HETS Online Journal
Volume 13, Issue 2: May 2023
ISSN 2693-9193

Enhancing Linear Algebra Learning through Computational Thinking: A Project-Based Approach

Prince, Tanvir [1]

[1] Hostos Community College, City University of New York (CUNY),
tprince@hostos.cuny.edu

Abstract

This article discusses the integration of computational thinking concepts, including algorithm, coding, abstraction, decomposition, debugging, and pattern recognition, into a Linear Algebra course in a community college in the fall of 2022. Through the implementation of project-based learning (PBL), we aimed to enhance students' understanding of linear algebra topics while familiarizing them with essential computational thinking concepts. A pre-and post-survey assessed the students' familiarity with the concepts. The results indicated a significant improvement in the students' understanding of computational thinking concepts and their application in linear algebra.

Keywords: computational thinking, education, mathematics, algorithm, coding, abstraction, decomposition, debugging, pattern recognition

Resumen

Este artículo analiza la integración de conceptos de pensamiento computacional, incluidos algoritmos, codificación, abstracción, descomposición, depuración y reconocimiento de patrones, en un curso de álgebra lineal en un colegio comunitario en el otoño de 2022. A través de la implementación del aprendizaje basado en proyectos (PBL), nuestro objetivo fue mejorar la comprensión de los estudiantes de los temas de álgebra lineal mientras los familiarizábamos con los conceptos esenciales del pensamiento computacional. Una pre y una posprueba evaluó la familiaridad de los estudiantes con los conceptos. Los resultados indicaron una mejora significativa en la comprensión de los estudiantes de los conceptos de pensamiento computacional y su aplicación en álgebra lineal.

Palabras clave: pensamiento computacional, educación, matemáticas, algoritmo, codificación, abstracción, descomposición, depuración, reconocimiento de patrones

Introduction

The growing importance of computational thinking in various fields has sparked interest in incorporating computational thinking concepts into traditional mathematics courses, such as linear algebra (Papert, 1996; Wing, 2006). Computational thinking concepts, such as algorithm design, coding, abstraction, decomposition, debugging, and pattern recognition, can help students develop a deeper understanding of the subject matter and prepare them for the computational demands of their future careers (Grover & Pea, 2013; Weintrop et al., 2016).

Previous research has shown that integrating computational thinking concepts into mathematics education can positively impact students' problem-solving skills, critical thinking abilities, and overall academic performance (Lye & Koh, 2014; Yadav et al., 2014). This study aims to contribute to this growing body of research by analyzing the effects of integrating computational thinking concepts into a linear algebra course in a community college. We have employed project-based learning (PBL) to introduce these computational thinking concepts into the curriculum.

Background and Literature Review

There are hundreds of articles and presentations dedicated to the importance of computational thinking in education. It is impossible to summarize all of them, and thus we will very briefly discuss some important ones pertaining to the current topic.

Wing (2006) discusses the importance of computational thinking (CT) as a fundamental skill for problem-solving and asserts that it should be incorporated across various disciplines, including mathematics. The article highlights the benefits of integrating CT in teaching and learning to develop students' problem-solving skills. Grover and Pea (2013) provide a thorough review of computational thinking in K-12 education, discussing its relevance and potential to transform mathematics education. The authors suggest that computational thinking should be an essential component of mathematics curricula, as it promotes higher-order thinking skills and fosters a deeper understanding of mathematical concepts. On the other hand, Lee et al. (2011) explore the application of computational thinking in a practical context, showing how it can help students develop problem-solving skills and enhance their understanding of mathematical concepts. The authors emphasize the value of embedding computational thinking in mathematics instruction to promote a more meaningful learning experience. Yadav et al. (2017) examine various pedagogical approaches to integrate computational thinking (CT) in K-12 education, including mathematics. The authors discuss the benefits of integrating CT in mathematics instruction, such as fostering students' abilities to analyze problems, design solutions, and evaluate outcomes effectively. Gadanidis et al. (2015) explore coding as a tool to teach mathematics in K-12 education. The authors emphasize the potential for coding to engage students in mathematical problem-solving, develop their computational thinking skills, and enhance their understanding of abstract mathematical concepts. Barr and Stephenson (2011) discuss the role of the computer science education community in promoting CT integration in K-12 education, including mathematics. Weintrop and Wilensky (2015) examine students' perceptions of blocks-based programming, a tool often used to introduce CT concepts in mathematics education. Kazimoglu et al. (2012) explore the use of digital game-play to teach programming at the CT level. The authors assert that engaging students in game-based learning can facilitate the development of computational thinking skills and help them apply these skills to mathematical problem-solving. However, there are also many challenges to implementing computational thinking in a classroom setting. Angeli and Giannakos (2016) discuss the challenges associated with implementing computational thinking (CT) education in mathematics classrooms, such as the lack of standardization and curriculum resources. The authors call for a more systematic approach to integrating CT in mathematics education to prepare students for a better digitally driven world. Also, Sentance and Csizmadia (2017) investigate the challenges teachers face in integrating computing and CT into the curriculum, including mathematics education. The authors suggest that teachers require adequate professional development, resources, and support to implement CT effectively in their mathematics classrooms. However, it is rare to find adequate research and articles in community college-level mathematics. One of the goals of this paper is to close this gap.

Methodology

In the fall of 2022, a project-based intervention was introduced in a linear algebra course of a community college to integrate computational thinking concepts, specifically algorithm, coding, abstraction, decomposition, debugging, and pattern. Each of these concepts was introduced and explained during several lectures at the beginning of the semester. These CT concepts were introduced during our regular teaching (both in person and online mode). Students were divided into groups and tasked with selecting a topic in linear algebra to complete a project. The project was required to include the concepts of coding or algorithm and debugging, in addition to two other computational thinking concepts. Some of the examples of the topics for the project are as follows:

1. Solving a 2-by-2 linear system.
2. Finding the intersection of two lines.
3. Find the intersection of three planes.
4. Gauss Jordan elimination for 2 by 3 matrix.
5. Gauss Jordan elimination for 3 by 3 matrix.
6. Find the cross-product of two 3D vectors.
7. Find the determinant of a 3 by 3 matrix by expansion method.
8. Find the determinant of a 3 by 3 matrix by row/column operation.
9. Find the Adjoint of a 3 by 3 matrix.
10. Find the inverse of a 3 by 3 matrix.

11. Find eigenvalues of a 2 by 2 matrix.

12. Apply Cramer's rule for a system of 2 equations and 2 unknowns.

Each group needed to select one topic and explain it through CT concepts. Each group must use “Coding” and “Debugging” and two other CT concepts. A pre- and post-survey were administered to measure students' familiarity with five computational thinking concepts, “coding,” “abstraction,” “decomposition,” “debugging,” and “pattern.” The survey contained questions rated on a scale of 1 to 10, with 1 indicating no familiarity and 10 indicating high confidence. The survey results were analyzed to assess the impact of the project-based intervention on students' understanding of computational thinking concepts.

Results

Twelve students participated in this activity. The results of the pre-survey and post-survey are given below:

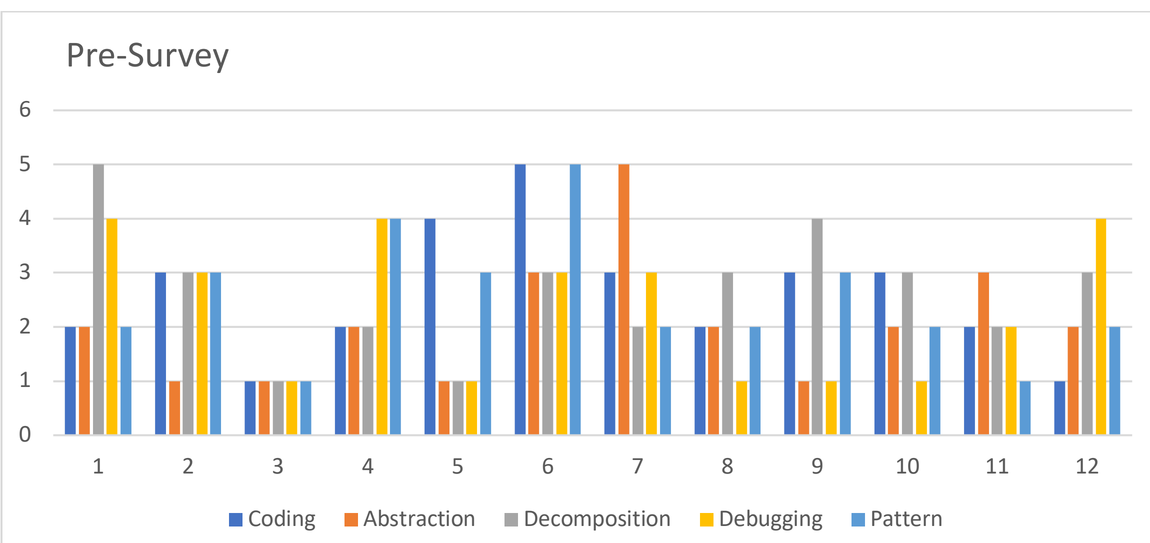

Figure 1

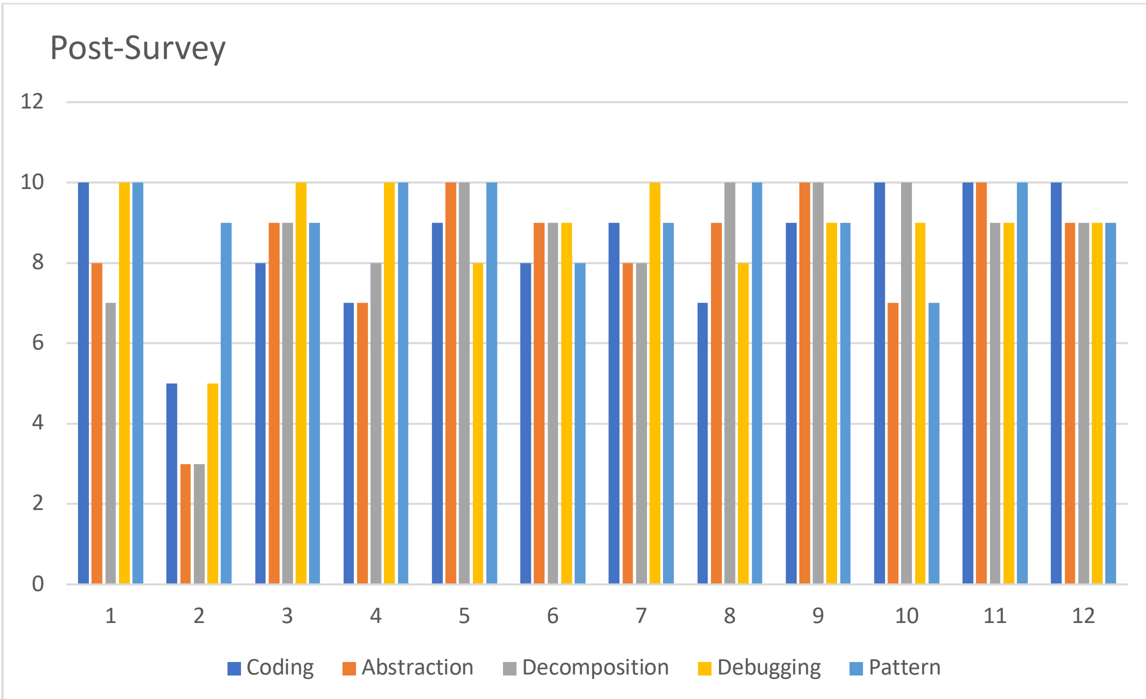

Figure 2

When we look at the pre-survey (figure 1), all the numbers are 5 or below, indicating unfamiliarity with CT concepts. Moreover, after the intervention (Learning CT concepts through project-based learning (PBL)), when we look at the post-survey (figure 2) results, there is a clear indication that almost all the students who participated in the activity learned and understood these critical CT concepts.

This is a small-scale study with only 12 participants. A larger scale study is needed before we can conclude anything scientifically. However, we want to point out that these findings align with prior research on the benefits of integrating computational thinking into mathematics education (Kalelioğlu et al., 2016; Lye & Koh, 2014). The project-based learning approach enabled students not only to learn CT concepts but also to develop critical problem-solving skills, collaboration, and creativity, which are essential in academic and professional settings. This also aligns with previous research findings (Blumenfeld et al., 1991; Bell, 2010).

Conclusion

This study provides evidence that incorporating computational thinking concepts into a linear algebra course through a project-based learning approach significantly enhances students' understanding and familiarity with these concepts. Additionally, it promotes student engagement, motivation, and the development of essential problem-solving skills. Given the increasing importance of computational thinking in various fields, integrating these concepts into mathematics courses is crucial for preparing students for future careers and success in a rapidly changing world.

References

Angeli, C., & Giannakos, M. N. (2016). Computational thinking education: problems and perspectives. International Journal of Child-Computer Interaction, 5, 1-9.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? ACM Inroads, 2(1), 48-54.

Bell, S. (2010). Project-Based Learning for the 21st Century: Skills for the Future. The Clearing House: A Journal of Educational Strategies, Issues and Ideas, 83(2), 39-43.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning. Educational Psychologist, 26(3-4), 369-398.

Gadanidis, G., Hughes, J. M., & Minniti, L. (2015). Coding as a Trojan horse for mathematics education reform. Journal of Computers in Mathematics and Science Teaching, 34(2), 155-173.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. Educational Researcher, 42(1), 38-43.

Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. Baltic Journal of Modern Computing, 4(3), 583-596.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. Procedia Computer Science, 9, 522-531.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... & Werner, L. (2011). Computational thinking for youth in practice. ACM Inroads, 2(1), 32-37.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? Computers in Human Behavior, 41, 51-61.

Papert, S. (1996). An exploration in the space of mathematics educations. International Journal of Computers for Mathematical Learning, 1(1), 95-123.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. Education and Information Technologies, 22(2), 469-495.

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(1), 127-147.

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. Proceedings of the 14th International Conference on Interaction Design and Children, IDC '15, 199-208.

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

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for teacher education. Communications of the ACM, 59(8), 55-62.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education (TOCE), 14(1), 1-16.

Note

This paper is supported by “Computing Integrated Teacher Education.” This is a four-year initiative to support CUNY education faculty at all ranks to integrate state standards-aligned computing and digital literacies content and pedagogy into required education courses, fieldwork, and student teaching. Supported by public funding from the New York City Department of Education (NYC DOE) Computer Science for All (CS4All) program and private funding from the Robin Hood Learning, Technology Fund, Google, and Gotham Gives.