WIP: Supporting Student Understanding of Finite Element Analysis and Computational Science: Classroom Scaffolding and ChatGPT

Jose L. De La Hoz
Department of Education
Universidad del Norte
Barranquilla, Colombia
vegalj@uninorte.edu.co

David Restrepo
Department of Mechanical Engineering
The University of Texas at San Antonio
San Antonio, Texas
david.restrepo@utsa.edu

Camilo Vieira
Department of Education
Universidad del Norte
Barranquilla, Colombia
cvieira@uninorte.edu.co

***Abstract*—This work-in-progress research paper presents the preliminary results of a study exploring the effectiveness of using computational notebooks to enhance student learning in a Finite Element Analysis (FEA) course for undergraduate Mechanical Engineering students. Our previous work has shown that students often face difficulties in grasping abstract concepts from mechanics of materials while simultaneously learning computational modeling. However, students recognized several advantages of using MATLAB for FEA compared to manual calculations, including significant time savings, increased efficiency, and reduced errors. Nevertheless, they also faced challenges, including a steep learning curve for MATLAB and concerns about how this limitation hinders their conceptual understanding. Despite these drawbacks, they recognized the importance and value of developing computational skills for their future careers. In this study, we extended the scaffolds and changed the sequence of activities to address the challenges the students faced in the previous iteration of our work. Specifically, we provided worked examples that students needed to use, self-explain, and modify before they engaged in programming from scratch. Also, after developing a basic understanding of how to implement FEA in MATLAB, the students used ChatGPT to generate a code that would do the same task. This activity required them to evaluate and refine automatically generated MATLAB code, as ChatGPT may provide alternative solutions that might not always work correctly. We explore three main topics to understand student experiences with and perceptions of this approach: (1) the value of using computational methods compared to manual completion for FEA; (2) the challenges and support of using MATLAB for FEA; and (3) the effectiveness of simulation tools to learn FEA. The goal of this project is two-fold: (1) supporting student learning of intricate phenomena explored in mechanics of materials, like distribution of stress, and stiffness, and (2) fostering essential computational thinking skills through practical disciplinary coding experience. By implementing these elements, the study anticipates a substantial improvement in students' understanding of FEA principles and their ability to translate them into solutions for real-world engineering challenges.**

Keywords*— Mechanics of materials, computational thinking, threshold concepts, Finite Element Analysis, ChatGPT, Scaffolding.*

I. BACKGROUND

The ability to leverage computational tools for finite element analysis (FEA) is crucial for modern mechanical engineers. While FEA offers significant advantages in design and analysis, mastering it presents challenges for students due to the interplay of abstract concepts form mechanics of materials and computational modeling [1]. Mastering computational tools and analyzing complex systems stand as fundamental pillars in contemporary engineering and scientific endeavors. FEA takes advantage of powerful computational tools, enabling students to simulate the behavior of materials and structures under various loading conditions. Understanding FEA empowers professionals to design and optimize products, predict material failure, and analyze complex physical phenomena.

Similarly, computational science (CS) encompasses a wide range of computational methods and concepts used to model and solve scientific problems. Mastery of CS equips engineers and researchers with the tools to analyze data, simulate natural systems, and accelerate scientific discovery. However, FEA and CS pose significant challenges for students [2, 3]. These subjects often involve complex mathematical concepts and abstract theoretical underpinnings that require students to translate theoretical knowledge into practical applications. Recent studies have described some of the difficulties students face in grasping core FEA principles, including issues with mesh generation, interpreting results, and understanding the limitations of the method [4]. CS education, like FEA education, faces shared challenges, as learning about complex algorithms, data structures, and programming languages is not trivial [5]. Hence, while computation is better learned within disciplinary contexts, such a process involves a form of complex learning [6]. These difficulties can lead to frustration and hinder student success in these crucial fields.

Educators have increasingly turned to pedagogical approaches that provide targeted support and structure student learning to address these challenges and support student understanding. Classroom scaffolding, a technique that involves providing temporary support to help students complete tasks they might not be able to develop independently, remains a well-established approach to supporting student learning [7, 8].

Scaffolding techniques can encompass a variety of methods, such as providing prompts and cues, breaking down complex concepts into smaller, more manageable steps, and offering guided practice opportunities. Studies have shown that effectively implemented scaffolding can significantly improve student learning outcomes in STEM education [9, 10]. Learning about complex subjects like FEA and CS often requires students to bridge the gap between theoretical knowledge and practical application. Research supports the effectiveness of established pedagogical approaches like self-explanation with worked examples in facilitating student understanding of complex subjects [11]. Self-explanation activities, where students articulate their understanding of a concept in their own words, foster deeper cognitive processing and reinforce key ideas [12]. Worked examples, which showcase the step-by-step solution to a problem, provide a concrete model for students to emulate [13]. Recent studies have shown that combining these techniques leads to improved learning outcomes in STEM fields, including problem-solving skills and knowledge retention [14, 15].

In parallel, the emergence of large language models (LLMs) like ChatGPT brings opportunities and challenges for classroom practices. LLMs are sophisticated AI models trained on massive amounts of text data, enabling them to generate text, translate languages, write different kinds of creative content, and answer questions in an informative way [16]. The potential applications of LLMs in education are vast and still under exploration. However, early indications suggest that LLMs like ChatGPT could prove to be valuable tools for enhancing student learning, particularly in subjects like FEA and CS [16, 17, 18]. This work-in-progress research paper explores the potential of integrating classroom scaffolding techniques and ChatGPT to support student understanding of FEA and disciplinary programming. We hypothesize that the strategic integration of ChatGPT's capabilities alongside established scaffolding methods like self-explaining activities for worked-examples will support student complex learning.

II. METHODS

A. Participants

The participants in this study included 32 senior mechanical engineering students enrolled in a Finite Element Methods course at a Hispanic Serving Institution (HSI) located in the southern United States.

B. Procedures, data collection and data analysis

Students in this course had limited MATLAB experience, so they were given multiple opportunities to develop their understanding through structured learning activities. The teaching strategy utilized the use-modify-create progression, which has proven effective for learning programming [19]. It began by engaging students in self-explanation of completed examples before guiding them to modify and extend these examples and ultimately create new solutions. The initial activity aimed to familiarize students with MATLAB syntax and basic matrix operations, followed by an introduction to the direct stiffness method. Students then coded the algorithm in MATLAB and were assigned to annotate their code with self-explanatory comments [20].

Subsequent activities required students to modify their existing solutions to handle trusses and compare their results with an instructor-provided example. They further extended this solution to grids and frames. The final assignment engaged students in generating a working code for FEA using the direct stiffness method and ChatGPT to solve grid and frame problems. Students submitted their entire interaction with the AI tool and reflected on their learning experiences through this process. At the end of the course, students completed a series of open-ended questions inviting them to provide more information, share their opinions, or elaborate on their experiences.: What is your perspective on the value of using computational methods, such as Finite Element Analysis, compared to traditional hand calculations in engineering? (Q1); Is there any value in using the provided MATLAB code to complete homework assignments? Please explain (Q2); Did you encounter any challenges or difficulties when working on homework assignments using the provided MATLAB code? Please explain (Q3); and How did you find the process of learning Finite Element Analysis (FEA) methods in MATLAB before transitioning to Abaqus? (Q4). Abaqus is a commercial software for FEA. Although Abaqus allows element and material customization, it is typically used in undergraduate courses as a 'black-box' simulation solver, and the students do not have access to the code.

This work-in-progress paper will focus on analyzing student responses to these questions using content analysis to understand their experiences and perceptions of learning FEA and CS simultaneously. The categorization process involved analyzing each sentence of students' responses to identify key words and phrases. These key words (codes) led to various categories. The entire text was then reviewed to ensure accurate classification according to the resulting categories. The categorization process was supervised and refined by an expert in engineering and computational education to ensure consistency and reliability. This expert's oversight helped validate the accuracy and coherence of the categorization. Through this analysis, we were able to pinpoint the strengths and weaknesses of the integrated classroom scaffolding and ChatGPT approach.

III. RESULTS AND DISCUSSION

1. The value of using computational methods compared to manual completion for FEA

The first two questions served as an initial exploration of the value proposition offered by computational methods in this context. Table 1 describes the identified categories for the two questions, along with their frequencies (F). Content analysis of student responses revealed a consensus on the advantages of using MATLAB for practical FEA problems, highlighting the significant value of computational methods like FEA in modern engineering practices.

A total of 23 out of 32 students acknowledged the significant role of computational methods in addressing complex engineering problems (A1). An example of the student's response categorized into this category is: "*The*

natural progression of technology has allowed us to expedite the computational demand for engineering and has allowed for exponentially complex problems to be solved quickly. It is incredibly useful". This student emphasizes the pivotal role of technology in accelerating computational capabilities within engineering. Category A2, mentioned in nine out of 32 responses, focuses on the adoption and integration of computational methods in engineering practices: "*FEA is an important thing to understand how to do. It can analyze complex geometries and simulate numerous boundary conditions that would be way too time consuming to perform by hand. A lot of hand calculations also involve making assumptions such as 'no slip' or 'frictionless' surfaces which can all be simulated easily in any simulation software. However, it's important to note that FEA should be used judiciously, as it requires expertise to set up and interpret results accurately. Additionally, it relies on accurate input data, including material properties and boundary conditions, to yield meaningful results*". This student highlights how computational methods offer substantial advantages over manual calculations, particularly in efficiently analyzing complex geometries and simulating various boundary conditions. The adaptability of the code to similar FEA problems, achievable through editing, further piqued students' interest in its effectiveness for problem-solving [21]. Finally, the analysis of responses to Q1 revealed a dominant theme: the efficiency and speed advantages provided by computational methods in solving complex or large-scale engineering problems (A3). A consensus emerged among all 32 students regarding this characteristic: "*Computational methods allow for quicker calculations, which can save money in industry*".

TABLE I. CODING SCHEME FOR Q1 & Q2 STUDENTS' RESPONSES.

Question	Category	Code	F
Q1	Technology Progression	A1	23
	Computational Adoption	A2	9
	Time Saving & Streamlining	A3	32
Q2	MATLAB Visualization	A5	10
	Repeatable Problems	A6	7
	MATLAB Efficiency	A7	5
	Programming Skill Development	A8	20

To explore student perceptions of the value of using the provided MATLAB code for completing homework assignments (Q2), student responses centered around two key subthemes: the advantages of visualization and data analysis capabilities offered by the provided MATLAB code (categories A5 & A6) and the benefits of using MATLAB for efficient variable handling (categories A7 & A8). Ten out of 32 students' responses highlight the use of MATLAB in generating visual representations of data and results (A5), aiding in better understanding and interpretation of engineering concepts and solutions. Through simulation, students could gain deeper insights into the specific patterns of stress and force distribution within the analyzed structure [21]. Likewise, seven out of 32 responses highlight the efficiency and effectiveness of using MATLAB to change parameters in calculations that need to be performed repeatedly (A6): "*MATLAB allows you to change variables without breaking certain components, and has the syntax/code to make functions faster, unlike using excel. Which makes doing a repetetive calculation like iteration simpler and faster.*" MATLAB's versatility in modifying variables and streamlining function creation supports students' grasp of complex engineering concepts, making it invaluable for simulation and analysis tasks in engineering education. This enables students to test their ideas through simulation runs, enhancing their learning experiences [22]. Categories A7 and A8 contribute to the emergence of a second subtheme. Responses in category A7 emphasize how MATLAB allows for efficiently data processing compared to other methods like manual calculations: "*Yes, MATLAB is yet another powerful tool that allows for computing and analyzing vasts amount of data quickly…*". Finally, category A8 is the most common theme in student perceptions. These responses typically emphasized how working with MATLAB enhances students' abilities in coding: "*Yes. The MATLAB code homework assignments allow for both a deeper understanding of the underlying concepts, but also an ability to work with the code and understand what it is doing and how it is doing it in order to solve the problems that are being worked on. Through this approach, it helps to provide a better overall understanding of the coding involved to solve a problem computationally*".

2. Challenges and support of using MATLAB for FEA

Analysis of student responses provided insights into the specific challenges students encountered while working on MATLAB assignments, and how students addressed them. Table 2 describes the identified categories for Q3, along with their frequencies.

TABLE II. CODING SCHEME FOR Q3 STUDENTS' RESPONSES.

Question	Category	Code	F
Q3	Troubleshooting Difficulty	B1	21
	Small Errors	B2	25

Students' responses discussed the steep learning curve (Category B1) and the relationship between the student's experience or skill level and their likelihood of making syntax errors (Category B2). Category B1, mentioned in 21 out of 32 responses, reflected on the challenges related to troubleshooting MATLAB code given their limited experience: "*Yes. The learning curve and troubleshooting made it a pain to use if not completely fleshed out by the professor already*"; and "*Yes, due to my lack of experience with coding, it was very difficult for me to understand the code and therefore it was hard for me to create my own or duplicate one*". Students encountered difficulties comprehending specific code segments, ultimately limiting their ability to resolve the presented problems due to insufficient coding

knowledge. This aligns with recent studies that hypothesize similar challenges [23, 24].

A second category describes how small errors in MATLAB code led to significant issues during the task (B2). This category, evidenced 25 out of 32 responses, included errors that could be syntax-related, logic errors, or other small inaccuracies that had a disproportionate impact on the functionality of the code: "*I usually run into problems when it comes to the syntax of MATLAB. I am used to programming in C/C++ which is similar but has small differences. This has led to some frustrating errors when compiling*"; and "*There were times when I had an error, and I couldn't figure out what I needed to change to fix it. It becomes frustrating when this happens and prevents you from being able to move forward with the work*".

3. The Effectiveness of Simulation Tools

Student perceptions of learning Finite Element Analysis (FEA) together with the MATLAB environment before using a 'black-box' simulation were examined through question four. Student responses revealed insights into how the computational tool influenced their grasp of FEA theory and core concepts. Table 3 describes the identified categories for Q4, along with their frequencies. At first sight, the results depict the different points of view that emerged from the transition experience. Some students found the transition challenging, while others described a smooth adaptation.

TABLE III. CODING SCHEME FOR Q3 STUDENTS' RESPONSES.

Question	Category	Code	F
Q4	Fundamentals Grasp	C1	6
	Difficulty of Learning	C2	9
	Adaptation Experience	C3	11
	Benefits	C4	6

Responses classified into the fundamental grasp category (C1) emphasized the importance of understanding fundamental concepts before delving into simulations. This perception was reflected in the responses of six out of 32 students, suggesting that they found it crucial to grasp theoretical underpinnings before transitioning to practical applications in Abaqus. The following quote exemplifies a student response classified under this category: "*Learning the process in MATLAB makes sense, as the fundamentals of FEA should be learned first before using simulations*". Similarly, six out 32 responses indicated the benefits (C4) students perceived from learning FEA methods in MATLAB before transitioning to Abaqus: "*MATLAB really helped when it came to knowing what Abaqus would be doing behind the scenes*".

Students' perceptions regarding the difficulty of learning FEA methods in MATLAB centered on several key challenges (C2): a lack of confidence in coding abilities, difficulty establishing a clear connection between MATLAB and Abaqus, and the use of unfamiliar software in general. The following student's response illustrates this category: "*I found it super confusing because I think I was more focused on abiding by the laws of the MatLab coding language and so it was hard to see what was actually happening when coding and running the code. I think abaqus has helped me a great deal understanding the effects of the elements of your design (e.g., material property, part geometry)*". Research suggests that students face challenges in comprehending specific code segments, ultimately hindering their ability to address the presented problems due to insufficient general coding knowledge [20, 22]. Lastly, the adaptation experience category (C3) reflects on the positive experiences and opinions regarding the transition from learning FEA in MATLAB to using Abaqus. Some students appreciated the transition to Abaqus because it allowed them to build parts or structures, aiding in their understanding and visualization of concepts. Also, MATLAB was perceived as beneficial for understanding the math behind simulations and gaining insight into the core functionality of Abaqus. Based on the responses, the transition from MATLAB to Abaqus was facilitated by the understanding gained in MATLAB, making it easier to comprehend the processes in Abaqus: "*It was helpful because for example on the abaqus activity 4, removing the left pressure and replacing it with a fixed boundary condition simplified the analysis for abaqus and made the results come out faster. This was helpful because we know the way the abaqus program will do the analysis, so we helped it by making the problem easier for it*".

IV. CONCLUSIONS

This work-in-progress study explored students' experiences and perceptions of learning Finite Element Analysis (FEA) in parallel with disciplinary programming in MATLAB. Since this is a complex learning process, the research team created an instructional design to support student learning including: (1) scaffolding strategies in the form of worked-examples and self-explaining activities; and (2) an activity where students engaged in asking ChatGPT to automatically generate MATLAB code for FEA.

The results suggest that, despite the scaffolding approaches integrated into the learning environment, some students still struggle to develop their programming skills in the context of a disciplinary course. However, they value engaging in this process as it teaches them the FEA process in detail before using a black-box software such as Abaqus.

This preliminary study has the potential to transform engineering curricula by integrating advanced computational tools, making complex concepts more accessible and engaging. It establishes a precedent for updating engineering education and creating useful skills applicable to contemporary industry practices by utilizing technologies such as ChatGPT.

Our future work will look into their experiences of specifically engaging with Chat-GPT as a co-pilot for coding activities. This work will also explore how students engage in an interaction with the LLM, and the kind of support they may need to succeed in these coding tasks.

V. ACKNOWLEDGMENTS

This research was supported in part by the U.S. National Science Foundation under award No. CAREER 2237313, and the Army Research Office: W911NF2310192.

VI. REFERENCES

[1] S.B. Montfort, and D. Pollock, "An investigation of students' conceptual understanding in related sophomore to graduate-level engineering and mechanics courses", Journal of Engineering Education, vol. 98, no. 2, pp. 111-129, 2009.

[2] H. Nguyen, J. Wu, and F. Gosselin, "Understanding student difficulties in finite element analysis courses: A literature review," in 2021 ASEE Annual Conference & Exposition, 2021, pp. 1-11.

[3] A. Yadav, J. Nourbakhsh, and A. K. Tew, "Understanding student challenges in learning computational science concepts," in 2022 ASEE Annual Conference & Exposition, June 2022, pp. 1-12.

[4] V. Plevris and G. Markeset, "Educational Challenges in Computer-based Finite Element Analysis and Design of Structures," Journal of Computer Science, vol. 14, no. 10, pp. 1351–1362, 2018, doi:10.3844/jcssp.2018.1351.1362.

[5] C. Watson and F. W. B. Li, "Failure rates in introductory programming revisited," Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (ITiCSE '14), 2014, doi:10.1145/2591708.2591749.

[6] C. Vieira, A. J. Magana, A. Roy, and M. L. Falk, "Student explanations in the context of computational science and engineering education," Cognition and Instruction, vol. 37, no. 2, pp. 201-231, 2019.

[7] C. Vieira, A. J. Magana, A. Roy, and M. Falk, "Providing students with agency to self-scaffold in a computatiopnal science and engineering course," Journal of Computing in Higher Education, vol. 33, pp. 328-366, 2021.

[8] B. R. Belland, "Scaffolding: Definition, Current Debates, and Future Directions," in Handbook of Research on Educational Communications and Technology, pp. 505–518, 2013, doi:10.1007/978-1-4614-3185-5_39.

[9] K. D. Simons and J. D. Klein, "The Impact of Scaffolding and Student Achievement Levels in a Problem-based Learning Environment," Instructional Science, vol. 35, no. 1, pp. 41–72, 2006, doi:10.1007/s11251-006-9002-5.

[10] V. Vijayan, S. Fang, S. A. Barclay, M. E. Reissman, and T. Reissman, "Impact of scaffolding and hands-on assignments within mechatronics on student learning outcomes of KEEN's entrepreneurial mindset," International Journal of Mechanical Engineering Education, vol. 0, no. 0, 2024, doi:10.1177/03064190241240463.

[11] J. L. De La Hoz, C. Vieira, and C. Arteta, "Self-explanation activities in statics: A knowledge-building activity to promote conceptual change," Journal of Engineering Education, vol. 12, no. 3, pp. 741–768, 2023, https://doi.org/10.1002/jee.20531.

[12] B. Rittle-Johnson, A. M. Loehr, and K. Durkin, "Promoting self-explanation to improve mathematics learning: A meta-analysis and instructional design principles," ZDM, vol. 49, no. 4, pp. 599–611, 2017, doi:10.1007/s11858-017-0834-z.

[13] J. Sweller, "Cognitive Load Theory," Psychology of Learning and Motivation, pp. 37–76, 2011, doi:10.1016/b978-0-12-387691-1.00002-8.

[14] A. Renkl, "Learning from worked examples: How to prepare students for meaningful problem solving," in Applying science of learning in education: Infusing psychological science into the curriculum, V. A. Benassi, C. E. Overson, and C. M. Hakala, Eds. Society for the Teaching of Psychology, 2014, pp. 118–130.

[15] K. M. McGinn, L. K. Young, A. Huyghe, and J. L. Booth, "The effect of worked Examples and Self-Explanation prompts on mathematics standardized assessments," Journal of Research on Educational Effectiveness, pp. 1–23, 2023, https://doi.org/10.1080/19345747.2023.2243254.

[16] G. Orlando, "Assessing ChatGPT for coding finite element methods," Journal of Machine Learning for Modeling and Computing, vol. 4, no. 2, pp. 135–171, 2023. https://doi.org/10.1615/jmachlearnmodelcomput.2023049326.

[17] B. Qureshi, "Exploring the use of ChatGPT as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges," arXiv preprint arXiv:2304.11214, 2023.

[18] H. Singh, M. H. Tayarani-Najaran, and M. Yaqoob, "Exploring computer science students' perception of ChatGPT in higher education: A descriptive and correlation study," Education Sciences, vol. 13, no. 9, p. 924, 2023.

[19] C. Vieira, R. L. Gómez, M. Gómez, M. Canu, and M. Duque, "Implementing Unplugged CS and Use-Modify-Create to Develop Student Computational Thinking Skills," Educational Technology & Society, vol. 26, no. 3, pp. 155-175, 2023.

[20] C. Vieira, A. J. Magana, M. L. Falk, and R. E. Garcia, "Writing in-code comments to self-explain in computational science and engineering education," ACM Transactions on Computing Education (TOCE), vol. 17, no. 4, pp. 1-21, 2017.

[21] C. Vieira, D. Restrepo and J. L. De La Hoz, "Computational Notebooks in a Finite Element Analysis Course: Engineering Students' Reflections on the Value and Challenges of Computational Approaches," 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA, 2023, pp. 1-5, doi: 10.1109/FIE58773.2023.10342890.

[22] A. J. Magana, M. L. Falk, C. Vieira, M. J. Reese, O. Alabi, and S. Patinet, "Affordances and challenges of computational tools for supporting modeling and simulation practices," Computer Applications in Engineering Education, vol. 25, no. 3, pp. 352–375, 2017. [Online]. Available: https://doi.org/10.1002/cae.21804

[23] M. Wermelinger, "Using GitHub Copilot to Solve Simple Programming Problems," in Proceedings of the 54th ACM Technical Symposium on Computer Science Education, vol. 1, pp. 172-178, 2023. [Online]. Available: https://doi.org/10.1145/3545945.3569830.

[24] S. Garces, C. Vieira, G. Ravai, y A. J. Magana, «Engaging students in active exploration of programming worked examples», Education and Information Technologies, pp. 1-18, 2022.