2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #39157

Differences Between First- and Third-Year Students' Attitudes Toward Computational Methods in Engineering (WIP)

Nina Perry
Dr. Timothy Chambers, University of Michigan

Dr. Chambers is a Lecturer in Materials Science & Engineering at the University of Michigan. He teaches advanced laboratory courses in MSE as well as introduction to engineering.

Abstract

This Work-In-Progress study investigates differences in freshman and junior engineering students' valuation of and self-efficacy for computational work in engineering. We administered a survey to N=58 total students and performed a mixed-methods analysis to better understand what factors may influence students' attitudes in this area. We found that freshmen's intended major (CS or non-CS) was strongly correlated to differences in their response patterns across survey items. Interestingly, while MSE juniors had significantly higher self-efficacy scores for computational work than those of freshmen, their valuation scores were slightly lower than those of freshmen, despite their much greater experience in the area. We are currently conducting and analyzing follow-up interviews with survey participants to investigate the causes of these outcomes.

Introduction

Programming and simulation skills are frequently undervalued by students in engineering disciplines that are not perceived by novice learners as computational in nature. Previous research indicates that students majoring in subjects that are not programming-heavy might think they will not need these skills in their careers, or they are less capable [1]. However, both students and professionals across different engineering disciplines commonly accept that diversifying one's skill set makes one more marketable and favorably positioned for career advancement [2][3]. Additionally, studies suggest that materials science and engineering (MSE) faculty favor incorporating computational tools into their teaching and think that computation is an essential component of the curriculum [4]. However, more research is necessary to understand how students appreciate these tools or if they perceive a need for them.

This research builds upon earlier work that aimed to identify the distinctions that exist in programming-related motivational factors for first-year engineering students [5]; here we investigate whether previous findings about student perceptions regarding computational skills are generalizable across multiple institutions. The motivational factors examined in the previous study, which served as a framework for this one, include self-efficacy, expectancy value, and utility value as research indicates that these might have an impact on student learning, academic achievement, and career aspirations [6]. We also expand upon prior work to examine differences in perspectives regarding simulation tools in addition to programming. Here, we analyze survey data to compare the attitudes of undeclared freshmen and juniors who have declared MSE. This will allow us to investigate how attitudes regarding these tools change over time as a result of academic and extracurricular activities.

The primary questions driving this study include:

1) Do MSE juniors have different valuations of self-efficacy for programming versus simulation?
2) Do MSE juniors value programming differently than freshmen?
3) Do MSE juniors believe their experiences have adequately prepared them to use computational tools?

Ultimately, understanding engineering students' perspectives on computation and the experiences that shape their attitudes can guide educators in their teaching to aid students in recognizing the importance of computational skills and feel more confident in using them.

Methodology

The study methodology centers on delivery and analysis of a survey for freshman (N= 29) and junior (N= 29) students at a large Midwestern research university. The study was approved by the IRB before the Fall 2022 semester, which is when we performed data collection. During Fall 2022, we recruited freshmen study participants from the Introduction to Engineering course, which is predominantly taken by first-year students who have not yet declared their major. We recruited junior participants from the Materials Laboratory I course, which is an advanced lab course predominantly taken by third-year students who have declared MSE as their major. Students were offered a small extra credit bonus to their course grades to incentivize participation in the study; students who did not opt-in to the study were allowed to complete a short essay assignment for an equal amount of extra credit.

The survey was delivered via Qualtrics, and was open to students for the last two weeks of the semester. The response rate for freshmen was 81% and the response rate for juniors was 91%, so we are confident that our participant group is representative of the entire classes.

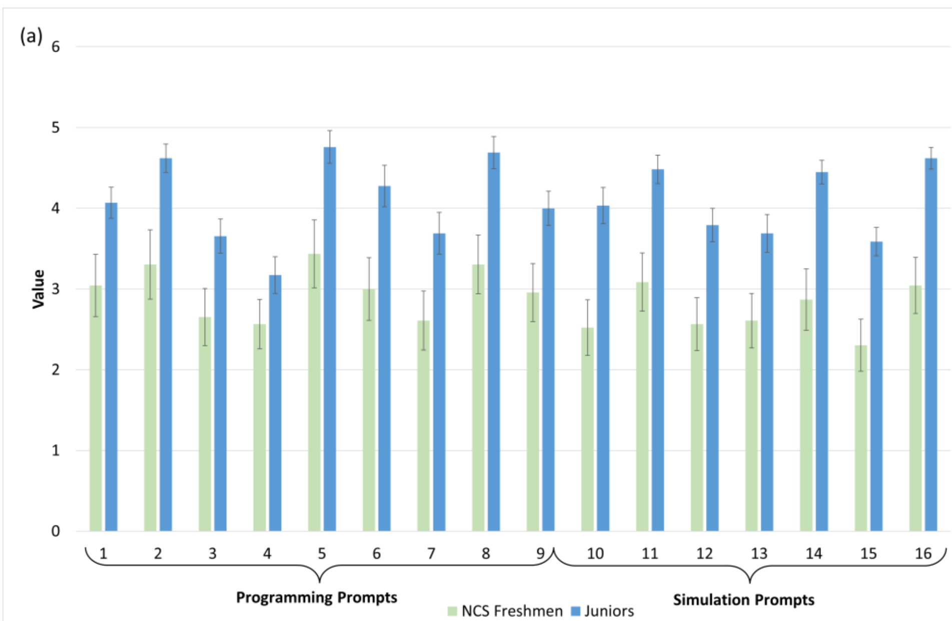
We employed a mixed-methods approach to data analysis, combining a quantitative analysis of student responses to Likert-scale survey items with qualitative analysis of student-generated text responses to short writing prompts embedded in the survey. Our qualitative analysis consisted of a simple consensus-coding scheme that grouped students' written item responses into categories such as "low confidence" or "high valuation" or "no experience" and counting the frequency of different codes. As we are currently conducting follow-up interviews to gain a richer understanding of students' experiences and motivations, the qualitative analysis largely served to inform the interview protocols by highlighting areas that require a more in-depth investigation.

Because we expected the intended major of freshmen students to be an important factor in their beliefs and attitudes about computational work in engineering, we separated the freshmen into two groups: CS, meaning they indicated computer science as their first or second major choice on the survey, and NCS for others. We then calculated the point-biserial correlation coefficient between intended major (CS or NCS) and Likert-scale response for each survey item. Large correlation coefficients ($r_{pb} \geq 0.5$) indicated that major choice may be an important explanatory variable driving response patterns on those items.

To determine the statistical significance of differences in Likert-scale responses between participant groups, we used two-sample t-tests with a significance threshold of $\alpha = 0.05$. Effect size of statistically significant differences was quantified using Cohen's *d*. We performed the following pairwise comparisons: juniors vs. CS freshmen, juniors vs. NCS freshmen, and juniors vs. all freshmen. Additionally, we compared juniors' responses about simulation to their responses about programming to quantify differences in their valuation of and confidence with respect to these two distinct categories of computational engineering work. Qualitatively, we then looked for trends in their self-described experiences with computational work to inform future interviews on the topic.

Results

Quantitatively, we found for freshmen, intent to declare CS strongly correlated with both desire to pursue a career in programming ($r = 0.65$) as well as a higher self-perceived competency in programming in response to six of the eight questions relating to self-efficacy ($0.51 \leq r \leq 0.60$). We observed CS freshmen responded with lower mean valuation and self-efficacy ratings compared to that of juniors regarding simulation. This supports our hypothesis that major is an explanatory variable for some of our observed statistical outcomes. Comparing just NCS freshmen to juniors, we found a statistically significant difference in the mean value responses for all programming and simulation questions related to self-efficacy, indicating that freshmen generally exhibited lower confidence in their abilities compared to juniors. The average NCS freshman responses to all nine programming self-efficacy questions were lower than those of the juniors (see Fig. 1a). For simulation self-efficacy questions, the average freshmen responses were also lower compared to that of the juniors (see Fig. 1a). We reasonably expect that most juniors have more practice and experience with computational tools than freshmen, and therefore would have higher confidence in their abilities.



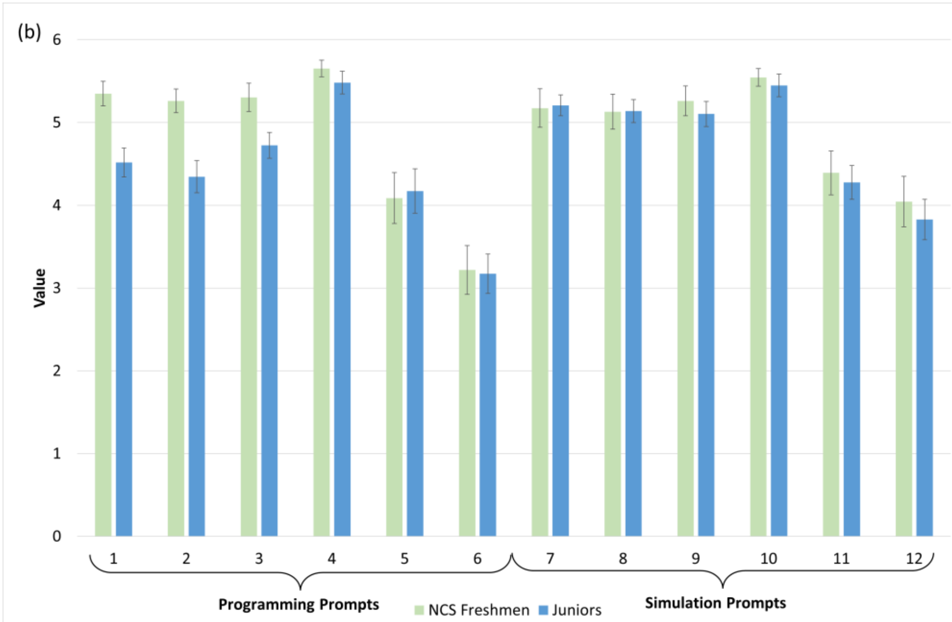


Fig. 1. Plots depicting comparisons between NCS freshman and junior mean responses regarding programming/simulation self-efficacy (a) and valuation (b). For all questions relating to self-efficacy, the Likert scale translates to 1 = "Not at all confident" to 6 = "Extremely Confident." For self-efficacy questions, the scale translates to 1 = "Strongly Disagree" to 6 = "Strongly Agree."

Our findings revealed a surprising similarity in NCS freshmen and junior mean responses related to motivation and ability to strategize for programming and simulation-related projects ($0.26 \leq p \leq 0.79$). Despite having a lower perceived competence in these areas, freshmen were more confident in their capacity to manage their time and plan for computational assignments. Our initial hypothesis was that lower self-efficacy would lead to a lower perceived ability to plan and execute programming and simulation tasks, especially considering many NCS freshmen indicated having little-to-no experience working with these tools. However, we believe that the freshmen students ranked their confidence higher due to a belief that their abilities to strategize and efficiently manage their time for other disciplines would transfer to computational tasks, even though they didn't have high confidence in their abilities to complete them.

Additionally, we noticed that there was no significant difference in the means comparing the NCS freshmen's and juniors' valuations of simulation tools (see Figure 1b.). We expected that juniors, who have more experience using simulations and likely have a better understanding of its utility and capabilities within MSE, would have provided a higher mean rating.

Regardless of freshmen's major, juniors valued simulation more than programming compared to freshmen, suggesting that more juniors would likely agree that knowledge of simulation skills is more relevant to their careers. Additionally, in just comparing juniors' responses to programming and simulation, we found a statistically significant difference in mean response for three out of the four questions relating to value between these tools ($0.00 \leq p \leq 0.03$). We believe juniors may have acquired a deeper appreciation towards simulation tools than programming tools because simulation tools are tightly integrated into the MSE lab courses at the study institution, while programming assignments are given only sporadically in the core MSE courses. However, we are examining this idea more deeply through student interviews because experiential learning through academic research or internships may also have an impact on these findings.

Our qualitative analysis of the free response questions revealed significant differences between the NCS freshman and junior students' perceptions of simulation and programming. While many NCS freshmen claimed they had less expertise and were less confident in programming, many indicated they had no experience with regards to simulation. On the other hand, juniors reported that their experience was limited with both programming and simulation, and many of them used words like "shallow" or "narrow" to indicate that, despite their experience, their level of expertise was rather limited.

Specifically within the junior dataset, we expected to see mean scores as well as written responses indicating higher self-efficacy than what was actually observed in both programming and simulation tools. However, we think that juniors who have presumably used computational tools more often than freshman have a greater awareness of the breadth of knowledge and experience needed to be proficient with these techniques. This may be an instance of the Dunning-Kruger effect, with novices overestimating their abilities while those with more awareness of what is required to achieve expertise underestimate their own competence [7]. Considering this, we believe juniors might have given themselves a lower competency rating since they understood that proficiency requires continual learning and improvement.

Implications & Conclusions

In this study, we compared quantitative and qualitative survey data for engineering freshmen and juniors to understand differences in their attitudes toward programming and simulation work in engineering. Overall, we found significant correlations between intended major and attitudes, with CS-oriented freshmen having much higher valuation of programming in engineering than MSE majors. Non-CS-oriented freshmen also had a greater valuation of programming than MSE majors, but neither as consistently (across survey items) nor as strongly as CS freshmen.

Interestingly, we found no significant difference between NCS freshmen and MSE juniors in their valuation of simulation. Given that the juniors have much more academic experience with simulation tools, and are better informed about their career paths, one might expect that they would value simulation more highly than the freshmen, but this is not the case. As such, we plan to focus on this point in interviews to understand if there are competing explanatory factors leading to a zero net change in valuation between the two populations.

As the institution being studied, the junior-level MSE lab courses have robust computational modeling and simulation curricular content. Our findings therefore suggest a strong positive impact that frequent use of simulation tools in MSE courses can have on students' attitudes toward these tools in the context of engineering work. However, because we did not directly measure students' actual competency, but only their self-efficacy, it is not clear whether their lack of confidence with these tools accurately reflects a low level of proficiency or whether it reflects a greater level of appreciation of the complexity of these tools, which novices would not appreciate. It would be valuable for a future study to examine the relationship between actual proficiency and self-efficacy in this context. Extrapolating from the above, we expect that incorporating simulation tools into engineering curriculum earlier in the sequence should have additional positive effects on students' attitudes as well as their actual proficiency, and we recommend this course of action to any instructors/programs seeking to improve their students' attitudes toward simulation in MSE.

Finally, there is the question of whether MSE juniors feel adequately prepared with computational tools & skills to begin their careers. For all four preparedness items, the mean response was approximately 4 (somewhat agree), with students responding slightly more positively about simulation than programming and slightly more positively about their combined experiences than just coursework. That said, we found no statistically significant difference between mean responses among the four items; it is unclear whether this reflects a true lack of difference or whether it is a consequence of the statistical power limitations of a small study. Regardless, the lack of strong positive feeling about preparedness indicates a need to better engage students with respect to how computational skills are valuable to their professional growth and what an appropriate level of proficiency looks like at the beginning of one's career.

Our ongoing work focuses on follow-up interviews with the survey participants to better understand how their specific experiences both in and out of the classroom have affected their attitudes toward computational skills and tools. We expect that the interview results will provide additional insights into the causal mechanisms driving the survey results and thereby inform curricular improvements to support MSE students' development and appreciation of computational skills.

Acknowledgements

The authors would like to thank Dr. Alison Polasik for her support on this project. We also wish to acknowledge the NSF for sponsoring this research.

References

[1] A. K. Polasik and D. Riegner, "Successes and lessons learned in an undergraduate computational lab sequence for materials science and engineering," in *Proc. 2017 ASEE Annu. Conf. & Expo.*, 2017.

[2] A. Ritchie, "The Impact of Creating the Next-Generation Materials Genome Initiative Workforce," *Jom*, vol. 72, (7), pp. 2466-2468, 2020, doi: 10.1007/s11837-020-04230-4.

[3] A. J. Fletcher, A. A. Sharif, M. D. Haw, "Using the perceptions of chemical engineering students and graduates to develop employability skills," *Educ. Chem. Eng.*, vol. 18, pp. 11-25, 2017, doi: 10.1016/j.ece.2016.07.001.

[4] R. A. Enrique, M. Asta and K. Thornton, "Computational materials science and engineering education: An updated survey of trends and needs," *Jom*, vol. 70, (9), pp. 1644-1651, 2018, doi: 10.1007/s11837-018-2989-7.

[5] A. K. Polasik, A. Suggs, R. Kajfez, "Work in Progress: A Study of Variations in Motivation and Efficacy for Computational Modeling in First-year Engineering Students," in *2021 IEEE FIE*, 2021, pp.1-6, doi: 10.1109/FIE49875.2021.9637463.

[6] R. W. Lent, F. G. Lopez and K. J. Bieschke, "Mathematics self-efficacy: Sources and relation to science-based career choice." *J. Couns. Psychol.*, vol. 38, (4), pp. 424, 1991, doi: 10.1037/0022-0167.38.4.424.

[7] J. Kruger and D. Dunning, “Unskilled and Unaware of It: How Difficulties in Recognizing One’s Own Incompetence Lead to Inflated Self-Assessments,” *J. Pers. Soc. Psychol.*, vol. 77, no. 6, pp. 1121–1134, 1999.

Appendix A

A.1 Self-efficacy prompts with responses on a 1-6 Likert scale (“not at all confident” to “extremely confident”). Questions 1-9 are programming-related and questions 10-16 are simulation-related.

1. Write syntactically correct code (where there are no errors that prevent the code from running).
2. Read and understand the structure of computer code that contains appropriate comments included by the writer.
3. Read and understand the structure of computer code that does not contain appropriate comments.
4. Read provided computer code that does not contain appropriate comments and identify errors in the code.
5. Write a small programming script (5-25 lines) to solve a simple problem that is familiar to me.
6. Write a medium sized programming script (40-100 lines) to solve a problem that is familiar to me.
7. Write a long programming script (more than 120 lines) with nested commands (for example, calculations within a for loop) to solve a problem that is familiar to me.
8. Make use of a pre-written program that is provided to me, making minor modifications as necessary.
9. Debug (correct all the errors) as I write my program.
10. Create a computer simulation that runs successfully (where there are no errors that prevent the simulation from finishing and finding a solution).
11. Understand the structure and purpose of a computer simulation that contains appropriate comments and/or documentation included by the simulation developer.
12. Understand the structure and purposes of a computer simulation that does not contain appropriate comments and/or documentation.
13. Create a physically accurate computer simulation (where the simulation will find a correct solution to model the physical system as intended).
14. Create a simple simulation (applying 1-2 physics equations to a simple geometry) to solve a problem I am familiar with.
15. Create a more complex simulation (applying more than 2 physics equations and/or having multiple geometry elements) to solve a problem that I am familiar with.
16. Make use of a pre-built simulation that is provided to me, making minor modifications as necessary.

A.2 Value prompts with responses on a 1-6 Likert scale (“strongly disagree” to “strongly agree”). Questions 1-6 are programming-related and questions 7-12 are simulation-related.

1. In order to successfully complete my engineering degree, I will need to develop the skills to use programming tools.
2. In order to successfully complete my engineering degree, it is important that I learn how to write computer code and programs.
3. To be a successful engineer, I will need to develop the skills to use programming tools to solve problems
4. Developing skills with programming tools will offer me a wider range of employment options.

5. I enjoyed learning to work with programming tools.
6. I would like to have a career that requires me to use programming tools frequently.
7. In order to successfully complete my engineering degree, I will need to develop the skills to use simulation tools.
8. In order to successfully complete my engineering degree, it is important that I learn how to build and develop computer simulations.
9. To be a successful engineer, I will need to develop the skill to use simulation tools to solve problems.
10. Developing skills with simulation tools will offer me a wider range of employment options.
11. I enjoy learning to work with simulation tools.
12. I would like to have a career that requires me to use simulation tools frequently.

A.3 Free response questions to allow for more detailed and open-ended answers. Questions 3 and 4 were only asked in the junior survey.

1. What are three words you would use to describe your experience in learning programming?
2. What are three words you would use to describe your experience in learning simulations?
3. What experiences have been the most beneficial for the development of your programming skills (e.g. programming-based classes, lab classes, extracurriculars, etc.)? Please be as specific as possible.
4. What experiences have been the most beneficial for the development of your simulation skills (e.g. simulation-based classes, lab classes, extracurriculars, etc.)? Please be as specific as possible.