coding enabled students to experiment with mathematics and create multiple visualizations of concepts, which provided unique opportunities for advancing mathematical creativity through originality and risk-taking. This in turn advances our understanding of the relationship between coding and mathematical creativity, a connection previously unexplored in the field of mathematics education.

By using coding, students were able to engage in originality and try new approaches to their mathematical problems. During exercises such as working with linear transformations, students used the provided code to explore mathematical patterns thereby providing the basis for future explorations. Students also used the computational notebook to insert a new cell and expand an existing piece of code or develop new code to pursue their inquiry. The prediction and reflection cycles in Claim 1 highlight the ways in which students experimented and tried different approaches with mathematical concepts to advance their problem-solving. A common question that arises within literature and practice is whether coding is necessary for mathematical learning, or whether an applet would be enough to produce the same opportunities (Grover et al., 2019; Wassie & Zergaw, 2019). Within this study, this develops into the question of whether interactive graphics could provide the same opportunities for students to foster mathematical creativity. Although researchers have found that students can engage in experimentation through provided interactive modules, only through the coding do students have full freedom with originality (Liang & Sedig, 2009; McDonald & Stewart, 2023). When engaging with the applets and prefabricated tools, students were confined to engaging with the applet in predetermined ways or were only able to modify a subset of features. This thereby constrains how the student can use this tool and what potential ideas they can explore. In comparison, this study highlights how, by engaging students in coding, students were able to follow their own line of thinking and either appropriate the tool or develop a new tool in the context of coding to explore the mathematical object of interest.

Students cannot engage in complete originality unless they have full access to the inner workings of the program and can expand upon any direction that they would like to pursue. Students within the study highlighted how the transparent nature of coding aided in developing their mathematical understanding and the ability to follow their own ideas. Specifically, they could read through the provided code if they wanted to dig into a particular concept or could modify and adapt the code for new situations. The difference between the impacts of given code and developing code is exemplified in Harper's reflection, where she stated,

> *I think being able to do it myself definitely helped me understand the math a bit better because I had to figure it out and get it to work in my brain. I guess figuring it out on my own was helped me understand it more than when we were given it and had to read through it… it [coding] was a bit broader [than math] and you could pull anything. How I connected Matrix multiplication, and the dot product were very similar and things like that. I definitely think it led to more exploration.*

Within this quote, Harper highlighted how she learned more when she was able to develop her own code. Further, within the coding she drew from multiple disciplines, experiences, or different coding tools. She attributed this feature of coding to developing opportunities for exploration. Once again, the coding environment and tool created conditions of freedom in how the students could use the code because they were able to add any sort of modifications or create a new cell and develop their mathematical hypothesis. When working in other environments students were often limited by the ways in which the designer anticipated students interacting with the tool, thereby constraining their creativity (Resnick, 2013). In coding, students are able to follow their own ideas and interests, a key connection to mathematical creativity.

Students used the computational module to test their predictions rapidly, thereby seeking patterns and connecting between different mathematical concepts. This is exemplified both within Claim 1 and Claim 2. The ability to test predictions, or abstract different matrix transformations has been noted within prior research on Inquiry-Oriented Linear Algebra (IOLA) materials (Andrews-Larson et al., 2017; Wawro et al., 2019). However, the code that students developed during this study provided a much quicker response time, in comparison to students having to manually sketch out matrix transformations. Coding enabled students to test their ideas much more quickly. Further, as the cost to try different matrices was minimized, students were able to try different matrices quickly and repeat the process in order to derive connections and abstractions. Further, as noted within the observations, many students began to alter the code to either personalize the plotting routine or pursue different mathematical ideas such as affine transformations. The students used the scaffolding to expand out their ideas and pursue new approaches with regards to originality, and take risks more easily, thereby highlighting the ways in which the tool mediated their engagement with the mathematical and computational ideas.

By using coding, students needed to engage with multiple representations while they coordinated both the lines of code and the output of the code itself, since their computational idea or abstraction was represented in both forms, as detailed in Claim 2. Using tools gave students the opportunity to coordinate among multiple mathematical representations. While prior work noted the potential area for students to see the different representations as disjointed and existing as separate entities (Castle, 2021; McDonald & Stewart, 2023), this study highlighted how when the students wrote code, or had the code open to view, and used the coding platform, the students were able to see across the multiple representations and develop a coordinated view. Students increased their mathematical flexibility through encounters with concepts in multiple forms and representations, which allowed them to approach problems in a variety of ways. Further, as the outputs could include numerical, algebraic, and graphical output, there was opportunity for visualization. However, this goes further. When students engage in the coding process and develop their own visual representations, they create personal conceptual visualizations, as with the example of the determinant in Claim 2. Students' personal mental conceptions transform the way in which they use the tool and relate the tool to the mathematics. This work highlights the ways in which computation enacted through coding can mediate students' mathematical understandings and provide novel ways for students to engage with the concepts while simultaneously promoting mathematical creativity - specifically in the ways that coding may provide more opportunities for developing creative environments when compared to traditional applet-based applications while simultaneously promoting connections across mathematical representations. This work provides the forefront of developing more nuanced affordances of the computational environment, allowing for the development of richer research and pedagogy.

Mediation of Sociocultural Rules & Practices

As noted within the theoretical framing, the sociocultural rules within CHAT are quite broad, and incorporating the additional framework provided additional refinement to the understandings of some of the implicit and explicit rules surrounding how the students interacted with their communities and the mathematics. Once again, mathematical world-image refers to how students conceptualize mathematics, its role in the world, and what it means to do mathematics. Mathematical self-image denotes how students relate to mathematics, their believed self-efficacy, and their mathematical identity. Finally, mathematical habits are the habits that students engage in based upon their beliefs about the mathematical world-image. These dimensions govern the perceived rules that regulate how the student engages in these mathematical modules and with the linear algebra concepts.

Mathematical World-Image

Engaging in mathematics enacted through coding changed student perspectives about the nature of mathematics, specifically shifting the view from a single-solution, single-pathway discipline to one that involved exploration and meaningful discovery, a shift that is sought after in literature (Boaler, 2018; Selbach-Allen et al., 2020). This shift thereby facilitated greater ease to students engaging in flexibility, risk, and originality, something not captured within the literature on computation and creativity. By challenging the assumed norms about what mathematics is, and what it entails, the use of coding enables students to have multiple opportunities to foster mathematical creativity rather than restricting students where they see mathematics as a boring and dead subject (Boaler, 2018). As detailed in Claim 3, students such as Harper entered into the study with a perception that mathematics as a discipline centered on getting the formula, plugging in the 'right' numbers and following the previously established steps. This view of mathematics and the ways in which students had to engage in their previous mathematics as a result restricted student originality, and there was no true risk in approach as students had already seen all the approaches thereby hindering mathematical creativity. The formula was given by the instructor, the goal was to find new numbers, and then the student was supposed to simply plug in the numbers. This left no room for the student to try new approaches towards a problem, and specifically no incentive to do this. The coding experience countered this, as many students commented about the creativity in choosing how to solve a problem. Harper noted that she was able to bridge knowledge from different courses and try to incorporate what she has seen in new and innovative ways. Further, when students described mathematics as a single pathway, it stymied any potential mathematical flexibility. The sole-solution view of mathematics problem-solving blocks flexibility because flexibility inherently requires multiple solutions. As students noted, coding challenged their sole-solution views, because it opened multiple pathways through choice in variable names, coding approaches, and solution steps. Although variable names may seem trivial in comparison to the other choices, having this freedom was something that students repeatedly emphasized and allowed them to be in control. While the overall approach could have been similar, or utilized the same mathematical phenomenon, the implementation of their solution path would almost always be unique. When students acknowledge this difference, then there is deviation from the perceived universal answer that students may search for. In turn, students began to acknowledge multiple solution paths, and this was reified within Claim 2, as students developed multiple representations of mathematical concepts, thereby allowing students to develop approaches and solutions along with different representations. These modules leveraged the perceived freedom that coding affords (Isomottonen et al., 2020) to change the norms surrounding what mathematics is, thereby developing new opportunities for risk, flexibility, and originality.

At the start of this study, some students demonstrated the viewpoint that mathematical creativity is for a gifted and select few and that mathematical creativity can only arise within the context of mathematical competence. This view persisted through the study and was still implicitly embedded within some students' conceptualizations. A subset of students openly discussed the need for mathematical knowledge in order to be able to engage in mathematical creativity. They held the belief that to be creative in mathematics, you must have a baseline competency established or already have mastery of the subject. For example, Nate detailed an account of this during his final interview and stated:

> *I think it [mathematical creativity] all comes down to experience, honestly. The more you know about something, the more creative you can be with it. In my opinion. I would say that math is a little bit more procedural. You may look at a problem and there may be two, there actually could be a lot of solutions, but you're not going to know about 'em until you've learned about some new concept or whatever.*

Nate highlighted how when students consider mathematics to be procedural, then their creativity is constrained until they have achieved mastery. He stated that to be creative, you have to first learn the concept before engaging in this creativity, a sentiment that echoed the origins of Poincare (1906) and others who advocated for mathematical creativity being for a select few. There is a notion within mathematics, that in order to explore in mathematics, you must be good at it. However, as noted within the results section, students did not have the same concerns surrounding coding. This sentiment and view were echoed by Jack:

> *I think at least with the way that I work on problems, I tend to notice more creativity in computation because I just don't know enough mathematical stuff to be confident in my ability to try a different method and have it be correct. Especially when you get into the more specialized forms of mathematics. I normally just the one method to solve this type of problem, I do it because it works.*

Jack expressed his concern over trying new methods and being correct. While this can relate to Jack's mathematical self-image, it is also important for understanding his mathematical world-view. In this way, there is an implicit assumption of needing to be correct when trying something. This may in part be due to the hidden messiness of mathematics that students experience during lecture and within textbooks in comparison to coding where debugging is part of the norm. Within mathematics, there is a notion that creativity needs to be preceded by mathematical knowledge and ability. Jack expanded upon this within his initial interview by stating,

> *[Math] solutions are things that, you know, someone very smart came up with and unless you're like, as smart as they are, you're probably not gonna come up with some new solution to it. So, they're very much, okay, this kind of problem is solved by this method*

and it's kinda like a lock and key thing where you have to find the, the thing that matches up with the, the question that they gave you. Not really a whole lot of room to innovate.

The very nature of the mathematical world-image of having a lock and key highlights the restriction of creativity. If an instructor gives a student a procedure, then there is no incentive to deviate because they have already discovered the key that works. Further, to do anything with regards to mathematics, he noted that you must be smart to develop something novel or original. Therefore, there is an implied notion that to excel in mathematics or be creative, it is for a sole subset of students. In David's initial interview he reflected,

I feel like you need the computational stuff down, which is why I think it's important that calc is a little more computation based before we get into the proof-based stuff in linear algebra. Because I feel like if you don't, if you're kind of shaky on your computation, you shouldn't really be getting into like the creative part of math quite yet. I feel like you need to have that down before you venture into the creativity.

Note that in this excerpt, David uses the word computation not in the same way that I have been using it, but rather to denote the ability to follow a procedure or arithmetic. Therefore, he believed that students cannot even engage in creativity unless they have the procedural portions down. Once again participants demonstrated an assumption of mathematical competency as a necessary precursor mathematical creativity. This view aligns with the 'genius view' of mathematical creativity (Helson, 1983) and the inherent and dangerous assumption that there is a single best solution path within mathematics which in turn limits student thought (Riling, 2020). However, as noted within Claim 3 and Claim 4, by the end of the study the students' world-image of mathematics allowed for some openness and creativity, because of the coding environment.

Therefore, this work challenges the genius view of mathematical creativity through students' use of code and aids students in viewing it as a process rather than an attribute, in line with similar pushes within research (Karakok et al., 2018; Savic, 2016; Savic et al., 2017; Sriraman, 2009; Yaftian, 2015).

An interesting observation is that students who expressed these views were all male. I am not trying to make an argument for whether or not this occurs across all populations or whether this is an artifact from the data for consideration. Conversations enforcing mastery before creativity within mathematics did not arise within the interviews and reflections with the women in this study. As the mathematical world-image is situated within sociocultural practices, this observation questions how the introduction of the new tool thereby influences the rules surrounding students within the activity system and how this in turn impacts the objective of engaging in mathematical creativity. This observation raises questions and potential future avenues to explore both in light of research and pedagogy. Current work probes how teaching for mathematical creativity can enhance equity in the classroom (Luria et al., 2017) and this study contributes how computation can be a tool used to assist this process, specifically challenging the reified notion of mathematics being a single solution discipline for a select few.

Mathematical Self-Image

Coding and computation changed the rules of what it meant to be good at mathematics, thereby countering students' negative mathematical experiences and bolstering confidence in their abilities. This is intricately tied with the changes in students' mathematical world-images as previously detailed. As student's view of the nature of mathematics within the computational modules changed, this provoked an affective shift in students because what it meant to be 'good at math' also broadened. As detailed in Claim 4, students became frustrated when they got the '*wrong'* answer or when there was a focus on the speed of being able to solve a mathematics problem quickly. These frustrations and findings are consistent with literature (Dunleavy, 2018), as there is a dominant narrative that mathematics values the quickest students with the right answer. However, this work is novel due to how student's computational confidence mediated their mathematical confidence. For students such as Ivy, Kylie, and Nate, their computational self-image focused on their ability to be able to problem solve and figure out how to implement their ideas into coding. During the activities, they drew on this view to learn the linear algebra concepts, and Nate detailed how he was able to leverage his knowledge of coding to feel more confident and do more. By the end of the study, these students noted how they felt stronger in mathematics, less intimidated, and able to expand their mathematical knowledge and try new things. Inquiry-based learning (IBL) has documented some of the positive affective changes in students, such as increased confidence (Laursen et al., 2014), and this is in part due to changing the nature of what it means to do and learn mathematics, shifting the student's mathematical world-view. However, this study highlights how, by engaging students in activities that mirror tenants of IBL, the actual coding itself and students' computational self-image mediated students' mathematical self-image. There is very little work in the mathematics education literature exploring this phenomenon or how computing can bolster self-confidence. Many student-centered approaches to learning have resulted in documented affective shifts (Boaler & Staples, 2008). However, this study focuses on how the introduction of computation and coding shifts students' affective relationship with mathematics, specifically due to coding. As noted, this was especially evident within cases such as Ivy or Kylie, where their confidence in their coding abilities mediated their confidence in mathematics. This study demonstrates the potential that computation enacted in a creative environment through coding has, especially for those who have prior negative mathematical experiences. Considering the ubiquitous nature of computing within STEM concurring with that of mathematics anxiety, this specifically points to how the use of computing can aid in students challenging negative affective beliefs about themselves in mathematics. However, this work does not address the mathematical self-image for students who have technology or coding anxiety, and more work is needed to ensure that computation does not become another source of mathematics trauma. Nonetheless, there is a missing corpus of literature surrounding coding mediating students' relationships to mathematics and none that explore it within a creative environment. This work specifically emphasizes the need for a creative environment to development the space for these affective shifts to occur and highlights the potential affordances that coding has for the mathematics education community.

Mathematical Habits

The introduction of learning mathematics through coding and computation enabled students to develop new habits to engage with mathematics and problem-solving, including modularization and the intermediate testing of ideas and solutions. The transformation of students' mathematical habits observed in this study developed directly from the shift in self- and world-views discussed above. As students saw mathematics as no longer a single solution, single pathway discipline, students like Izzie could then engage with the code in a way that aligned with her current conception of an idea. This is in contrast to following the step-by-step procedure that prior instructors gave her. Again, students perceive rules for how to engage in the mathematical habits, such as Nate's articulation of the need to repeat multiple mathematical procedures to learn because mathematics is step by step. This in turn means that students' mathematical creativity would be restricted because the habit would center on repetition and the need to repeat an identical procedure rather than exploring how the mathematical algorithm could be applied to novel problems. These mathematical habits are based upon the ways in which students view the nature of mathematics and how they view themselves in relation to the mathematics. As a result of the profound shift in students' mathematical beliefs through their participation in this study, their actions and habits also shifted in ways that demonstrated increased creativity and more positive affect. This shift in habits and actions were observed throughout the observations, and students' actions corresponded to the expressed change in their views during the final reflection and interview. As there were demonstrated and sustained shifts in belief and action, then there is potential for the longevity of the view of creativity within mathematics. Of course, this is dependent upon future mathematical endeavors and the sociocultural rules governing those interactions. However, this provides a historical perspective where students established a more positive affect with mathematics thereby allowing for creative mathematical habits.

Mediation of Division of Labor

One of the most important elements of discussion is analyzing the division of labor within the study, as the importance of groups becomes evident and highlights how coding is not enough to guarantee mathematical creativity. On its own, computation is not *necessarily* enough to support creativity. The division of labor is a direct mediation between the student and the mathematical material. This means that when considering the ways in which students engaged with mathematical creativity, coding and computation are not enough to guarantee the fostering of mathematical creativity. This was further reified within this study, as one of the most important elements, as identified by students, was their group. Specifically, collaboration was a crucial part of the student experience, and the opportunities for mathematical creativity through coding otherwise would not been realized had this study focused solely on the experiences of individual students without any relational framing or understanding. This sentiment is reflected and wonderfully summarized within Jack's final interview. When asked if there was anything else that he would like to add, he reflected:

> *I think I've talked about pretty much everything conceptually at one point or another that I thought was important. I did want to say, I think the group aspect of it is an important part that I don't think it can be understated that this would've been a lot more I don't know, frustrating, irritating, boring, annoying, or some negative word to do alone and just try and stare at a computer screen and just mess around with stuff that you don't really know what you're doing. And having people to just bounce ideas off of or think out loud to, or just even not be entirely serious all the time and break the ice a little bit. It was important for the learning experience in a way that I don't think necessarily gets conveyed just through the results of the actual modules themselves.*

Jack highlighted the different supports that his group provided and explicitly pointed out how crucial these were for the experience. The work was not done where each student focused on their module. Rather, the way in which the group accomplished the task implored discussing different ideas, which enabled flexibility, and being willing to take risks and investigating pieces that were unknown, enabling risk. The group allowed for multiple views and perspectives when trying to learn the material which enabled different ideas to come to fruition. Further, Jack predicted it would most likely have been a negative experience if it was just him, the module, and the code.

Jack identified that the if the sole consideration is the modules themselves, or only looking at the computation within the modules, then a key portion of the learning experience would be missing. This is one of the reasons why this work cannot be divorced from the context in which it was enacted. To simply implement the modules as individual explorations would deny students the opportunity to learn from each other as well as could mitigate some of the ways in which computation mediated mathematical creativity. Further, the language shift that Jack highlights was important. When working alone he stated that he would *"mess around with stuff that you don't really know what you're doing."* His description of working alone was no longer focused on experimenting, trying new ideas, or exploring, all phrases that were used at one point or another to describe his experience with the computational modules, but rather focused on simply messing with stuff he didn't know. For the mathematical creativity to be fostered, the context in which the computation was enacted was critical for Jack. He even stated that *"it lets you be creative in a way that's not just throwing code at a wall and seeing what sticks."* This is one of the key points of this experience is that students were able to engage in responsible risk and originality when supported by their group.

Throughout the final interviews, nearly all students discussed the importance of groups within the computational context, and the specific roles that the group plays within their learning experience and enactment of creativity. The groups served multiple roles for students, including generating alternate ideas and being a sounding board for potential explorations, and it also provided students with a sense of community that oftentimes was missing from their prior mathematical experiences, as detailed in prior claims. During Harper's final interview, she repeatedly referenced her group. The importance of her group was reflected in her statements such as:

> *But it was super fun to be able to work together and collaborate and then actually have it work out and see things working and then being able to help others as well if they were confused. And get help from others if I was confused… I definitely enjoyed, again, the group aspect of it and being able to have, I think having the whiteboard and area where we can work together was very nice too. It gave us different options to try and collaborate either through code or through the whiteboard or just talking out loud and everything.*

Within Harper's experience, her group served many purposes. Her group was able to support her if she hit a stuck point, and she was able to do the same for them. Further, the way in which she engaged was not through coding alone, as the whiteboard and different collaboration options allowed new collaborative opportunities to arise. While the group was still engaged in the computational process, having the mediums outside of code was critical for their problem-solving and developing flexibility. The computation enacted through coding would not be enough to guarantee the development of flexibility and originality in Harper's case.

Flexibility and the importance of multiple methods was a common theme across participants. Allison specifically noted in her final interview that:

> *I liked the whole working with other people. It was nice to go in and you weren't doing this whole thing by yourself and that you had other people, or if you got stuck, you could ask someone and be like, hey, what did you do for this? Or just to get a different idea of the different ways you can do things.*

She specifically accredited developing multiple solution methods to the group discussions and being able to work with others; this was echoed by students including Izzie, Theo, Ivy, Kylie, Alex, Versha, Lee, and Nate. As previously mentioned, the computation and coding enabled multiple solutions due to the customizable nature of code and how there can be multiple solution paths. However, for Allison what spurred her mathematical flexibility was being able to work with others. Therefore, the coding and computation were not enough for the mathematical creativity.

During the study, Olivia was by herself in a group since her groupmates had either dropped out of the study or needed to switch groups due to scheduling conflicts. Working through the modules on her own forced the burden of mathematical creativity to fall on her shoulders alone. She stated,

> *I think the most challenging part was trying to take some of the information written in text and then applying it to solve problems in code. And I think that might have just been also because I didn't have a group which is fine. Yeah. So it wasn't, there wasn't a lot of people for me to bounce off ideas from, and I couldn't really hear a lot of different ideas either. And along with that, this is all stuff that I haven't seen before, so trying to apply stuff that I haven't seen before, but I'm trying to understand and then apply it to code, but not really knowing what it should look like. That kind of was a little difficult.*

Within this excerpt, Olivia highlights how by not having a group, she had to generate all the ideas and she had no one to collaborate with or discuss possible approaches. In doing this, some of the dimensions such as originality and flexibility are stifled. Further, the experience was more overwhelming compared to other reflections. For example, Nate discussed how when he was confused about the mathematics, he could rely on his coding expertise and listen to groupmates' ideas. In this case, Olivia had to understand the new mathematics, the coding syntax, and how everything fit together. This had the potential to cause frustration for students or potentially limit risk taking. As seen in the observations, Olivia still took risks, but regarding the mathematics she stated that:

> *You could still be kind of creative in how you approach problems. But for me, I only know a set way. So sometimes it can be harder for me to be creative. But I've definitely learned from this semester and then also from these modules that with computation, you can be really creative in how you solve different things.*

Olivia still had a positive experience and engaged in mathematical creativity through the study, and she viewed mathematics as more creative compared to where she started. However, since it can be hard for her to deviate from her set way, coupled with her not having any group members that she could discuss ideas with, her case highlights why computation on its own is not necessarily enough to support mathematical creativity.

The focus within this portion of the discussion is that students reported that other elements, such as their groupwork, were crucial for their computational experience. The notion of computation creating space to foster creativity cannot be divorced from the context in which it was enacted. The tool, namely the engagement with the code and Python modules, would not have functioned in the same manner if implemented alone. A few students did their modules independently because they were sick or traveling. This was such a different experience than when they worked in their small groups that they actually brought this up in their interviews. David noted that it was easier to hit roadblocks when he completed the module on his own, and that there was no one to be able to *"fill in the gaps"* so he had to *"just come up with some sort of solution."* This is in contrast with his prior comments about the modules developing an intuitive understanding of linear algebra and how he was able to play around with the code to see different effects. Computation enacted through coding on its own does not guarantee mathematical creativity. This caveat is extremely important for both the research and pedagogical implications, as it is known that the intended curriculum does not guarantee the translation into enacted curriculum (Johnson et al., 2020).

RESEARCH IMPLICATIONS

There are multiple implications for research that results from this study. This section will detail the research implications for the Research in Undergraduate Mathematics Community, the Computer Science Education Community, and Computing Education more broadly.

Answering the Call Within the Research in Undergraduate Mathematics Education Community for Research on Computation and Mathematics

This work answers the call given to the Research in Undergraduate Mathematics Education (RUME) community for research that explores the relationships between computing and mathematical activity. As argued by Lockwood & Mørken (2021), compared to other areas of undergraduate mathematics education, there is relatively little known about how introducing machine-based computing affects students' mathematical and computational thinking, and what this means for developing equitable classrooms. Therefore, this work begins to answer several key questions including:

- *What kinds of affordances and barriers to student's mathematical* understanding arise *from the integration of machine-based computing into post-secondary curricula?*
- *What are examples of content areas and specific topics that are particularly suitable or unsuitable in which to integrating computing?*

- *Is it possible for engagement with computing in mathematics to change students' views of their own mathematical identities?*

Each of these questions will be addressed using the work from this study, as well as contextualizing the claim with the initial research that seeks to understand the relationship between computation and mathematics in the undergraduate context.

What Kinds of Affordances and Barriers to Student's Mathematical Understanding Arise from the Integration of Machine-Based Computing into Post-Secondary Curricula?

The integration of machine-based computing into student's post-secondary curricula allows for the development of multiple mathematical representations thereby enhancing students' mathematical understanding by being able to view mathematical concepts in multiple ways as well as provides students with new tools that can mediate their mathematical understandings. While all the claims presented within the results section highlighted different affordances of computation, one of the most striking for mathematical understanding was how students engaged with multiple representations. Coding itself provided an opportunity for students to predict and reflect in a low-cost way that enabled experimentation within the code itself and the mathematics. Students do not have to erase all their work if they make a small change, therefore the environment frees them to try different approaches or combination of ideas. Further, the very act of machine-based coding necessitates a constant coordination of multiple computational representations thereby developing a natural way to engage students in multiple mathematical representations. This is a key affordance of coding as the notion of coding in and of itself necessitating multiple representations is something that has just begun to be explored within this community (Lockwood, 2020). Once again, this is an area that widgets or other interactive apps have fallen short in the past. The separation that is presented within the app and not having students 'open the hood' forces students to coordinate the different objects and make the connections. In comparison, when they are coding, they are necessarily having to coordinate the different computational representations. In this study, the students coordinated multiple computational representations as well as the traditional mathematical representations of algebraic, numerical, and graphical. This is a unique affordance that is difficult to replicate outside of coding. As the research in machine-based computation is fairly limited, this study provides a novel perspective with affordances, such as the visualization capabilities, and potential starting barriers that need more research, such as code comprehension and linear coding writing methods, while also highlighting the need for creativity.

What Are Examples of Content Areas and Specific Topics That Are Particularly Suitable or Unsuitable in Which to Integrate Computing?

This work is one of the first studies within the RUME community that explores the integration of computation into Linear Algebra. Prior efforts include but are not limited to calculus, combinatorics, numerical analysis, and mathematical physics (Barichello, 2016; Buteau & Muller, 2017; Lockwood, 2022; Lockwood et al., 2019; Lockwood & Chenne, 2020, 2021; Lovric, 2018; Merkle et al., 2022; Odden et al., 2019; Sand et al., 2022). Students learned linear algebra concepts beginning with defining matrices and going through span, vector spaced, determinants, linear independence, and eigenvalue problems. While this study demonstrates the suitable nature of linear algebra for the integration of computation, some of the subtopics within linear algebra are particularly suitable for computation. Consider the module which focused on determinants and their connections across a geometric, algebraic, and numeric perspective. Within this module, students leveraged the power of the computational environment by coordinating the geometric representation through plotting, the algebraic representation through symbolic Python packages, and numeric by calculating the determinant of the matrix itself. A key point within the literature is how students are able to develop great insight and creativity when developing geometric interpretations of key linear algebra concepts (Larson et al., 2008). This is one of the particular strengths of these specific modules but more broadly within computation enacted through coding. Many programming languages have supports built in for visualization which allows students to plot different objects that were previously unattainable by hand. Therefore, when students are learning about determinants, coding is a highly suitable environment. For researchers, this demonstrates how the most apt topics are those that leverage multiple representations.

Is It Possible for Engagement with Computing in Mathematics to Change Students' Views of Their Own Mathematical Identities?

One of the narratives within this work surrounds how computation and coding can empower students and counter prior negative mathematical narratives by shifting student's world-image of mathematics and developing new mathematical practices. Therefore, this work acts as a call to researchers to consider how coding and computation can be used within the mathematics classroom to pursue issues of equity. This study demonstrated a proof of existence to Lockwood & Mørken's (2021) call, namely that computing does have the potential to change students' views of their own mathematical identities through the shifts in students' mathematical self-image. This work also draws on how this is done in part by the shift in the world-image of mathematics. It is worth noting that this work was done with the PI as the instructor and material designer, and with an implicit focus on ensuring positive mathematical experiences. Simply because a student engages with the coding does not imply a positive affective shift, or any real shift in student view of mathematical identity. As the RUME community has already faced how inquiry-based learning does not guarantee equity (Johnson et al., 2020), it is important to emphasize that computation does have the *potential* to shift student views but not a guarantee. Further, there is the potential that if coding was incorporated in ways that were synonymous with students' prior mathematical experiences, then it simply would have reified inequities and negative mathematical self-images. Therefore, as this study demonstrates the potential to shift, it is imperative to investigate how computation can be used in a more liberatory capacity for those who have been marginalized by mathematics rather than reifying inequities.

Countering Deficit Narratives About Student Mathematical Ability Within Computer Science Education Research

This study shifts the focus by leveraging computational science students' strengths and using computing as a pedagogical tool for students to engage in meaningful mathematical experiences. This work informs the computer science education community by countering some of the pervasive deficit language that researchers use when discussing mathematics and computer science education (Castle, 2023). By leveraging student assets, namely computing, this work provides a different framing of the relationship between mathematics and computation since many studies within computer science education have focused on the relationship between mathematical ability and computation, student perception of the two disciplines, or why mathematics is needed in computing (Konvalina et al., 1983; Whalley et al., 2020). One of the most striking results is the ways in which students who previously had negative mathematical experiences were able to engage in mathematics in novel and positive ways. The mathematics requirements for a computational science degree, or even the connotation of the dominance of mathematics in coding, has the potential to eliminate students who would otherwise be interested (Lavy, 2021). By demonstrating the shift in student perceptions of what it means to do mathematics, as well as their relationship to it, this brings about the engagement of students in mathematics that highlights their strengths rather than reinforcing previous traumatic mathematics experiences.

Furthermore, as previously discussed, students no longer perceived mathematics as being a single solution, single pathway discipline composed of rules and procedures for students; rather, the process of doing mathematics was opened to exploration, multiple methods, and meaningful discovery, consistent with literature (Odden et al., 2019). Students experienced a marked shift from doing mathematics by following a specific set of steps to centering mathematics on exploration. Therefore, this study strongly motivates the utilization of computation as a pedagogical approach within the mathematics classroom, as well as also encouraging the research community to consider how to leverage students' strengths for the promotion of learning, rather than solely focusing on identifying student mistakes.

Pushing the Research Community Beyond Integration

Within the research community, there has been a push for considering the way in which coding is able to support mathematical integration, and yet this study asks the research community to think bigger. How can we do this so that students develop confidence? It raises questions surrounding cognitive processes when students are using the code, versus modifying or creating the code. Much of the work surrounding the integration of computation into mathematics has centered upon providing coding as a tool or using it as a way to do more realistic mathematics. However, this work challenges the community to dream bigger and think of how machine-based computation can be used as a pedagogical tool and to consider in what ways computation can bring about a unique opportunity for mathematical creativity and understanding.

METHODOLOGICAL IMPLICATIONS

Within this study, a key point of discussion centers on how the scope of the tool influences the type of claim and raises questions surrounding whether coding and computational modules are the tool, or the individual functions and coding concepts are the tool. For example, Claim 3 and Claim

4 center on the notion of the tool being computation and coding as a whole. When using this as the unit of analysis for an activity system within the CHAT framework, this emphasizes a broader view of the system.

Figure 5-2: Nate's activity system when considering the tool of mediation to be the Python module as a whole.

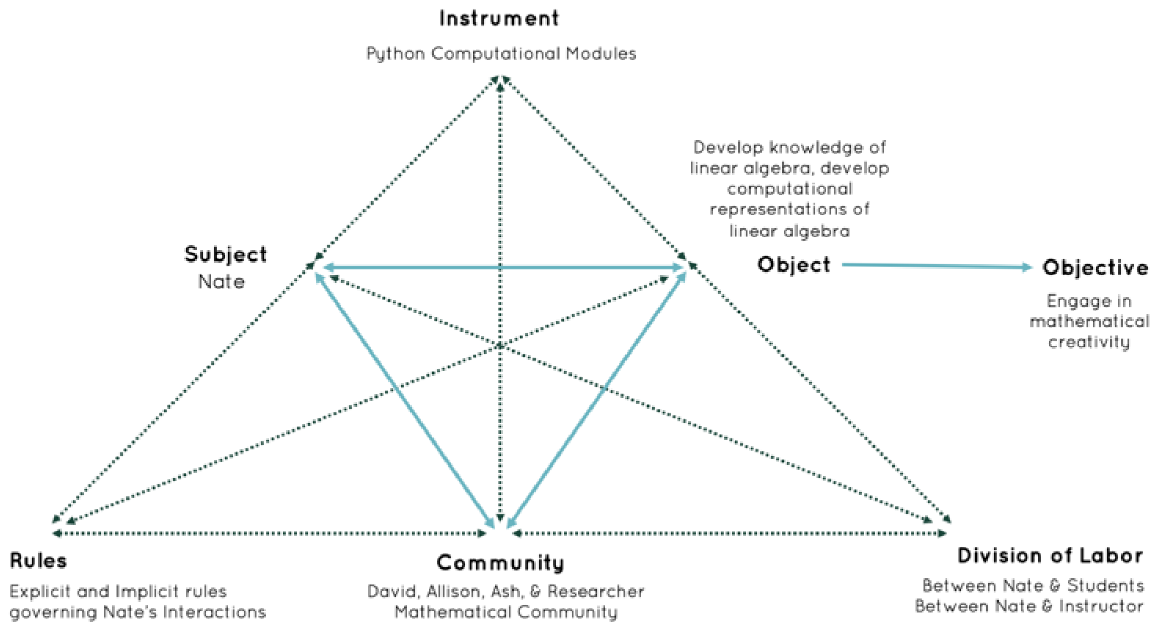
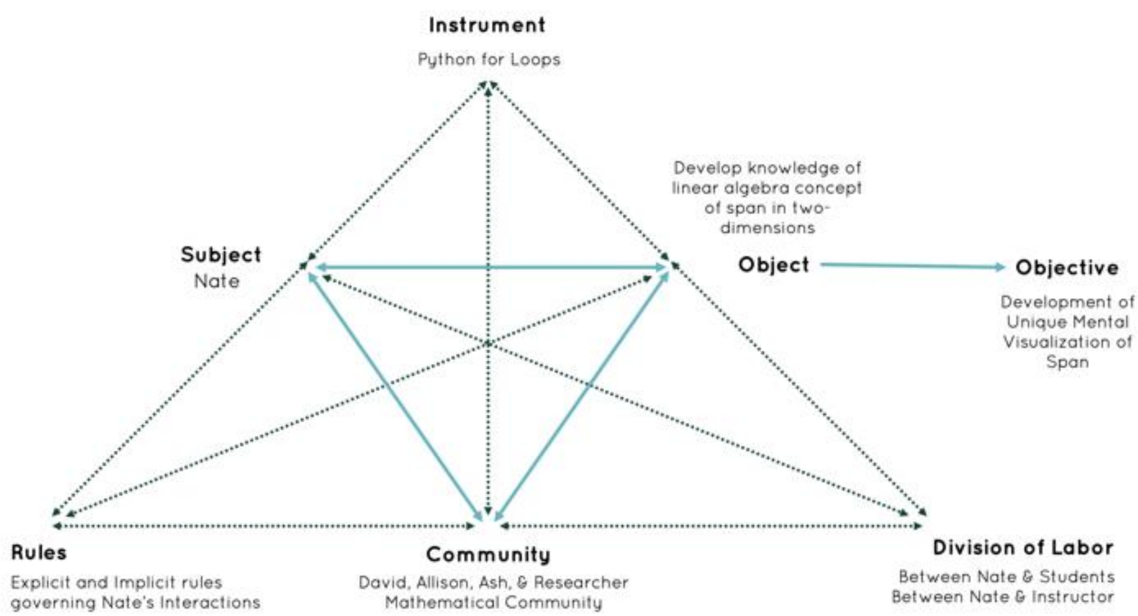


Figure 5-3: Nate's activity system when considering the tool of mediation to be Python for loops, rather than Python as a whole.



Consider the two visualizations of the different activity systems for Nate. The system shown in Figure 5-2 uses the instrument as the Python computational modules whereas the system shown in Figure 5-3 highlights the tool as Python for loops. Specifically, the activity system reflected in Figure 5-3 corresponds to Nate's reflection surrounding why he felt Python was beneficial for his understanding.

> *Being able to visualize stuff, honestly. Cause being able to, I guess take two vectors and then make some nested for Loop to show that it actually can reach basically every point in $\mathbb{R}^2$ or $\mathbb{R}^n$ and graph it and see it visually is really interesting.*

Note that this was not a given task or a specific problem within the computational modules. However, Nate's understanding was actually tied to the computational construction. He took the for loop code, and even though for loops can serve many different purposes, he used it to develop his understanding of span and linear combinations. Because he thought about how when going through the x-y plane, he could iterate through with points, and if vectors were linearly independent, then he should be able to reach any point in $\mathbb{R}^2$. He also referred to the visual component in $\mathbb{R}^2$, and noted that this does expand up to $\mathbb{R}^n$. What is of interest here is that this incorporated a graphical visualization but through the constructs of coding and specific syntax. One of the concerns with $\mathbb{R}^2$ and $\mathbb{R}^3$ is that students develop a conceptualization of vectors that is reality or physics based. But one of the key points is that by conceptualizing span as a nested for loop, this expands to the broader dimensions to which Nate alluded. This highlights how Nate took a computational tool and then adapted it for his own use and developed his understanding of the linear algebra concepts. This was actually not designed for within the modules, nor was this a specific task. Rather this conception is something that works for Nate and so when asked to describe what span is, he notes that it is some scalar combination of the points to achieve any point in $\mathbb{R}^n$. This is in line with his conceptualization because any point is represented by some stage within the nested for loops, but his definition did not rely on graphical interpretation. When asked to discuss basis, one of the key parts was that it was linked to the coordinate system.

While the original question centered on Python as an instrument in general, when considering the tool of the for loop, this led to a specific understanding of span. However, when he reflects on Python as a whole, he breaks apart some of the affordances specifically related to coding. Therefore, the scope of the tool is important to consider as the claims can differ and bring about different key understandings. When considering some of the affective effects of computation enacted through coding, then a focus on this as a whole may be a more apt lens, consistent with trends taken during instrumental genesis and other prior work done using CHAT and mathematical coding. However, when considering the affordances of mathematical understanding, it might be more apt to consider some of the more syntactical level tools. In particular there are certain constructs that may illuminate unique understandings. Further, consider the work of Lockwood (2020) discussed within the literature review. Her assertion is that the code itself and the enumeration strategies give rise to unique combinatorial understandings. However, consider the nested for loop itself: this specific computational syntactical piece reveals unique combinatorial understanding. The hierarchical level of the code itself brings about different understandings of the combinatorial process. Further, it is the function that they created and not necessarily all of coding. This example can go back and forth but this is a way of considering the different mathematical structures. The for loop represents the combination, but the if statement represents whether or not there is a repeated pattern. It is the coding structures that result in the understanding. However, one of the claims at large is that the coding allows for empirical reconceptualization as students are able to generate vast combinatorial sets, something that would not be accessible using traditional methods. This ultimately opens the doors for students to be able to engage in new habits while learning the mathematical content.

In this way, the analysis can then be seen about the initial tool presented, namely the for loop for example, and then from there see how the student takes it up and how that specific one ends up mediating the interaction with the mathematical objectives. Then examination about how the student makes it their own and how the student takes it up can be conducted. From there it may be a more interesting perspective for theories such as instrumental genesis, which originates from activity theory. While the argument has been made for computation as a tool in general, now that some of the unique benefits have been established, for the investigation of mathematical understanding, this has the potential to focus on the desired understandings. For more general affective and other sociocultural implications, then perhaps a more macro scale perspective is appropriate.

PEDAGOGICAL IMPLICATIONS

New Ways to Engage Students in Linear Algebra in an Intuitive Manner

This study provides a fresh perspective on the ways in which to introduce linear algebra to undergraduate students in a way that is focused on developing student understanding, rather than typical content coverage. As discussed within methods, these modules were designed using understanding by design, so the focus was on developing student thinking and understanding, rather than trying to cover all materials. These types of modules could be used to introduce concepts and have students build key theorems followed by the next course session focusing on different proof developments or alternate applications. It is worth noting that conjecturing, a key point in the proof schema, is built within these modules. As discussed throughout, students found the modules extremely helpful and felt that they walked away with a general understanding of the key liner algebra concepts. However, one of the students, David, had already taken linear algebra during his mathematics minor. Therefore, when he entered this study, David was already familiar with some of the topics, or at least had name recognition of the topics. Within this study, many times he seemed to rely on a definitional view of the mathematics and even stated that when the modules were trying to develop a *"more intuitive view"* of the mathematics, this would often be a sticking point. Now this note is not focused on David as an individual, but rather the ways in which his mathematics courses were previously enacted and why this study provides key pedagogical impact. By all traditional standards he was a confident mathematics student who even became an undergraduate learning assistant within the mathematics department. However, when asked what challenged him during the study, he responded,

> *I think the most challenging part was getting stuck on the modules because they kind of try to lead to some conclusion or some theorem or something like that. But you want us to have a more intuitive view of it. Good example is the last one with the eigenvalues. I just wasn't getting to that property where you take the eigenvalues raised to a power. So, I think sometimes when I got stuck there, that was probably the hardest part.*

During his final interview, he continued with multiple similar statements that highlighted how the challenging portions were where he either had to develop or implement a more intuitive view rather than find a *"derivative or find the equation of tangible line or something like that"* because he stated that *"I have that, you know, pretty easy."* Part of this was due to his experience, as highlighted earlier, he reflected,

> *I thought the modules were great, and I thought they were, I would just say that there, unlike anything I've really seen in a math course before I haven't really seen linear algebra presented in that intuitive way before. And I think it's one of the, it's a very unintuitive field,*

but when you're learning it, but once you've learned it, you kind of get a feel for how to deal with matrices and linear, independent, things like that. And it becomes a lot more intuitive to you once you've actually, and I feel like you're kind of putting that intuitive forefront, which is a very good thing to get comfortable from the get-go and the concepts kind of on a deeper level.

David is a student who took numerous mathematical courses and engaged with mathematics a great deal during his undergraduate career and yet this is one of the first times that the linear algebra concepts are intuitive. When he initially engaged with the definitions and concepts, they were unintuitive to him and yet through the experience and coding, the intuition was at the forefront. Of course, as noted prior part of this is due to the understanding by design framework. However, his developed understanding of the linear algebra concepts seems in part due to the code itself. David stated,

I think it requires more creativity to actually write a code that's going to matrix multiplication. I think it's a lot easier to be given two matrix matrices and multiply them than it is to write a code that can multiply any two matrices it's given, which is what we did.

There is a different type of thinking and learning that David appears to be hinting at, and when students engage in coding, there is more required. Consider Olivia’s reflection on her experience within the study, and it is important to note that she was the participant who did not have a group.

I think it definitely did help me understand a lot of stuff, especially for, since a lot of the stuff that we talked about we're talking about things that were like you could place onto a graph, especially the vectors. I feel like that's something that's really hard to explain, and being able to put that into code, and then also messing with your dimensions or messing *with your matrices to see what happens to the vector in code was really helpful in understanding them. And then also was really helpful in understanding the linearly independent vectors or your span and stuff like that especially that little 3D model that we made.*

Within this reflection, Olivia highlighted at how engaging with the code developed her understanding of vectors. For her, there was something qualitatively different about being able to code and investigate the concepts that help gave a natural understanding of the concepts.

This notion of developing an intuitive view of linear algebra, that was done in the context of coding, highlights the potential for pedagogy. Specifically, one of the questions that arises is when you add something in, something else has to come out and how do you make that decision. Further, how can computation be meaningfully integrated? This study offers one potential pathway, namely, to use the computation to introduce a concept and have students then test out different constraints and affordances to then arrive at a theorem. One of the pieces that was not covered, as will be discussed within limitations, is the relation to proving. For a mathematics course that needs students to be able to understand the concepts, prove related theorems, and apply the concepts to novel proofs and worldly applications, this is something to be considered. These types of modules could be used to introduce concepts and have students build key theorems. Then the next day in the course could be dedicated to the proof element, or potentially another application. It is worth noting that conjecturing, a key point in the proof schema, is built within these modules.

Role of Jupyter Notebooks in Mathematical Creativity

Within this study Jupyter notebooks were used to enact the use-modify-create cycle, allowing for an environmental structure to promote mathematical creativity. Jupyter notebooks are used widely within industry (Jin & Johnson, 2020) and computational education, and have been used to design curriculum reform. However, within mathematics education these have been used to a much lesser extent, especially when considering courses beyond mathematical physics, or numerical analysis and most of the research centers on teacher development of the interactive platform (Koehler & Kim, 2018). This study is a research-based look at the affordances of Jupyter notebooks and the design features that connect to specific desired outcomes, namely mathematical creativity. Rather than students developing traditional script, this work highlights the pedagogical power of Jupyter notebooks with respect to creativity. The claims previously discussed focused on the power of experimentation enacted through prediction and reflection cycles. When students engaged in these cycles, often they had multiple cells with all the different trials recorded and the corresponding outputs. For students it was important to have the ability to make changes and have those documented with the corresponding methods. Jupyter notebooks enabled having their trials and thoughts all together, something that is not readily achievable with a script. Further, when using these multiple cells and outputs, students compared between the results, as detailed withing Claim 1, prompting key insights and connections between trials and abstraction of structure. This type of use is unique to the notebook style environment, such as Jupyter, and provides key insights into the ways educators can leverage the coding environment to bring about mathematical creativity.

Role of Groupwork within Mathematical Computation

One of the key implications for pedagogy is that within this study, groupwork was critical for the development of mathematical creativity and the affective shift students discussed within their final interviews. Students leveraged their group members for new ideas. Working in groups also challenged some of the students' prior views of mathematics being a purely individualistic subject. This work highlights the potential that enacting computation within the mathematics classroom has for mathematical creativity, specifically when done in groups.

One of the points that was less salient throughout the results was the potential power that students may hold when they are perceived as good at mathematics and the transferability of this social power to computational settings. During multiple observations of a group, it was documented that when sharing his work, Nate would only ask questions to David even when Allison was sitting next to him. David had the mathematical power in the group due to prior courses, but Allison had more prior coding experience. For both mathematical and computational questions, Nate would default to David. Even when specifically prompted by the module to work with his partner, he did not engage with Allison and waited for David and Ash to be finished to discuss. I want to make it clear that all group members reported positive group experiences and gave specific anecdotes as to why they valued their group. However, from an observer standpoint there were times that I felt uncomfortable and documented this within observation field notes and during the analytical memos. The implication for pedagogy is that a tool, such as Python, may introduce more ways to be successful in mathematics. However, the sociocultural rules can still dominate and give the most power based on the mathematical identity of students rather than valuing all ways that students can participate. This is to serve as a reminder that inquiry does not guarantee equity (Johnson et al., 2019). Therefore, while computational groupwork has a great potential to develop student understanding and create positive mathematical experiences, it is critical to consider the ways in which it can reify inequities.

LIMITATIONS AND FUTURE DIRECTIONS

This study has multiple limitations, both avoidable and unavoidable. The purpose of this section is to detail the present study's limitations and some of the implications. The latter section will then detail how these limitations may inform future research.

Honors Students

To begin with, this study recruited students within the honors college, as this study offered students an alternate to an additional final project for the honors credit. This was done intentionally, as each student would be engaging with the study for approximately 20 hours over the course of the semester. This was a large time commitment and financial compensation would not necessarily have been enough to motivate students to participate, nor was that within the research budget to ensure adequate compensation. This option was simply replacing a prior engagement so there was not a net loss of time for students. However, in doing this the group of students that resulted were those that had been academically successful in the past. It is important to note that simply because they were a part of the honors college did not guarantee a positive relationship with mathematics. Consider Kylie, Ivy, and Nate. Each of these students had negative mathematical experiences that impacted their mathematical self-image. However, each were motivated to complete the study and to participate to ensure that they earned honors credit. Further, what was being asked of students during this study was very different than some of the extrinsic motivators within the honors college. These students are high-achieving students who pursue academic excellence. Therefore, in expansion of this work I hope to introduce these modules and this approach to integration of computation within the mathematics classroom to students across the academic spectrum. I want to ensure that systems-based inequities are not reified using this approach to learning the mathematics. Therefore, the next course of work would be to expand the work to a full linear algebra course. This widens the target audience and also develops the modules in a class-based setting, a non-trivial effort.

Supplemental Modules

One of the strengths of the materials is that the modules were developed as self-contained units and can serve as an addition to a mathematics course. However, these materials were piloted within individual small groups in a grade-free environment. This is a noteworthy change from students' prior interactions with mathematics which were in the classroom where there was external motivation from high stakes testing. It is not my desire to advocate for high stakes grading within the mathematics classroom, but it is the reality that many students experience. In the study, students were detached from this setting as there was no official grading of the products. Rather, the instructor of record simply received a checklist of whether a student had completed their reflections, sessions, and interviews. This in turn frees the student from some of the pressures associated with grading and the desire to do it 'right'. Much of this pressure comes from the ways in which instructors grade their course and the established norms. Therefore, I want the next phase of the development to be within my own classroom where I can simultaneously use the modules to introduce these concepts while also having control over the grading. This also provides insight into how different groups can support each other and what happens when this is broken into smaller time constraints rather than the weekly two-hour meetings.

Students' Prior Coding Exposure in Their Introduction to Computing Course

One of the key considerations about this study is that the students were recruited from an introduction to computational modeling course. This was explicitly done to leverage student assets, namely their confidence in their computational abilities, in order to engage students in mathematical creativity and development of linear algebra concepts. However, this study circumvents one of the dominating questions in the literature about the relation of teaching coding syntax and new mathematics simultaneously (Castle, 2021; DeJarnette, 2018, 2019) and the required cognitive demand. This study does provide some potential suggestions to start with, namely perhaps that some initial coding specific syntax lessons should be developed first. Consider the case of Nate who already felt 'shaky' about the mathematics. He was able to lean on his understanding of Python when he was not as confident in the mathematics. If he had to learn the coding syntax simultaneously then he would be unable to do this to the same degree. As universities are developing more introduction to computation courses, then perhaps one of the potential avenues is focusing on how to support the development of these courses and having mathematics students enroll in these courses prior to their computing-based mathematics explorations. However, this circumvents the problem and offloads the work to a different set of individuals who may not be as committed to enacting the computation in a student-centered manner. Therefore, perhaps initially scaffolding in the coding where the mathematical concepts are more straightforward, and the coding is being introduced in the beginning of the course is more apt. Then there could be a constant check on the balance of coding and mathematical concepts to reduce the strain on the student. However, this would need to be investigated further, and across multiple student groups.

Student Identity

Within this study student identity is not specifically addressed and this is a critical dimension to consider, especially when scaling up materials and widening the student demographic. Students' relationship to mathematics was examined within the study, but what was less evident was how social identity such as race, ethnicity, gender and other demographics intersected with their experience. There are countless pieces in literature examining how identity mediates mathematical experiences and how mathematics is not a neutral space (Bishop, 2012; Darragh, 2016; Esmonde, 2009; Larnell, 2011; Leyva, 2017; McGee & Martin, 2011; Mendick, 2005; Stinson, 2013).

However, what is less known is what happens when computing is introduced. The computer science education community is starting to consider questions surrounding equity and have documented the ways in which systemic inequities minoritize students within the classroom. Therefore, is the integration of coding into the mathematics a compounding effect? Are there ways in which some of the stereotypes about who should be present within the course are minimized or exasperated? These are critical questions that this study does not address. There are instances such as the observations between Allison, Nate, and David that highlight potential gender dynamics mediating the interaction but there is not much exploration along these lines given the scope of the study. Therefore, I plan to follow this work up with a study that looks at identity across computation and mathematics, and how it mediates students' understandings. This explores along the sociocultural practices and rules, specifically understanding how those mediate student action.

CONCLUDING REMARKS

This study provides a novel framing of the integration of computation into the undergraduate mathematics classroom, both with respect to the focus on mathematical creativity as well as providing a different conceptualization to counter some of the deficit messaging within the computer science education community. This study shifted the focus by leveraging computational science students' strengths and using computing as a pedagogical tool for students to engage in meaningful mathematical experiences. This work puts forth multiple claims that contribute directly to the research field at large as well as provides different pedagogical suggestions and potential implementations. However, this study specifically calls on the mathematics education and computational education communities to dream bigger than simply the integration of computation into the mathematics classroom. Rather, the focus is on introducing this tool to change students' relationship with mathematics by changing the very ways in which students perceive what it is to do mathematics. This in turn enables mathematical creativity and opens doors for students who had been previously marginalized by mathematics. If I have one hope for this study, it is to challenge researchers and educators to push beyond the simple use of a tool within the classroom but to consider the ways in which coding and computation could radically change the mathematics classroom and students' lives.

REFERENCES

Ahmed, G., Nouri, J., Zhang, L. & Norén, E. (2020). *Didactic Methods of Integrating Programming in Mathematics in Primary School*. 261–267. https://doi.org/10.1145/3328778.3366839

Aiken, L. R. (1973). Ability and Creativity in Mathematics1. *Review of Educational Research*, *43*(4), 405–432. https://doi.org/10.3102/00346543043004405

Andrews-Larson, C., Wawro, M. & Zandieh, M. (2017). A hypothetical learning trajectory for conceptualizing matrices as linear transformations. *International Journal of Mathematical Education in Science and Technology*, *48*(6), 809–829. https://doi.org/10.1080/0020739x.2016.1276225

Arney, K., Blyman, K., Cepeda, J., Lynch, S., Prokos, M. & Warnke, S. (2020). *Going Beyond Promoting: Preparing Students to Creatively Solve Future Problems*. *10*, 348–376. https://doi.org/10.5642/jhummath.202002.16

Attallah, B., Ilagure, Z. & Chang, Y. K. (2019). *The Impact of Competencies in Mathematics and beyond on Learning Computer Programming in Higher Education*. 77–81. https://doi.org/10.1109/ctit.2018.8649527

Baldwin, D., Walker, H. M. & Henderson, P. B. (2013). The roles of mathematics in computer science. *ACM Inroads*, *4*(4), 74–80. https://doi.org/10.1145/2537753.2537777

Barichello, L. (2016). The movement towards a more experimental approach to problem solving in mathematics using coding. *International Journal of Mathematical Education in Science and Technology*, *47*(5), 791–797. https://doi.org/10.1080/0020739x.2015.1109147

Batiibwe, M. S. K. (2019). Using Cultural Historical Activity Theory to understand how emerging technologies can mediate teaching and learning in a mathematics classroom: a review of literature. *Research and Practice in Technology Enhanced Learning*, *14*(1), 12. https://doi.org/10.1186/s41039-019-0110-7

Battista, A. (2015). Activity Theory and Analyzing Learning in Simulations. *Simulation & Gaming*, *46*(2), 187–196. https://doi.org/10.1177/1046878115598481

Benadé, T. & Liebenberg, J. (2019). The relevance of a Mathematics course for computer science students. *Swan Delta Proceedings: The 12th Delta Conference on the Teaching and Learning of Undergraduate Mathematics and Statistics*, 2–10.

Bishop, J. P. (2012). “She’s Always Been the Smart One. I’ve Always Been the Dumb One”: Identities in the Mathematics Classroom. *Journal for Research in Mathematics Education*, *43*(1), 34–74. https://doi.org/10.5951/jresematheduc.43.1.0034

Blyman, K., Arney, K., Adams, B. & Hudson, T. (2020). Does Your Course Effectively Promote Creativity? Introducing the Mathematical Problem Solving Creativity Rubric. *Journal of Humanistic Mathematics*, *10*(2), 157–193. https://doi.org/10.5642/jhummath.202002.09

Boaler, J. (2018). The Creativity and Beauty in Mathematics. In *Mathematical Mindsets* (pp. 21–32). Jossey-Bass.

Boaler, J. & Staples, M. (2008). Creating Mathematical Futures through an Equitable Teaching Approach: The Case of Railside School. *Teachers College Record*, *110*(3), 608–645.

Broley, L., Caron, F. & Saint-Aubin, Y. (2018). Levels of Programming in Mathematical Research and University Mathematics Education. *International Journal of Research in Undergraduate Mathematics Education*, *4*(1), 38–55. https://doi.org/10.1007/s40753-017-0066-1

Buteau, C., Gueudet, G., Muller, E., Mgombelo, J. & Sacristán, A. I. (2020). University students turning computer programming into an instrument for 'authentic' mathematical work. *International Journal of Mathematical Education in Science and Technology*, *51*(7), 1020–1041. https://doi.org/10.1080/0020739x.2019.1648892

Buteau, C. & Muller, E. (2017). Assessment in Undergraduate Programming-Based Mathematics Courses. *Digital Experiences in Mathematics Education*, *3*, 97–114. https://doi.org/10.1007/s40751-016-0026-4

Buteau, C., Muller, E. & Marshall, N. (2015). When a University Mathematics Department Adopted Core Mathematics Courses of an Unintentionally Constructionist Nature: Really? *Digital Experiences in Mathematics Education*, *1*, 133–155. https://doi.org/10.1007/s40751-015-0009-x

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 Experiences in Mathematics Education*, *2*, 142–166. https://doi.org/10.1007/s40751-016-0017-5

Castle, S. D. (2021). Connecting Computation: Mediating Mathematical Knowledge Through Computational Modules. In S. S. Karunakaran & A. Higgins (Eds.), *2021 Research in Undergraduate Mathematics Education Reports* (pp. 30–38).

Castle, S. D. (2023a). Leveraging Computational Science Student's Coding Strengths for Mathematics Learning. *Proceedings of the 54th ACM Technical Symposium on Computing Science Education*, 263–269. https://doi.org/10.1145/3545945.3569861

Castle, S. D. (2023b). Exploring How Computation Can Foster Mathematical Creativity in Linear Algebra Modules. *Proceedings of the 25th Annual Conference on Research in Undergraduate Mathematics Education*.

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. https://doi.org/10.1080/00031305.2015.1093029

Cole, M. & Engeström, Y. (1993). A cultural-historical approach to distributed cognition. *Distributed Cognitions: Psychological and Educational Considerations*, 1–46.

Craft, A., Cremin, T., Burnard, P., Dragovic, T. & Chappell, K. (2013). Possibility thinking: culminative studies of an evidence-based concept driving creativity? *International Journal of Primary, Elementary and Early Years Education*, *41*(5), 538–556. https://doi.org/10.1080/03004279.2012.656671

Darragh, L. (2016). Identity research in mathematics education. *Educational Studies in Mathematics*, *93*(1), 19–33. https://doi.org/10.1007/s10649-016-9696-5

DeJarnette, A. F. (2016). Students' discourse when working in pairs with Etoys in an eighth-grade mathematics class. *Language and Education*, *30*, 485–499. https://doi.org/10.1080/09500782.2016.1141934

DeJarnette, A. F. (2018). Using student positioning to identify collaboration during pair work at the computer in mathematics. *Linguistics and Education*, *46*, 43–55. https://doi.org/10.1016/j.linged.2018.05.005

DeJarnette, A. F. (2019). Students' Challenges with Symbols and Diagrams when Using a Programming Environment in Mathematics. *Digital Experiences in Mathematics Education*, *5*, 36–58. https://doi.org/10.1007/s40751-018-0044-5

Denning, P. J. (2010). What is Computation? *Ubiquity*, *2010*(November), 1. https://doi.org/10.1145/1880066.1880067

diSessa, A. A. (2018). Computational Literacy and "The Big Picture" Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning*, *20*(1), 3–31. https://doi.org/10.1080/10986065.2018.1403544

Dunleavy, T. K. (2018). High School Algebra Students Busting the Myth about Mathematical Smartness: Counterstories to the Dominant Narrative "Get It Quick and Get It Right." *Education Sciences*, *8*(2), 58. https://doi.org/10.3390/educsci8020058

Engelman, S., Magerko, B., McKlin, T., Miller, M., Edwards, D. & Freeman, J. (2017). Creativity in authentic STEAM education with Earsketch. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 183–188. https://doi.org/10.1145/3017680.3017763

Engeström, Y. (2014). *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research*. Cambridge University Press. https://doi.org/10.1017/CBO9781139814744

Esmonde, I. (2009). Ideas and Identities: Supporting Equity in Cooperative Mathematics Learning. *Review of Educational Research*, *79*(2), 1008–1043. https://doi.org/10.3102/0034654309332562

Fetterly, J. (2020). Fostering Mathematical Creativity While Impacting Beliefs and Anxiety in Mathematics. *Journal of Humanistic Mathematics*, *10*(2), 102–128. https://doi.org/10.5642/jhummath.202002.07

Feurzeig, W., Papert, S. A. & Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, *19*(5), 487–501. https://doi.org/10.1080/10494820903520040

Gadanidis, G., Hughes, J. M., Minniti, L. & White, B. J. G. (2016). Computational Thinking, Grade 1 Students and the Binomial Theorem. *Mathematics and Programming*, *3*, 77–96. https://doi.org/10.1007/s40751-016-0019-3

García-Perales, R. & Palomares-Ruiz, A. (2020). Education in Programming and Mathematical Learning: Functionality of a Programming Language in Educational Processes. *Sustainability*, *12*(23), 1–15. https://doi.org/10.3390/su122310129

Gontijo, C. H. (2018). Mathematics Education and Creativity: A Point of View from the Systems Perspective on Creativity. In N. Amado, S. Carreira & K. Jones (Eds.), *Broadening the Scope of Research on Mathematical Problem Solving* (pp. 375–386). Springer. https://doi.org/10.1007/978-3-319-99861-9_16

Gries, D., Marion, B., Henderson, P. & Schwartz, D. (2001). How mathematical thinking enchances computer science problem solving. *ACM SIGCSE Bulletin*, *33*(1), 390–391. https://doi.org/10.1145/366413.364754

Hart, M., Early, J. P. & Brylow, D. (2008). A novel approach to K-12 CS education: linking mathematics and computer science. *ACM SIGCSE Bulletin*, *40*(1), 286–290. https://doi.org/10.1145/1352322.1352234

Haylock, D. (1997). Recognizing mathematical creativity in schoolchildren. *ZDM*, *29*, 68–74. https://doi.org/10.1007/s11858-997-0002-y

Helson, R. (1983). Creative mathematicians. *Genius and Eminence: The Social Psychology of Creativity and Exceptional Achievement*, 311–330.

Henriksen, D., Mishra, P. & Mehta, R. (2015). Novel, Effective, Whole: Toward a NEW Framework for Evaluations of Creative Products. *Journal of Technology and Teacher Education*, *23*(3), 455–478. http://www.editlib.org/p/151574/

Hurst, M. & Cordes, S. (2017). When Being Good at Math Is Not Enough: How Students' Beliefs About the Nature of Mathematics Impact Decisions to Pursue Optional Math Education. In U. X. Eligio (Ed.), *Understanding emotions in mathematical thinking and* *learning* (pp. 221–241). Elsevier Academic Press. https://doi.org/10.1016/B978-0-12-802218-4.00008-X

Isomottonen, V., Lakanen, A. J. & Nieminen, P. (2020). Exploring Creativity Expectation in CS1 Students' View of Programming. *2020 IEEE Frontiers in Education Conference*, 1–8. https://doi.org/10.1109/fie44824.2020.9274134

Israel, M. & Lash, T. (2020). From classroom lessons to exploratory learning progressions: mathematics + computational thinking. *Interactive Learning Environments*, *28*(3), 362–382. https://doi.org/10.1080/10494820.2019.1674879

Johnson, E., Andrews-Larson, C., Keene, K., Melhuish, K., Keller, R. & Fortune, N. (2020). Inquiry and Gender Inequity in the Undergraduate Mathematics Classroom. *Journal for Research in Mathematics Education*, *51*(4), 504–516. https://doi.org/10.5951/jresematheduc-2020-0043

Jonassen, D. H. & Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning environments. *Educational Technology Research and Development*, *47*(1), 61–79. https://doi.org/10.1007/bf02299477

Kaptelinin, V. & Nardi, B. A. (1997). Activity theory: Basic Concepts and Applications. *International Conference on Human-Computer Interaction*, 189–201. https://doi.org/10.1145/1120212.1120321

Kaptelinin, V. & Nardi, B. A. (2006). *Acting with technology: Activity theory and interaction design*. MIT press.

Karakok, G., Savic, M., Tang, G. & Turkey, H. E. (2018). Mathematicians' views on undergraduate students' creativity. *CERME 9 - Ninth Congress of the European Society for Research in Mathematics Education*, 1003–1009. https://doi.org/10.1090/spec/079

Karwowski, M. & Lebuda, I. (2017). Creative Self-Concept: A Surface Characteristic of Creative Personality. In G. J. Feist, R. Reiter-Palmon & J. C. Kaufman (Eds.), *The Cambridge Handbook of Creativity and Personality Research* (p. Creative Self-Concept pp 84-101 A Surface Characteristic of Creative Personality). Cambridge University Press. https://doi.org/10.1017/9781316228036

Kattou, M., Kontoyianni, K., Pitta-Pantazi, D. & Christou, C. (2013). *Connecting mathematical creativity to mathematical ability*. *45*, 167–181. https://doi.org/10.1007/s11858-012-0467-1

Kaufmann, O. T. & Stenseth, B. (2020). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, *52*(7), 1029–1048. https://doi.org/10.1080/0020739x.2020.1736349

Krause, A. J., Maccombs, R. J. & Wong, W. W. Y. (2020). Experiencing Calculus Through Computational Labs: Our Department's Cultural Drift Toward Modernizing Mathematics

Instruction. *Problems, Resources, and Issues in Mathematics Undergraduate Studies*, *3–5*. https://doi.org/10.1080/10511970.2020.1799457

Kuutti, K. (1996). *Activity Theory As A Potential Framework for Human-Computer Interaction Research* (B. A. Nardi, Ed.; pp. 17–44). The MIT Press.

Larnell, G. V. (2011). More Than Just Skill: Examining Mathematics Identities, Racialized Narratives, and Remediation Among Black Undergraduates. *Journal for Research in Mathematics Education*, *47*(3), 233–269. https://doi.org/10.5951/jresematheduc.47.3.0233

Laursen, S. L., Hassi, M. L., Kogan, M. & Weston, T. J. (2014). Benefits for Women and Men of Inquiry-Based Learning in College Mathematics: A Multi-Institution Study. *Journal for Research in Mathematics Education*, *45*(4), 406–418. https://doi.org/10.5951/jresematheduc.45.4.0406

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J. & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

Leikin, R. (2019). Exploring Mathematical Creativity Using Multiple Solution Tasks. In A. Berman & B. Koichu (Eds.), *Creativity in Mathematics and the Education of Gifted Students* (pp. 129–145). Sense Publishers. https://doi.org/10.1163/9789087909352_010

Leontiev, A. N. (1974). The problem of activity in psychology. *Soviet Psychology*, *13*(2), 4–33.

Leontiev, A. N. (1978). *Activity, consciousness, and personality*. Prentice-Hall.

Leslie, S.-J., Cimpian, A., Meyer, M. & Freeland, E. (2015). Expectations of brilliance underlie gender distributions across academic disciplines. *Science*, *347*(6219), 262–265. https://doi.org/10.1126/science.1261375

Leung, S. S. (1997). On the role of creative thinking in problem posing. *ZDM*, *29*(3), 81–85. https://doi.org/10.1007/s11858-997-0004-9

Lew, K., Fukawa-Connelly, T. P., Mejía-Ramos, J. P. & Weber, K. (2016). Lectures in advanced mathematics: Why students might not understand what the mathematics professor is trying to convey. *Journal for Research in Mathematics Education*, *47*, 162–198. https://doi.org/10.5951/jresematheduc.47.2.0162

Leyva, L. A. (2017). *Unpacking the Male Superiority Myth and Masculinization of Mathematics at the Intersections: A Review of Research on Gender in Mathematics Education*. *48*, 397. https://doi.org/10.5951/jresematheduc.48.4.0397

Liao, Y.-K. C. & Bright, G. W. (2005). Effects of Computer Programming on Cognitive Outcomes: A Meta-Analysis. *Journal of Educational Computing Research*, *7*(3), 251–268. https://doi.org/10.2190/e53g-hh8k-ajrr-k69m

Lockwood, E. (2013). A model of students' combinatorial thinking. *The Journal of Mathematical Behavior*, *32*(2), 251–265. https://doi.org/10.1016/j.jmathb.2013.02.008

Lockwood, E. (2022). Leveraging Prediction and Reflection in a Computational Setting to Enrich Undergraduate Students' Combinatorial Thinking. *Cognition and Instruction*, *40*(3), 413–455. https://doi.org/10.1080/07370008.2021.2020793

Lockwood, E. & Chenne, A. D. (2020). Enriching Students' Combinatorial Reasoning through the Use of Loops and Conditional Statements in Python. *International Journal of Research in Undergraduate Mathematics Education*, *6*(3), 303–346. https://doi.org/10.1007/s40753-019-00108-2

Lockwood, E. & Chenne, A. D. (2021). Reinforcing key combinatorial ideas in a computational setting: A case of encoding outcomes in computer programming. *The Journal of Mathematical Behavior*, *62*, 100857. https://doi.org/10.1016/j.jmathb.2021.100857

Lockwood, E., DeJarnette, A. F. & Thomas, M. (2019). Computing as a mathematical disciplinary practice. *The Journal of Mathematical Behavior*, *54*, 1–18. https://doi.org/10.1016/j.jmathb.2019.01.004

Lockwood, E. & Mørken, K. (2021). A Call for Research that Explores Relationships between Computing and Mathematical Thinking and Activity in RUME. *International Journal of Research in Undergraduate Mathematics Education*, *7*(3), 404–416. https://doi.org/10.1007/s40753-020-00129-2

Lonchamp, J. (2012). An instrumental perspective on CSCL systems. *International Journal of Computer-Supported Collaborative Learning*, *7*, 211–237. https://doi.org/10.1007/s11412-012-9141-4

Lovric, M. (2018). Programming and Mathematics in an Upper-Level University Problem-Solving Course. *Problems, Resources, and Issues in Mathematics Undergraduate Studies*, *28*(7), 683–698. https://doi.org/10.1080/10511970.2017.1403524

Marshall, N. & Buteau, C. (2014). Learning mathematics by designing, programming, and investigating with interactive, dynamic computer-based objects. *International Journal for Technology in Mathematics Education*, *21*(2), 49–63.

Maschietto, M. (2015). The Arithmetical Machine Zero + 1 in Mathematics Laboratory: Instrumental Genesis and Semiotic Mediation. *International Journal of Science and Mathematics Education*, *13*, 121–144. https://doi.org/10.1007/s10763-013-9493-x

McGee, E. O. & Martin, D. B. (2011). *"You Would Not Believe What I Have to Go Through to Prove My Intellectual Value!" Stereotype Management Among Academically Successful Black Mathematics and Engineering Students*. *48*(6), 1347–1389. https://doi.org/10.3102/0002831211423972

Mendick, H. (2005). A beautiful myth? the gendering of being/doing “good at maths.” *Gender and Education*, *17*(2), 203–219. https://doi.org/10.1080/0954025042000301465

Merkle, L., Doyle, M., Sheard, J., Soh, L.-K., Dorn, B., Kalathas, P., Parham-Mocello, J., Elliot, R. & Lockwood, E. (2022). Exploring Math + CS in a Secondary Education Methods Course. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, 689–695. https://doi.org/10.1145/3478431.3499405

Miller, L. D., Soh, L. K., Chiriacescu, V., Ingraham, E., Shell, D. F., Ramsay, S. & Hazley, M. P. (2013). Improving learning of computational thinking using creative thinking exercises in CS-1 computer science courses. *2013 IEEE Frontiers in Education Conference*, 1426–1432. https://doi.org/10.1109/fie.2013.6685067

Moore-Russo, D. & Demler, E. L. (2018). Linking Mathematical Creativity to Problem Solving: Views from the Field. In N. Amado, S. Carreira & K. Jones (Eds.), *Broadening the Scope of Research on Mathematical Problem Solving* (pp. 321–345). Springer. https://doi.org/10.1007/978-3-319-99861-9_14

Nardi, B. A. (1996). Activity Theory and Human-Computer Interaction. In *Context and Consciousness: Activity Theory and Human-Computer Interaction* (Vol. 436, pp. 7–16). The MIT Press.

Odden, T. O. B., Lockwood, E. & Caballero, M. D. (2019). Physics computational literacy: An exploratory case study using computational essays. *Physical Review Physics Education Research*, *15*(2), 020152. https://doi.org/10.1103/physrevphyseducres.15.020152

Papert, S. (1993). *Mindstorms: Children, Computers, And Powerful Ideas*. Basic Book.

Pei, C. (Yu), Weintrop, D. & Wilensky, U. (2018). Cultivating Computational Thinking Practices and Mathematical Habits of Mind in Lattice Land. *Mathematical Thinking and Learning*, *20*(1), 75–89. https://doi.org/10.1080/10986065.2018.1403543

Pérez, A. (2018). A Framework for Computational Thinking Dispositions in Mathematics Education. *Journal for Research in Mathematics Education*, *49*(4), 424–461. https://doi.org/10.5951/jresematheduc.49.4.0424

Poincaré, H. (1910). Mathematical creation. *The Monist*, 321–335.

Ralston, A. (2005). Do we need ANY mathematics in computer science curricula? *ACM SIGCSE Bulletin*, *37*(2), 6–9. https://doi.org/10.1145/1083431.1083433

Resnick, M. (2017). Peers Learning Communities. In *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*.

Rich, K. M., Spaepen, E., Strickland, C. & Moran, C. (2019). Synergies and differences in mathematical and computational thinking: implications for integrated instruction. *Interactive Learning Environments*, *28*(3), 1–12. https://doi.org/10.1080/10494820.2019.1612445

Riling, M. (2020). Recognizing Mathematics Students as Creative: Mathematical Creativity as Community-Based and Possibility-Expanding. *Journal of Humanistic Mathematics*, *10*(2), 6–39. https://doi.org/10.5642/jhummath.202002.04

Robins, A., Moskal, A., Ko, A. J., McCauley, R., Franklin, D., Coenraad, M., Palmer, J., Eatinger, D., Zipp, A., Anaya, M., White, M., Pham, H., Gökdemir, O. & Weintrop, D. (2020). An Analysis of Use-Modify-Create Pedagogical Approach's Success in Balancing Structure and Student Agency. *Proceedings of the 2020 ACM Conference on International Computing Education Research*, 14–24. https://doi.org/10.1145/3372782.3406256

Romero, M., Lepage, A. & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*. https://doi.org/10.1186/s41239-017-0080-z

Roth, W.-M. (2004). INTRODUCTION: "Activity Theory and Education: An Introduction." *Mind, Culture, and Activity*, *11*(1), 1–8. https://doi.org/10.1207/s15327884mca1101_1

Roth, W.-M. (2012). Cultural-historical activity theory: Vygotsky's forgotten and suppressed legacy and its implication for mathematics education. *Mathematics Education Research Journal*, *24*(1), 87–104. https://doi.org/10.1007/s13394-011-0032-1

Sand, O. P., Lockwood, E., Caballero, M. D. & Mørken, K. (2022). Three cases that demonstrate how students connect the domains of mathematics and computing. *The Journal of Mathematical Behavior*, *67*, 1–27. https://doi.org/10.1016/j.jmathb.2022.100955

Sangwin, C. J. & O'Toole, C. (2017). Computer programming in the UK undergraduate mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, *48*, 1133–1152. https://doi.org/10.1080/0020739x.2017.1315186

Savic, M. (2016). Mathematical Problem-Solving via Wallas' Four Stages of Creativity: Implications for the Undergraduate Classroom. *The Mathematics Enthusiast*, *13*(3), 255–278. https://doi.org/10.54870/1551-3440.1377

Savic, M., Karakok, G., Tang, G., Turkey, H. E. & Naccarato, E. (2017). Formative Assessment of Creativity in Undergraduate Mathematics: Using a Creativity-in-Progress Rubric (CPR) on Proving. In R. Leikin & B. Sriraman (Eds.), *Creativity and Giftedness* (pp. 23–46). Spinger. https://doi.org/10.1007/978-3-319-38840-3

Scharlau, B., McDermott, R., Pears, A., Sabin, M., Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A., Isvik, A., Bounajim, D., Wiebe, E. & Barnes, T. (2019). Use, Modify, Create. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 395–401. https://doi.org/10.1145/3304221.3319786

Schulte, C. & Knobelsdorf, M. (2007). Attitudes towards computer science-computing experiences as a starting point and barrier to computer science. *Proceedings of the Third International Workshop on Computing Education Research - ICER '07*, 27–38. https://doi.org/10.1145/1288580.1288585

Selbach-Allen, M., Williams, C. & Boaler, J. (2020). What Would the Nautilus Say? Unleashing Creativity in Mathematics! *Journal of Humanistic Mathematics*, *10*(2), 391–414. https://doi.org/10.5642/jhummath.202002.18

Selinski, N. E., Rasmussen, C., Wawro, M. & Zandieh, M. (2014). A Method for Using Adjacency Matrices to Analyze the Connections Students Make Within and Between Concepts: The Case of Linear Algebra. *Journal for Research in Mathematics Education*, *45*(5), 550–583. https://doi.org/10.5951/jresematheduc.45.5.0550

Sigurdson, N. & Petersen, A. (2017). Student perspectives on mathematics in computer science. *17th Koli Calling International Conference on Computing Education*, 108–117. https://doi.org/10.1145/3141880.3141888

Silver, E. A. (1997). Fostering creativity through instruction rich in mathematical problem solving and problem posing. *ZDM*, *29*, 75–80. https://doi.org/10.1007/s11858-997-0003-x

Singer, F. M. & Voica, C. (2015). Is Problem Posing a Tool for Identifying and Developing Mathematical Creativity? In *Mathematical Problem Posing: From Research to Effective Practice* (pp. 141–174). Springer New York. https://doi.org/10.1007/978-1-4614-6258-3_7

Sriraman, B. (2009). The characteristics of mathematical creativity. *ZDM*, *41*, 13–27. https://doi.org/10.1007/s11858-008-0114-z

Sriraman, B., Haavold, P. & Lee, K. (2013). Mathematical creativity and giftedness: A commentary on and review of theory, new operational views, and ways forward. *ZDM*, *45*, 215–225. https://doi.org/10.1007/s11858-013-0494-6

Stake, R. E. (1995). *The Art of Case Study Research*. Sage Publications.

Stewart, S., Axler, S., Beezer, R., Boman, E., Catral, M., Harel, G., McDonald, J., Strong, D. & Wawro, M. (2022). The Linear Algebra Curriculum Study Group (LACSG 2.0) Recommendations. *Notices of the American Mathematical Society*, *69*(05), 1. https://doi.org/10.1090/noti2479

Stewart, S. & Thomas, M. O. J. (2009). A framework for mathematical thinking: the case of linear algebra. *International Journal of Mathematical Education in Science and Technology*, *40*(7), 951–961. https://doi.org/10.1080/00207390903200984

Stinson, D. W. (2013). Negotiating the "White Male Math Myth": African American Male Students and Success in School Mathematics. *Journal for Research in Mathematics Education*, *44*, 69–99. https://doi.org/10.5951/jresematheduc.44.1.0069

Sysło, M. M. & Kwiatkowska, A. B. (2014). Learning Mathematics Supported By Computational Thinking. In *Constructionism and Creativity* (pp. 258–268).

Tall, D. & Thomas, M. (1991). Encouraging versatile thinking in algebra using the computer. *Educational Studies in Mathematics*, *22*(2), 125–147. https://doi.org/10.1007/bf00555720

Tedre, M. & Denning, P. J. (2016). The long quest for computational thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education*, 120–129. https://doi.org/10.1145/2999541.2999542

Tularam, G. A. & Hulsman, K. (2015). A Study of Students' Conceptual, Procedural Knowledge, Logical Thinking and Creativity During the First Year of Tertiary Mathematics. *International Journal for Mathematics Teaching & Learning*, 1–41.

Vale, I., Pimentel, T. & Barbosa, A. (2018). The Power of Seeing in Problem Solving and Creativity: An Issue Under Discussion. In N. Amado, S. Carreira & K. Jones (Eds.), *Broadening the Scope of Research on Mathematical Problem Solving* (pp. 243–272). Springer. https://doi.org/10.1007/978-3-319-99861-9_11

Veaux, R. D. D., Agarwal, M., Averett, M., Baumer, B. S., Bray, A., Bressoud, T. C., Bryant, L., Cheng, L. Z., Francis, A., Gould, R., Kim, A. Y., Kretchmar, M., Lu, Q., Moskol, A., Nolan, D., Pelayo, R., Raleigh, S., Sethi, R. J., Sondjaja, M., … Ye, P. (2017). *Curriculum Guidelines for Undergraduate Programs in Data Science*. *4*, 15–30. https://doi.org/10.1146/annurev-statistics-060116-053930

Vygotsky, L. S. (1986). *Thought and Language* (A. Kozulin, Ed.). MIT Press.

Wagh, A., Cook-Whitt, K. & Wilensky, U. (2017). Bridging inquiry-based science and constructionism: Exploring the alignment between students tinkering with code of computational models and goals of inquiry. *Journal of Research in Science Teaching*, *54*(5), 615–641. https://doi.org/10.1002/tea.21379

Wawro, M., Watson, K. & Zandieh, M. (2019). Student understanding of linear combinations of eigenvectors. *ZDM*, *51*(7), 1111–1123. https://doi.org/10.1007/s11858-018-01022-8

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. https://doi.org/10.1007/s10956-015-9581-5

Wiggins, G. & McTighe, J. (2005). *Understanding by Design*. Association for Supervision and Curriculum Development.

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

Yaftian, N. (2015). The Outlook of the Mathematicians' Creative Processes. *Procedia - Social and Behavioral Sciences*, *191*, 2519–2525. https://doi.org/10.1016/j.sbspro.2015.04.617

Zarestky, J., Bigler, M., Brazile, M., Lopes, T. & Bangerth, W. (2022). Reflective Writing Supports Metacognition and Self-regulation in Graduate Computational Science and Engineering. *Computers and Education Open*, *3*, 1–12. https://doi.org/10.1016/j.caeo.2022.100085

APPENDIX A: INITIAL INTERVIEW PROTOCOL

1. Opening
    a. Hi! My name is Sarah Castle and I am a fifth-year graduate student in the Program for Mathematics Education. Just so I have it correct, your name is ______, and is that the correct pronunciation?
    b. Consent form
2. General Background
    a. Please tell me about your current major and if you have any minors?
    b. Describe your previous experience with mathematics and your comfort level?
    c. Describe your previous experience with computing and your comfort level with it?
    d. What are you hoping to do after undergrad?
3. Mathematics and Computation Questions
    a. When I say mathematics – what associations come to mind?
    b. When I say coding – what associations come to mind?
    c. What about computation?
    d. How do you feel when you solve a mathematics problem?
    e. How do you feel when you solve a computational problem?
    f. Does computation or mathematics require more creativity? Why?
    g. What has been the most challenging part of CMSE 201?
    h. What has been the most rewarding part of CMSE 201?
    i. What was the most challenging part of your last math course?
    j. What was the most rewarding part of your last math course?
    k. How do you typically approach a mathematics homework problem?
    l. How do you typically approach a CMSE homework problem?
    m. Are you able to explore within mathematics? Computation?
4. Introduce Surveys and provide links
5. General
    a. What are you hoping to get out of these sessions?
    b. Are there any applications of linear algebra that you would be interested in exploring?
    c. Do you have any other questions for me?

APPENDIX B: FINAL INTERVIEW PROTOCOL

1. Relationship with Disciplines
    a. How do you feel when you solve a mathematics problem?
    b. How do you feel when you solve a computational problem?
    c. Does computation or mathematics require more creativity? Why?
    d. Are you able to explore within mathematics? Computation?
2. Reflection on Modules
    a. What has been the most challenging part of this experience?
    b. What has been the most rewarding part of this experience?
    c. What parts of this experience did you enjoy? Dislike?
    d. What elements do you think helped push your mathematical thinking? Didn't help?
    e. What module helped you learn the mathematical concept and why?
    f. The least helpful?
    g. Computational most and least helpful?
    h. Were you able to explore either within these modules? If so, how?
    i. Were there moments during this experience where you were able to be creative? If so when, and what enabled you?
3. Tasks
    a. After this experience, how would you explain the concept of what a matrix and/or vector is?
    b. After this experience, how do you conceptualize systems of linear equations and approaches to solve them?

c. What does linear independence of vectors mean to you and what is associated with this property?
d. After this experience, how would you explain the concept of span?
e. After this experience, how would you explain the concept of basis?
f. What is an eigenvector and what is an eigenvalue?
g. What is a Markov chain?

APPENDIX C: LINEAR ALGEBRA MODULE OVERVIEW

Table C-1: Mapping of the linear algebra modules with weeks, skills, and objectives.

Week	Linear Algebra Topic	Mathematical Learning Objectives and Skills	Computational Learning Objectives and Coding Skills	Context for Computation
1	**Introduction to Matrices**	Articulate what a matrix is and how to use them to represent systems of linear equations	How to represent a matrix in Python using NumPy	Data Analysis
		Be able to compute matrix multiplication and be able to perform matrix manipulations	How to perform matrix operations in Python	
		Understand algebraic and geometric representations of vectors in Rn and their operations	How to partition a matrix and take slices or access elements of matrix	
2	**Matrix Operations**	Discuss associativity and noncommutativity of matrix multiplication	Determine how images are stored as data in NumPy	Image Manipulation and Photo Filters
		Be able to multiply matrices using NumPy	Use coding to purposefully create digital content	
		Recognize when two matrices can be multiplied	Design, implement, and analyze a computing-based solution to develop a piece of digital art	
		Articulate how to matrix multiplication works		
		Define the transpose of a matrix		
3	**Solving Systems of Linear Equations**	Articulate how to use matrices to represent systems of linear equations	Ability to read pre-existing code, interpret steps, and make needed modifications	Balancing Chemical Equations
		Relate a matrix to a homogeneous system of linear equations		
		Relate an augmented matrix to a system of linear equations	Codify' physical systems and interpret results	
		Interpret solution(s) from echelon form		
4	**Matrix Transformations**	Relate various matrix transformations to geometric illustrations	Reflecting on their thinking and learning in order to transfer to new challenges	Image Manipulation
		Recognize common types of transformations	Engage in conjecture and computational experimentation	
		Interpret a matrix product as a composition of linear transformations		
		Interpret the inverse matrix as representing the inverse linear transformation	Translate mathematical equations into code	
		Distinguish between a matrix as a table of numbers and a linear transformation as a function		
		Define and identify when a function is injective, surjective, and bijective	Demonstrate ability to break apart a problem into smaller parts	
		Define the image of a linear transformation		

Table C-1 (cont'd)

5	**Vector Spaces Part 1**	List examples of subspaces of a vector space	Identify overarching approach for code, and develop a plan prior to coding	Signal Processing and Discrete-Time Signals
		Describe coordinates of a vector relative to a given basis		
		Recognize and use basic properties of subspaces and vector spaces	Ability to learn, practice, and refine approach during the coding process	
		Discuss linear independence for vectors in $\mathbb{R}^n$		
6	**Matrix Determinants**	Provide a definition of the determinant	Engage in conjecture and computational experimentation	Areas and Volumes of Parallelepiped
		Use determinants and their interpretation as volumes	Ability to read pre-existing code, interpret steps, and make needed modifications	
		Analyze the determinant of a product algebraically and geometrically		
		Describe properties of the determinant		
		Describe how the determinant of a matrix and its transpose are related	Implement code to determine the determinant using mathematical equations	
		Describe how the determinant of a matrix and its inverse are related		
		Explain what the determinant measures geometrically		
7	**Vector Spaces Part 2**	Define the dimension of a vector space	Demonstrate an ability to visualize a process in order to accomplish a task in their project	Markov Chains
		Discuss linear independence for vectors in $\mathbb{R}^n$		
		Determine a basis and the dimension of a finite-dimensional space	Demonstrate ability to break apart a problem into smaller parts	
		Define row space and column space of a matrix		
8	**Eigenvectors and Eigenvalues**	Find the eigenvalues and eigenvectors of a matrix	Engage in conjecture and computational experimentation	Cluster Analysis and PCA
		Explain the significance of eigenvectors and eigenvalues	Translate Mathematics into Code to analyze a phenomenon	
		Verify that a given vector is an eigenvector of a matrix.		
		Verify that a scalar is an eigenvalue of a matrix.	Develop a meaningful notebook that fulfills a task	
		Use eigenvectors to represent a linear transformation		

Figure D-1: The Jupyter Notebook for Module 4 which focused on linear transformations and matrices. Note this version is the printout example, but the cells where students would run code are denoted using [].

1 Computational Module 4 - Linear Transformations and Matrices of Linear Transformations

1.1 Name:

1.2 Date:

1.3 Group:

[ ]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import numpy.linalg as la
import sympy as sym
import math
import random as rand
from IPython.display import Image
from IPython.core.display import HTML
sym.init_printing(use_unicode=True)
```

2 Linear Transformations and Matrices of Linear Transforms

During the notebook, we have been working on editing different images and changing their array representation. We will be formalizing some definitions as well as looking at the difference between Matrix Transformations and Linear Transformations.

To start with, run the following code and you should see an image. We will use this as the base image for the next couple of sections

[ ]:
```
# Define some points
x = np.array([0.0, 1, 3, 4, 4, 3, 1, 0,0 ])
y = np.array([0.0, -1,-1,0,6,4,4,6,0])


p = np.matrix([x,y])


sm = p.copy()
```

Figure D-1 (cont'd)

```
#Plot Points
plt.plot([-20,20],[0,0], color='cyan')
plt.plot([0,0], [-20,20], color='cyan')
plt.plot(sm[0,:].tolist()[0],sm[1,:].tolist()[0], color='purple');
plt.grid()
plt.axis('scaled');
plt.axis([-10,15,-10,15]);
plt.title('Starting Image');
```

One of the nice features of the `sympy` module is that we are able to output our matrices and arrays in a way that is visually easier to see, rather than outputting the way in which the computer processes the arrays.

[ ]:
```
sym.Matrix(p)
```

How does this matrix represent the image that is plotted above?

[ ]:
```

```

Put your answer here

What are the dimensions of our matrix p?

[ ]:
```

```

Put your answer here

What are the potential dimensions of matrices we could multiply P with? *Put your answer here*

Let's say that you now wanted to make the cat face slightly wider, but want to keep the same height. How might you want to modify your matrix? Give it a try below!

[ ]:
```
# code goes here
```

Check in with your group at this point and do not proceed until all are at this point

Conjecture Now assume that you wanted to represent how you transformed the cat using a matrix. For example, we want to represent making the cat face wider.

If we represent our points vector as **p**, what size of matrix should we use? Let's call this A, and would we want to calculate $A\mathbf{p}$ or $\mathbf{p}A$ and **why**?

[ ]:
```

```

Put your answer here

Experimentation Let's say that I wanted to double the size of the cat outline. Discuss with your groupmates how you could accomplish this, write out the mathematical expression, and then implement this below. You can use the code above as scaffolding.

Put brainstorming and expression here

Figure D-1 (cont'd)

[ ]:
```
# Put code here
```

Experimentation What if I now wanted a cat outline that was half the size of the original. Discuss with your groupmates how you could accomplish this, write out the mathematical expression, and then implement this below. You can use the code above as scaffolding.

Put brainstorming and expression here

[ ]:
```
# Put code here
```

You have been performing transformations of your original matrix by contracting and expanding the original image.

Below I have included a function that allows you to pass in your points vector and a matrix of your choosing and it will plot the original and your new image.

Based on your discussion during the conjecture and experimentation, modify the line of code to represent the new scaled points vector based on your transformation `A` and your `points` vector

[ ]:
```
def plotTransformation(points, A):
    plt.figure(figsize=(5,5))

    scaled = A@points # Fill in this line

    plt.plot(scaled[0,:].tolist()[0],scaled[1,:].tolist()[0],␣
↪color='green',label='Scaled');
    plt.plot(points[0,:].tolist()[0],points[1,:].tolist()[0],␣
↪color='blue',label='Original');

    plt.grid()
    plt.legend()
    plt.xlabel('$x_1$',fontsize=18)
    plt.ylabel('$x_2$',fontsize=18)
    plt.axis('scaled');
    plt.axis([-10,15,-10,15]);

    plt.axhline(0, color='cadetblue')
    plt.axvline(0, color='cadetblue')
```

[ ]:
```
plotTransformation(p, [[2,0],[0,2]])
```

In the above matrix we have doubled the size of the cat head. Note that the point (0,0) stays the same.

It will now be your turn to explore! See what happens when you try different matrices. Use the format below to document your thoughts or whatever you find easiest! Try and state what you think will happen before you run the code.

Below is one example of how you can record your experimentation, but feel free to use your own system! Add subsequent trials after trial 1 and make sure to label them.

Figure D-1 (cont'd)

2.0.1 Trial 1:

[ ]:
```
check = [[],[]]
print('Transformation Matrix: ')
sym.Matrix(check)
```

What I think it will do: put thoughts here

[ ]:
```
plotTransformation(p, check)
```

2.0.2 Add Subsequent Trials Below

[ ]:
```

```

2.1 Looking at different types of Transformations

There are lots of different types of transformations that we can do using matrix mulitplication and you will now take your findings from above and see if you can generalize the different types of transformations.

Whenever we multiply by the identity matrix, such as this 2×2 matrix:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{1}$$

we get the same image back, because the matrix representing our points stay the same

2.1.1 Reflections

There are multipleways that we can reflect our image. What are the matrices that would correspond to the following reflections? - Reflection across the x_1-axis? - Reflection across the x_2-axis? - Reflection through the line $x_2 = -x_1$? - Reflection through the origin

[ ]:
```
# Code to check
```

2.1.2 Contraction and Expansion

- What are the matrices that would correspond to the being able to horizontally contract and expand? *shrink/stretch*
- What are the matrices that would correspond to the being able to vertically contract and expand? *shrink/stretch*

[ ]:
```
# Code to check
```

2.1.3 Shears

Go to this link and look at the visual of horizontal shear and vertical shear.

- What are the matrices that would correspond to the being able to horizontal shear?
- What are the matrices that would correspond to the being able to vertical shear?

Figure D-1 (cont'd)

```
[ ]: # Code to check
```

Were there any other types of transformations that your groups found interesting?
Put your answer here

3 Transformations

A **transformation** (or function/mapping) T from $\mathbb{R}^n$ to $\mathbb{R}^m$ is a rule that assings to each vector x in $\mathbb{R}^n$ a vector $T(\mathrm{x})$ in $\mathbb{R}^m$.

- $\mathbb{R}^n$: the **domain** of T
- $\mathbb{R}^m$ is called the **codomain** of T
- $T(\mathrm{x})$ in $\mathbb{R}^m$ is called the **image** of x (under the action of T)
- Set of all images $T(\mathrm{x})$ is called the **range** of T

The notation $T : \mathbb{R}^n \to \mathbb{R}^m$ indicates that the domain of T is $\mathbb{R}^n$ and the codomain is $\mathbb{R}^m$.

Computational Connection How can the concept of a function help you understand what a transformation is? If we were to relate the new vocabulary to this coding concept, what would each represent? Can you think of any other analogies that help in your understanding?

Put your thoughts here

```
[ ]: # Put any code here
```

Explain and Apply For the prior transformation where you were modifying your matrix P by multplying by a 2×2 matrix, pick a specific matrix and try applying the definitions of transformations. 1. What was the **transformation**? (ie. what was T) 2. What was the **image** of your x values in your points matrix under the action of T? 3. The points matrix, p, was composed of a vector of x values and y values. Each vector was an element of $\mathbb{R}^n$. What is n in this case? How do you know this?

Put your answer here

```
[ ]: # put any code here
```

```
[ ]:
```

3.1 Linear Transformations

We can denote the mapping associated with matrix multiplication, where for each x in $\mathbb{R}^n$, $T(\mathrm{x})$ is computed as $A\mathrm{x}$ where A is an $m \times n$ matrix, as $\mathrm{x} \mapsto A\mathrm{x}$.

One of the most important classes of transformations in Linear Algebra are *Linear Transformations* and we will define these below:

A transformation (or mapping) T is linear if:

1. $T(\mathrm{u} + \mathrm{v}) = T(\mathrm{u}) + T(\mathrm{v})$ for all **u**,v in the domain of T
2. $T(c\mathrm{u}) = cT(\mathrm{u})$ for all scalars c and all **u** in the domain of T.

Figure D-1 (cont'd)

Linear Transformation - Condition 1 Write a function to test if the first condition of a linear transformation is satisfied:

$T(\mathbf{u}+\mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$ for all **u**,**v** in the domain of T

[ ]:
```
# Put code here
```

Put your thoughts here

Linear Transformation - Condition 2 Write a function to test if the second condition of a linear transformation is satisfied:

$T(c\mathbf{u}) = cT(\mathbf{u})$ for all scalars c and all **u** in the domain of T

[ ]:
```
# Put code here
```

Put your thoughts here

Classification Which of the following matrices from above would be classified as linear transformations? How do you know this?

Put your thoughts here

[ ]:
```
# Put experimentation code here
```

True or False Every matrix transformation is a linear transformation.

Put your thoughts here

[ ]:
```
# Put experimentation code here
```

Affine Transformations An *affine* transformation $T : \mathbb{R}^n \mapsto \mathbb{R}^m$ has the form $T(\mathrm{x}) = A\mathrm{x} + \mathbf{b}$ where A is an $m \times n$ matrix and **b** is in $\mathbb{R}^n$.

Assuming that $\mathrm{b} \neq \mathbf{0}$, your goal is to figure out if an affine transformation a linear transformation? Why or why not?

[ ]:
```
# Put experimentation code here
```

Put your thoughts here

Order of Linear Transformations Is it true that you can apply linear transformations in any order? Why or why not?

[ ]:
```
# Put experimentation code here
```

Put thoughts and answers here

Combining Linear Transformations Is it true that when two linear transformations are performed one after another, the combined effect may not always be a linear transformation? Why or why not?

Figure D-1 (cont'd)

```
[ ]: # Put any experimentation code here
```

Put thoughts and answers here

Reflection How are you currently conceptualizing a transformation? A linear transformation? Are there any coding concepts that help you understand what a transformation is?

Put thoughts and answers here

3.2 Creating your Own Transformations

Now it is your turn! Take about 20 minutes and first design an image, you can use the cat outline as a scaffold. You can use scatter plot if you want to have individual points, or plot if you want connected lines. Then create a series of transformations that alter your original image.

At the end of the time, you will show your original image, and the series of transformations, and then your partners will have to hypothesize what they think the final image will be. (You will show the initial plot, and then the matrices for the transformations, and they will guess what they think the final image will be)

Put thoughts and brainstorms here

```
[ ]: # Put code here
```

Put your prediction of your partner's transformations here

4 Additional Activities

4.1 Exploring Week 2

Go back to your week 2 module with the image manipulation with Loki. Did you use any matrix transformations? If so, were they linear?

Put thoughts here

Now try using matrix transformations to alter your image. What did you need to make sure that the transformation will work?

4.2 Creating Linear Transformations

We will assume that T is a linear transformation. Find the matrix that represents T for the following problems and make sure to demonstrate that it works via code. You can also experiment with code as well. The process is up to you.

1. $T : \mathbb{R}^2 \to \mathbb{R}^2$ first reflects points through the horizontal x_1 axis and then reflects points through the line $x_2 = x_1$
2. $T : \mathbb{R}^2 \to \mathbb{R}^4$, $T(\mathbf{e}_1) = (3, 1, 3, 1)$ and $T(\mathbf{e}_2) = (-5, 2, 0, 0)$ where $\mathbf{e}_1 = (1, 0)$ and $\mathbf{e}_2 = (0, 1)$

Put thoughts here

```
[ ]: # Put Code here
```

Figure D-1 (cont'd)

4.3 Wrap Up

1. What is a linear transformation (in your own words)
2. What are matrix transformations?
3. How can matrix transformations be used outside of the mathematics classroom?

Put your responses here

```
[ ]:
```

APPENDIX E: SURVEY QUESTIONS

Likert Scale [Strongly Disagree/Disagree/Neutral/Agree/Strongly Agree]

Rate you (dis)agreement with the following:

- Mathematics is an innate ability
- Mathematics problems can have multiple solutions
- Mathematics is a collection of rules, formulas, and procedures
- If I don't know how to do a math problem, I will look for a similar problem in my notes or in the textbook
- If I am not able to solve a mathematics problem after my initial attempt, I will go and ask for help
- Mathematics is a creative endeavor
- Mathematics is best taught by direct instruction
- Mathematical creativity is a talent possessed by the most gifted students
- I can take different approaches to given mathematical problems
- I can pose mathematical problems on my own
- I create tools or tricks to help solve mathematics problems
- I can extend my mathematical knowledge to new situations
- When solving a mathematics problem, I look for cues in the problem that are similar to previously worked examples
- I can look at a mathematics problem and think of new ways to solve it, if the assumptions in the problem were changed
- I develop illustrations or representations to help me understand and clarify mathematical concepts

Rate you (dis)agreement with the following:

- Coding is an innate ability
- Computational problems can have multiple solutions
- Computation is a collection of rules, formulas, and procedures
- If I don't know how to do a computational problem, I will look for a similar problem in my notes, previous Jupyter notebooks, or in the textbook
- If I am not able to solve a computational problem after my initial attempt, I will go and ask for help
- Computation is a creative endeavor
- Computation is best taught by direct instruction
- Computational creativity is a talent possessed by the most gifted students
- I can take different approaches to given computational problems
- I can pose computational problems on my own
- I create tools or tricks to help solve computing problems
- I can extend my computational knowledge to new situations
- When solving a computation problem, I look for cues in the problem that are similar to previously worked examples
- I can look at a computation problem and think of new ways to solve it, if the assumptions in the problem were changed
- I develop illustrations or representations to help me understand and clarify computational concepts