The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #43059

Effects of Integrating Computational Tools into an Introductory Engineering Mechanics Course

Wayne Chang, University of Illinois Urbana-Champaign

Wayne Chang is an assistant teaching professor in the Aerospace Engineering Department at the University of Illinois Urbana-Champaign. He received his BS, MS, and Ph.D. in Mechanical and Aerospace Engineering from the University of California, Irvine. His current engineering education research interests include cross-course teaching tool development, implementation, and integration into curriculums.

Seung Woo Ok, University of Illinois Urbana-Champaign
Prof. Matthew West, University of Illinois Urbana-Champaign

Matthew West is an Associate Professor in the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign. Prior to joining Illinois he was on the faculties of the Department of Aeronautics and Astronautics at Stanfo

Sascha Hilgenfeldt, University of Illinois Urbana-Champaign
Prof. Mariana Silva, University of Illinois Urbana-Champaign

Mariana Silva is a Teaching Associate Professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign. Silva is known for her teaching innovations and educational studies in large-scale assessments and collaborative learning. She has participated in two major overhauls of large courses in the College of Engineering: she played a key role in the re-structure of the three Mechanics courses in the Mechanical Science and Engineering Department, and the creation of the new computational-based linear algebra course, which was fully launched in Summer 2021. Silva research focuses on the use of web-tools for class collaborative activities, and on the development of online learning and assessment tools. Silva is passionate about teaching and improving the classroom experience for both students and instructors.

Effects of Integrating Computational Tools into an Introductory Engineering Mechanics Course

Abstract

Integrating computational tools into engineering education has become pivotal, enhancing students' depth of knowledge and better preparing them for future careers. The Grainger College of Engineering at the University of Illinois Urbana-Champaign has embraced this shift since Fall 2021 by integrating computational Python exercises through Jupyter notebooks into their Statics course, a required course in several degree programs in the college. In each subsequent semester, additional resources were made available to students to bolster the implementation of computational tools. In addition, the course sequence was modified to require students to take a linear algebra course with emphasis on computational tools as a co-requisite or prerequisite for the Statics course. In this paper, we summarize the results of surveys completed by students who have taken or are currently taking the Statics course to identify the impact of these changes. In particular, we defined four different metrics to determine the impact of incorporating computational tools in this course: 1) effectiveness of the computational exercises in building Python skills; 2) students' confidence level in solving statics problems; 3) students' attitude towards the importance of computational tools; and 4) students' satisfaction regarding the revised curriculum. Our survey findings show that students feel their computational skills have improved during the semester, boosting their confidence in using these skills to solve statics problems. Additionally, as the use of computational tools increased throughout the semester, students' satisfaction with the course content increased. Lastly, most students believe acquiring computational skills is important for their post-graduation careers.

1 Introduction

Computational tools have become prevalent and critical in multiple engineering disciplines. One discipline, classical mechanics, in particular, relies heavily on accurate computation by analyzing different forces and torques on physical systems. Computational tools are necessary to prevent unnecessary mistakes when solving problems in classical mechanics. To address the need and demands for computational skills, engineering faculty at the University of Illinois Urbana-Champaign proposed a curriculum change to incorporate computational tools into several classes studying mechanics. These course updates occurred over several semesters, based on analysis of data collected from students, such as surveys and course grades.

There have been numerous precedents that show the effectiveness of computational tools when appropriately incorporated into instruction [1, 2, 3, 4]. These tools improved learning experiences via simulations, where students can visualize physical responses to changing system parameters [5]. Mansbach et al. [6] have shown that students' average grades increase with additional computational tools. Kononov et al. [7] have also demonstrated that students desire greater implementation of computational tools over one course and the entire curriculum, including all course levels. Zhang et al. [8] show similar results as students prefer computational content starting in their first year of education. These works have shown that students gained confidence using computational tools after taking the reformed courses. Furthermore, surveys by Lee et al. [9] found that graduating students are not only more comfortable using computational tools to solve materials science engineering problems but also that more than half of the surveyed students believed that computation had provided them with a better understanding of the course content.

This paper examines the effects of implementing computational tools into the Statics course offered to all Grainger College of Engineering students at the University of Illinois Urbana-Champaign over multiple semesters, starting in Fall 2021. In addition to the curriculum redesign, a new corequisite was imposed for the Statics course to ensure students have the needed background to complete the course successfully. Specifically, students registered in the Statics course are now required to have credit or concurrent registration in a computational linear algebra course, which has an introduction to programming course as a prerequisite.

To measure the effectiveness of computational tools along with the changes in the co/prerequisites, we first give a brief overview of how the student population of the Statics course changed with the addition of the computational linear algebra course as a co-requisite or prerequisite. The change in student population for the introduction to programming course was omitted because the computational linear algebra course is of greater importance when considering computation background for students taking the Statics course. We then investigated the students' comfort levels in using computational tools to solve statics problems each semester. Since there have been consistent changes in how computation is implemented in the course, we also compared survey results to analyze students' opinions on how successfully this course helps develop computational skills and their general views on the importance of computation in their future. We also determined students' satisfaction with the current Statics course following the introduction of computational tools.

2 Course and curricular reform

Prior to Fall 2021, the use of computational tools, such as MATLAB and Python, was suggested to students as the preferred method of solving engineering problems. However, while the engineering curriculum required an introductory programming course, students often waited until their degree program's final year to take it. The discrepancies in student programming capabilities before the Statics course made it challenging for students and instructors to tackle interesting yet complex engineering problems together. Furthermore, although both computational tools and physical calculators were available to students during exams, students who were uncomfortable using computational tools felt at a disadvantage compared to their peers who excelled in computational tools.

The Statics course underwent substantial revision in recent years as our institution recognized the significance of computational skills in modern engineering practice. To improve student computational literacy and have a lasting impact on their academic experience, the reform cannot be an isolated event within a single course. Hence, changes to the prerequisite requirements were also implemented as part of the reformation process. As a result, the proposed curriculum updates began with the development of a computational linear algebra course, with an introduction to a programming course as a prerequisite. This new computational linear algebra course initially served as an optional corequisite to the Statics course (the other option was Calculus 3). However, the curriculum is currently making the computational linear algebra course a core requirement for all students. In a later section, we will show how the percentage of students with this prerequisite requirement has increased over the years.

A significant challenge in this course reform was ensuring the use of computational tools was manageable for the students with additional learning goals added to an already demanding course. To make a smooth transition, the initial course improvement efforts focused on creating student resources and training material for all course staff across every course section. In Fall 2021, the transition to incorporating Python computation in the Statics course began with self-guided tutorials using Jupyter notebooks, a coding platform widely used for education purposes [10]. Furthermore, traditional discussion worksheets given to students during collaborative learning activities in the classroom were transformed into digitized Python activities using the online assessment system PrairieLearn [11, 12].

In this initial implementation, the digitized collaborative learning activities were not graded based on correctness. Instead, students received points for participation only. The lack of grading criteria left the students uncertain about the correctness of their solutions and created a higher volume of questions during office hours and postings on the online message board. Also, it was difficult for course instructors to gauge student understanding during these computational activities without a comprehensive grading rubric.

Building upon the initial implementation in Fall 2022, we supplemented the self-guided Python tutorials with structured workshops led by designated teaching assistants. These workshops provided students with interactive examples and demonstrations, facilitating a deeper understanding of Python concepts. Additionally, auto-grading capabilities were integrated into PrarieLearn, enabling students to receive immediate feedback on their submissions during the collaborative learning activities. The Spring 2023 semester saw continuity in the computational implementation established in Fall 2022.

3 Methods

Anonymous end-of-semester surveys were administered to students enrolled in the Statics course across three semesters: Fall 2021, Fall 2022, and Spring 2023. We aimed to use this data to understand better how the introduction of computational tools impacted students' ability to solve Statics questions and advanced their computational skills. This course is taken primarily by engineering students from various departments. Semester enrollments varied, with approximately 350 students in the fall and around 200 students in the spring semesters.

Survey questions were presented on a 5-point Likert scale using the following options and corresponding numeric values: "Strongly disagree" (1), "Disagree" (2), "Neither agree nor disagree" (3), "Agree" (4), "Strongly agree" (5). Students' scores for each question in each semester were averaged together to obtain the survey response mean score between 1 and 5. The collected data underwent analysis using pairwise t-tests to determine whether the introduction of computational elements and modifications to prerequisites and corequisites yielded statistically significant differences in the average responses [13, 14].

4 Results

In our study, we relied on self-reported data regarding students' status concerning the linear algebra corequisite requirement due to constraints on accessing students' records. Figure 1 illustrates the evolution of the number of students with experience in the computational linear algebra (CLA) course over the observed period. Notably, we observed a significant increase in the proportion of students who either had credit for or were concurrently enrolled in the CLA course, rising from 11% in Fall 2021 to 56% in Spring 2023. When considering credit or concurrent enrollment in an equivalent linear algebra (LA) course, these figures increased from 30% in Fall 2021 to 68% in Spring 2023. Such trends provide valuable insights into the changing landscape of students' preparation in computational linear algebra throughout our study period and how that may impact their perceptions about the course and their computational skill levels.

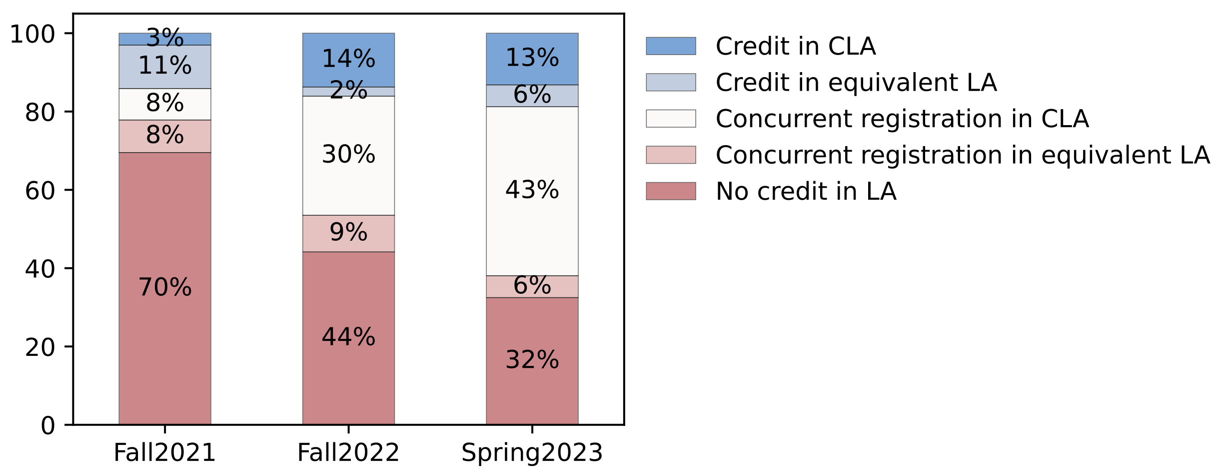

Figure 1: Students' linear algebra prerequisite status at the start of each semester. CLA: computational linear algebra offered at the same institution. LA: any other linear algebra course.

In the following sections, we analyze survey questions regarding students' perceptions of their Python abilities, comfort levels when using computation tools to solve statics problems, attitude towards computation in general following graduation, and satisfaction with the current course resources.

4.1 Developing computational skills

Over 50% of the students are now joining the Statics course with either credit or concurrent registration in the computational linear algebra course. However, we still have a group of students taking the Statics course without any programming experience. We want to know if integrating computational tools into the Statics course can help students develop computational skills; more specifically, we are interested in developing Python proficiency. Students were prompted to respond to the question: *"Q1: This class helped to develop my skills in working with Python."* Figure 2 presents the distribution of responses across the Likert scale options, offering insights into the perceived effectiveness of the course in enhancing students' Python abilities.

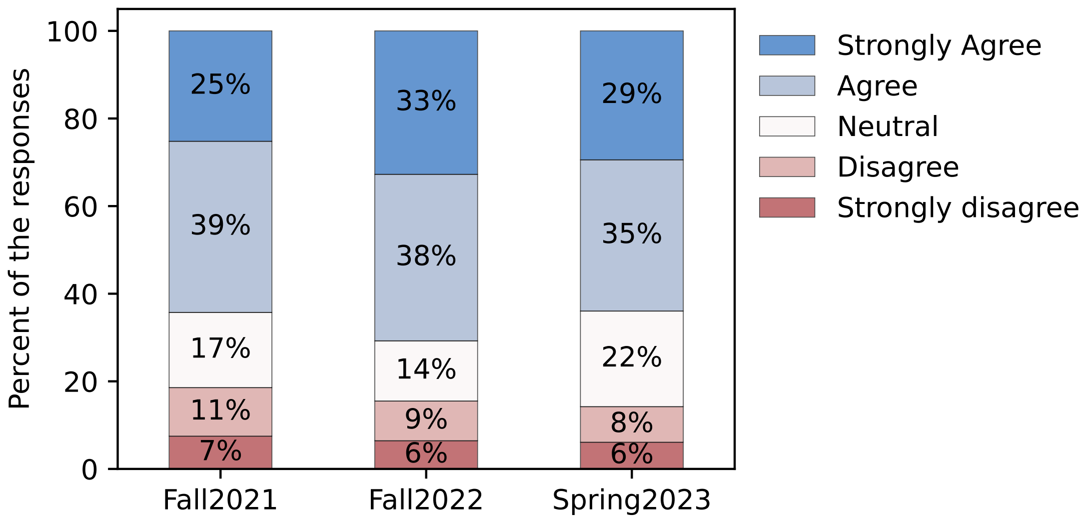

Figure 2: Survey question: "This class helped to develop my skills in working with Python."

Our analysis revealed a notable increase in students expressing confidence in their enhanced computational skills, as indicated by a rise in the mean score from 3.63 in Fall 2021 to 3.82 in Fall 2022, which was statistically significant ($p = 0.042$). However, from Fall 2022 to Spring 2023, while there was a slight decrease in the overall mean score regarding the belief that the course contributed to their Python skill development (from 3.816 to 3.741), the difference was not statistically significant ($p = 0.415$). This lack of significance between Fall 2022 and Spring 2023 is expected, considering no course changes occurred during this period.

These findings indicate that the supplementation of computational elements in the Fall 2022 semester (auto-graders and workshops), along with the increase in the percentage of students possessing programming backgrounds (stemming from increased CLA credit or concurrent registration), had a positive impact on student's perception of their skill development.

4.2 Comfort using computational tools

To measure students' comfort levels with computational tools in solving statics problems, they were asked to respond to the following question: *"Q2: I feel comfortable using computational tools to solve statics problems."*. Figure 3 displays the distribution of responses across the Likert scale options. Notably, over the initial two semesters, we observed a significant increase in the overall comfort level, with the mean score rising from 3.45 in Fall 2021 to 3.84 in Fall 2022 ($p < 0.001$). Another evident trend was the increase in the percentage of students strongly agreeing with the statement, going from 20% to 31% and further to 37% across the three semesters. Our hypothesis suggests that the increasing comfort levels signify that the additional

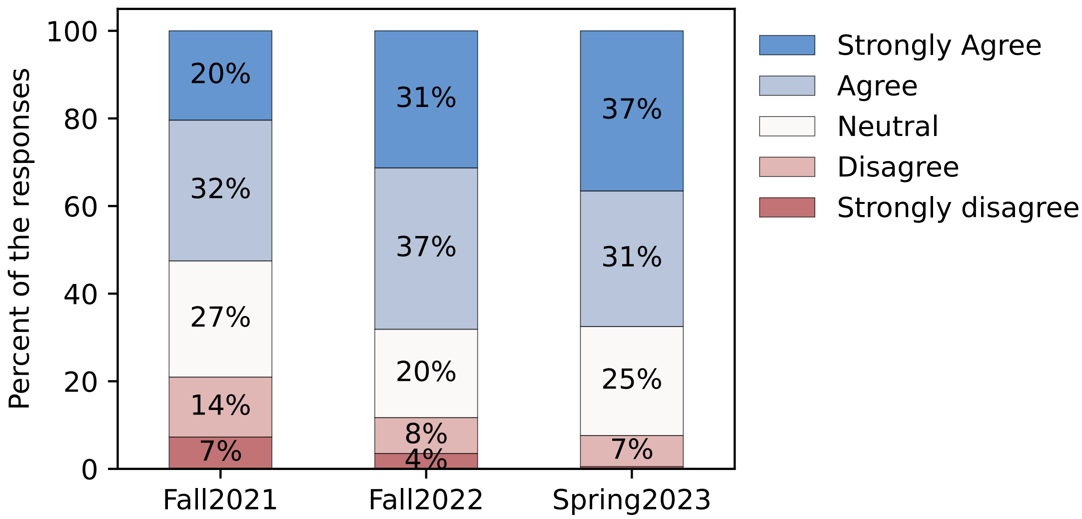

Figure 3: Survey question: "I feel comfortable using computational tools in order to solve statics problems." Percentage labels under 3% are omitted.

Python support via workshops and auto-graders and the programming foundation established through the CLA course enable students to gain confidence in utilizing computation to tackle statics problems.

Additionally, we found no statistically significant change in students' comfort levels from Fall 2022 to Spring 2023, with mean scores of 3.84 and 3.96, respectively ($p = 0.205$). Similar to previous results, this result is consistent with no changes in the course curriculum between these two semesters.

4.3 Attitude towards computational tools

To better understand students' general attitude towards computational skills, we prompted students to answer the following question: *"Q3: I think having computational skills will be important for my post-graduation career"*.

Figure 4 shows that there was a significant increase in the percentage of students who strongly agree with the statement from Fall 2021 to Fall 2022 (48% to 61% with $p < 0.001$). The mean score also experienced a significant increase from 4.27 to 4.49 ($p < 0.001$). The student responses from Fall 2022 to Spring 2023 had a similar distribution, with mean scores of 4.49 and 4.41, respectively ($p = 0.274$). Once again, there is an implication that providing additional Python support and the programming background impacts the students' perception of the importance of computation for their post-graduate careers.

4.4 Satisfaction with the support provided by the course

To assess students' satisfaction with the available course resources supporting the use of computational tools, we posed the question: *"Q4: I need more resources/help to feel comfortable solving systems of equations for this course."*. Figure 5 illustrates a downward trend in the

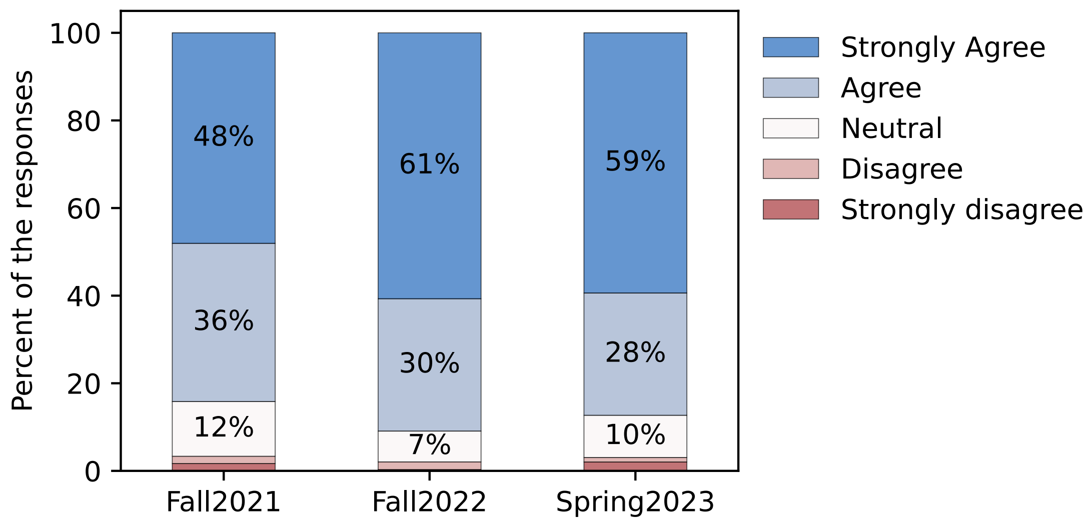

Figure 4: Survey question: "I think having computational skills will be important for my post-graduation career." Percentage labels under 3% are omitted.

percentage of students expressing the need for additional resources and assistance to enhance their proficiency in using computational tools. Specifically, the percentage of students agreeing and strongly agreeing with the statement decreased from 58% to 47% between Fall 2021 and Fall 2022, further declining to 39% in Spring 2023.

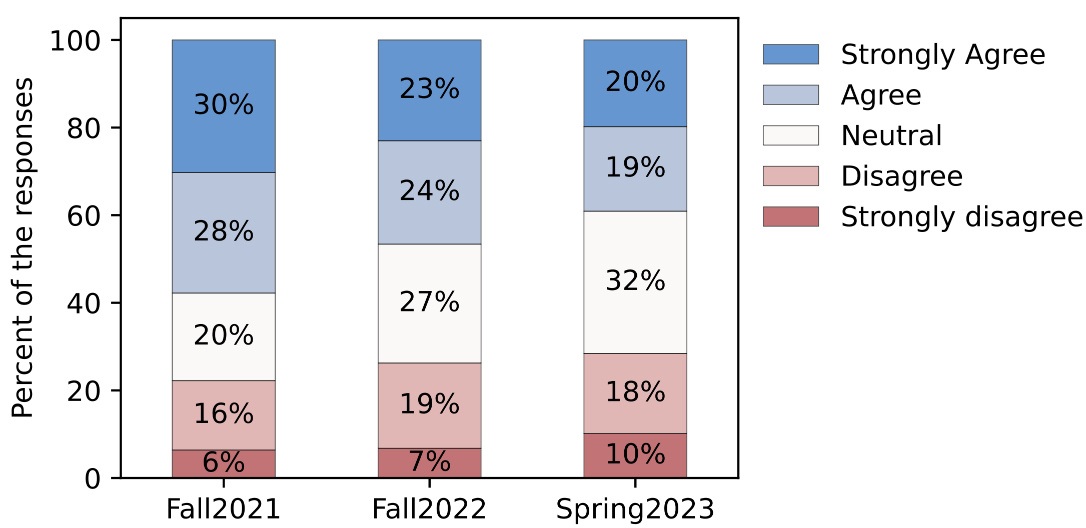

Figure 5: Survey question: "I need more resources/help to feel comfortable solving systems of equations for this course."

Furthermore, we observed a significant decrease in the mean score from 3.60 to 3.36 ($p = 0.014$) from Fall 2021 to Fall 2022. However, the change to a mean score of 3.2 in Spring 2023 was not statistically significant ($p = 0.139$). These findings suggest that students currently perceive a reduced need for additional resources and assistance in utilizing computational tools, indicating that the course changes and the inclusion of the computational linear algebra course as a corequisite have effectively equipped them with a solid foundation to apply computational skills in the Statics course comfortably.

4.5 Regression results

The results from the previous section suggest that the prerequisite change and the introduction of additional Python resources in the Statics course influence students' comfort levels in using computational tools and their perception of preparedness when tackling more complex statics questions. This section aims to separate the effects of the prerequisite requirement and the additional Python resources (auto-graders and workshops) through an ordinary least squares model. For each survey question, we propose the following model:

$$q_i = \mu + \alpha \mathrm{S}_i + \beta_1 \mathrm{LA}_i + \beta_2 \mathrm{CLA}_i \tag{1}$$

where the left-hand-side value q_i is the survey score submitted by student i. S_i is 1 if student i took the Statics course either in Fall 2022 or Spring 2023, otherwise S_i is 0. CLA_i is 1 if student i had credit or was concurrently enrolled in the computational linear algebra course during the semester they took the Statics course, otherwise, CLA_i is 0. And LA_i is 1 if student i had credit or was concurrently enrolled in a linear algebra course during the semester they took the Statics course, otherwise, LA_i is 0. We combined Fall 2022 and Spring 2023 students into one cohort since they all had access to the same course resources and assessments.

The regression parameters μ, α, β_1, and β_2 that we want to estimate can be interpreted as:

- μ: average question response for students in Fall 2021 without credit for linear algebra
- α: effect (additional average question response) due to a student having access to additional Python support via workshops and auto-grader (that is, taking the course in Fall 22 or Spring 2023)
- β_1: effect (additional average question response) due to a student having a "traditional" linear algebra course as co- or pre-requisite
- β_2: effect (additional average question response) due to a student having prior computational experience from taking the computational linear algebra (CLA) course as a co- or pre-requisite

Table 1 shows the coefficients resulting from regressions using survey responses.

Table 1: Coefficients from Eq. 1 obtained for each of the four survey questions.

	Q1	Q2	Q3	Q4
μ	$3.55\ (p < 0.001)$	$3.33\ (p < 0.001)$	$4.26\ (p < 0.001)$	$3.70\ (p < 0.001)$
α	$0.14\ (p = 0.105)$	$0.26\ (p = 0.001)$	$0.15\ (p = 0.010)$	$-0.15\ (p = 0.092)$
β_1	$0.38\ (p = 0.001)$	$0.28\ (p = 0.007)$	$-0.01\ (p = 0.893)$	$-0.30\ (p = 0.013)$
β_2	$0.11\ (p = 0.248)$	$0.53\ (p < 0.001)$	$0.09\ (p = 0.138)$	$-0.43\ (p < 0.001)$

The most significant impact from the introduction of additional Python resources in the Statics course was found in *"Q2: I feel comfortable using computational tools in order to solve statics problems"*. Students who took the course with more available resources had a statistically significant 7.8% increase in their average reported comfort value (a change of $\alpha = 0.26$ on a 5-point scale from an average of 3.33, $p = 0.001$). Moreover, students who took the computational linear algebra course had an additional 16% increase in their average reported comfort value, compared to an 8.4% increase if they took a traditional linear algebra course ($\beta_2 = 0.53$ and $\beta_1 = 0.28$ respectively, on a 5-point scale from an average of 3.33, $p < 0.001$ and $p = 0.007$). This result is unsurprising since students were exposed to Python in the computational linear algebra course.

Question *"Q3: I think having computational skills will be important for my post-graduation career"* had the highest average response for the baseline students (Fall 2021 without credit in linear algebra). Students who had access to more available resources had a small but statistically significant 3.5% increase in their average perception of the importance of computational skills (a change of $\alpha = 0.15$ on a 5-point scale from an average of 4.26, $p = 0.010$). Our results also suggest no notable impact on student's perception of the importance of computational skills when they have the credit or concurrent registration in a linear algebra course.

Regression results for *"Q4: I need more resources/help to feel comfortable solving systems of equations for this course"* revealed that introducing auto-graders and workshops decreased the perception of needing more resources. However, this effect was small and not significant (a change of $\alpha = -0.15$ on a 5-point scale from an average of 3.70, $p = 0.092$). However, having credit in a linear algebra course significantly reduced the perception of needing additional resources to solve a system of equations since this is a topic covered in traditional linear algebra courses. Moreover, having credit in the computational linear algebra course had a more significant effect than credit in traditional linear algebra courses (11.6% decrease in their average reported perception of needing more resources, compared to an 8.1% decrease, respectively).

Finally, the regression analysis for *"Q1: This class helped to develop my skills in working with Python"* reveals that students who had credit or concurrent registration in a tradition linear algebra course experienced the most improvement in their Python skills (a change of $\beta_1 = 0.38$ on a 5-point scale from an average of 3.55, or the equivalent of a 10.7% increase, $p = 0.001$), likely because this was their first time using this programming language.

5 Conclusion

The integration of computational elements into the Statics course represents a relatively recent development. Nonetheless, our findings underscore the positive and significant impact on students' learning experiences resulting from supplementing computational resources and including the computational linear algebra course as a corequisite. Specifically, students saw the value of computational skills in their future career preparedness through the intervention introduced in this study. Other positive outcomes in student learning were determined to be the results of the combination of changes in the corequisite requirement in the degree program curriculum and the introduction of additional computational resources.

Our study explored the impact of increased computational tools and resources in temporal trends such as student comfort and confidence with computation, attitudes toward computation's importance post-graduation, and overall satisfaction with course resources. With the incorporation of the computational linear algebra corequisite and the continual refinement of computational integration in the Statics course, students have reported enhanced effectiveness in developing computational skills. Moreover, their confidence and comfort levels with computation have grown, alongside an increasing recognition of computation's significance in their future careers.

However, to obtain more concrete insights, future research may involve surveying alums to assess the lasting impact of computational learning during their undergraduate studies. Additionally, students' heightened satisfaction with the course's computational aspect suggests that further augmenting computational resources could bolster satisfaction levels even more. Moreover, while our study relied on end-of-semester surveys, including multiple surveys throughout the semester would offer a more comprehensive dataset for analysis. To address this, we have initiated the collection of multiple student surveys throughout the semester and will report on our findings in future research publications.

References

[1] L. Rubin H., P. Manuel J., and B. Cristian C., *Computational Physics: Problem Solving with Python.* Wiley-VCH, 2007, vol. Second revised and enlarged edition.

[2] G. M. Lu, D. R. Trinkle, A. Schleife, C. Leal, J. Krogstad, R. Maass, P. Bellon, P. Y. Huang, N. H. Perry, M. West, T. Bretl, and G. L. Herman, "Impact of integrating computation into undergraduate curriculum: New modules and long-term trends," in *2020 ASEE Virtual Annual Conference*, 2020.

[3] A. D. Santo, J. C. Farah, M. L. Martinez, A. Moro, K. Bergram, A. K. Purohit, P. Felber, D. Gillet, and A. Holzer, "Promoting computational thinking skills in non-computer-science students: Gamifying computational notebooks to increase student engagement," *IEEE Transactions on Learning Technologies*, vol. 15, no. 3, p. 392–405, 2022.

[4] A. Ciobanu, C. Miron, C. Berlic, and V. Barna, "Integrating computational tools in teaching electromagnetic oscillations," *Romanian Reports in Physics*, vol. 75, no. 4, 2023.

[5] Y. Rodríguez, A. Santana, and L. M. Mendoza, "Physics education through computational tools: the case of geometrical and physical optics," *Physics Education*, vol. 48, no. 5, pp. 621–628, 2013.

[6] R. Mansbach, A. Ferguson, K. Kilian, J. Krogstad, C. Leal, A. Schleife, D. R. Trinkle, M. West, and G. L. Herman, "Reforming an undergraduate materials science curriculum with computational modules," *Journal of Materials Education*, vol. 38, p. 161–174, 2019.

[7] A. Kononov, P. Bellon, T. Bretl, A. L. Ferguson, G. L. Herman, K. A. Kilian, J. A. Krogstad, C. Leal, R. Maass, A. Schleife, J. K. Shang, D. R. Trinkle, and M. West, "Computational curriculum for MatSE undergraduates," in *2017 ASEE Annual Conference & Exposition*, 2017.

[8] X. Zhang, A. Schleife, A. Ferguson, P. Bellon, T. Bretl, G. L. Herman, J. A. Krogstad, R. Maass, C. Leal, D. R. Trinkle, and M. West, "Computational curriculum for MatSE undergraduates and the influence on senior classes," in *2018 ASEE Annual Conference & Exposition*, 2018.

[9] C.-W. Lee, A. Schleife, D. R. Trinkle, J. A. Krogstad, R. Maass, P. Bellon, J. K. Shang, C. Leal, M. West, T. Bretl, G. L. Herman, and S. Tang, "Impact of computational curricular reform on non-participating undergraduate courses: Student and faculty perspective," in *2019 ASEE Annual Conference & Exposition*, 2019.

[10] J. Wagemann, F. Fierli, S. Mantovani, S. Siemen, B. Seeger, and J. Bendix, "Five guiding principles to make Jupyter notebooks fit for earth observation data education," *Remote Sensing*, vol. 14, no. 14, 2022.

[11] M. West, G. L. Herman, and C. Zilles, "PrairieLearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning," in *2015 ASEE Annual Conference & Exposition*, 2015.

[12] M. West, N. Walters, M. Silva, T. Bretl, and C. Zilles, "Integrating diverse learning tools using the PrairieLearn platform," in *Seventh SPLICE Workshop at SIGCSE*, 2021.

[13] S. E. Harpe, "How to analyze Likert and other rating scale data," *Currents in Pharmacy Teaching and Learning*, vol. 7, no. 6, pp. 836–850, 2015.

[14] G. M. Sullivan and J. Artino, Anthony R., "Analyzing and Interpreting Data From Likert-Type Scales," *Journal of Graduate Medical Education*, vol. 5, no. 4, pp. 541–542, 12 2013.