The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
Paper ID #43332

Using AI Interactive Interfaces in Design of Machine Elements Education

Can Uysalel, University of California, San Diego

Can Uysalel is a Ph.D. graduate student researcher working at UCSD Department of Mechanical and Aerospace Engineering. His research interests include materials characterization, machine learning, and STEM education.

Zachary Fox, University of California, San Diego

Zachary Fox is an Undergraduate Mechanical Engineering Student & Researcher working under the UC San Diego Mechanical & Aerospace Engineering Department. Hos research interests include mechanical failure design and methods to bridge engineering topic understanding in academic environments, both in college and high school learning environments.

Maziar Ghazinejad, University of California, San Diego

Dr. Maziar Ghazinejad is an associate teaching professor in the Mechanical and Aerospace Engineering Department at UC San Diego. He received his Ph.D. in mechanical engineering from UC Riverside and holds M.S. degrees in mechanical and electrical engineering. His research interests center around advanced materials, mechanics, and engineering design. As an engineering educator, Ghazinejad applies active learning techniques to develop curricula and pedagogical models for engineering design, mechanics, and advanced manufacturing. He was named a Changemaker Faculty Fellow and a recipient of the 2022 UC San Diego Distinguished Teaching Award for his contributions to engineering pedagogy. Dr. Ghazinejad is a member of the American Society of Mechanical Engineers (ASME), Materials Research Society (MRS), American Society of Engineering Education (ASEE), and the International Society for Optics and Photonics (SPIE).

©American Society for Engineering Education, 2024

Using AI Interactive Interfaces in Design of Machine Elements Education

Abstract

The continuous advancements in artificial intelligence interfaces are poised to have a profound impact on STEM education and involvement. Engineering design educators are perhaps among those at the forefront of STEM education experiencing the first tides of this change. An example of such a trend is the course Design of Machine Elements, a mainstay of Mechanical and Aerospace Engineering (MAE) curricula, which embodies many algorithms that integrate a combination of scientific topics and industry protocols. In this work in progress, we assigned a class of 62 MAE machine design students to write computer codes that implement several required inputs to generate design parameters for shafts used for specific power transmission parameters. The students were also asked to explore the applicability of an open artificial intelligence interface, such as ChatGPT, to help develop a multi-step design code. After generating and verifying the AI-assisted design codes, students were required to evaluate their accuracy and functionality by comparing them to the codes they wrote in the first part. Analyzing the results of our novel design assignment offers valuable insights into the interplay between the users' design knowledge and the efficacy of the AI-assisted code. The results also indicate the future role of engineers as experts who can use their deep physical understanding of engineering systems to verify AI-generated algorithms, steering them from erroneous conclusions.

Keywords: Machine design, artificial intelligence, interactive learning, programming in design

I. Introduction

Design engineers are often tasked with developing computer codes to execute these multi-step technical procedures to design different machine elements. A relevant example of such machine elements would be the design of a shaft-based power transmission. The technical process for engineering interdependent mechanical components is introduced in the principles of machine design - both the course and the practice. The course delves into the critical aspects of failure theory, design methodology, and the practical application of these essential mechanical elements and structures. Students engage in analyzing components using theory from statics, dynamics, and mechanics of materials. The curriculum covers static and fatigue failure theories, emphasizing their relevance in the design process. Through these modes, the Course Learning Objectives (CLOs) address the design and key considerations of fundamental machine components in mechanical engineering: shafts, gears, bearings, springs, and fasteners. Students gain proficiency in analyzing and designing mechanical components under both steady and time-varying loads, considering factors such as cyclic fatigue, stress concentrations, surface finish, component size, load types, and environmental conditions.

Additionally, the course emphasizes the utilization of design codes and standards to develop, analyze, and specify common machine elements for applicable components. Students learn to specify and select common machine elements from catalogs, integrating them into complex systems such as gear trains for power transmissions. The objectives also focus on the application of materials selection procedures for optimal choices in various applications and components. Furthermore, students develop the ability to apply strength and deflection-based design methodologies in the analysis of both individual machine components and integrated machine systems, enhancing their proficiency in engineering design.

Processes as involved as those described above can be tedious and time-consuming, yet critical to component functionality. Student engineers endeavoring to become competent academic and industry engineers have to develop the ability to iterate and validate design choices efficiently. As such, the practice of effectively generating a user-driven script to solve complicated problems is invaluable to complex mechanical problem-solving. Generating such a script requires an intrinsic and robust understanding of the subject material. Such understanding must be comprehensive of unique cases, and able to discern critical failure points. However, creating such code that is comprehensive of all possible geometries and design criteria would be nearly impossible to complete manually. This is where the use of AI Large Language Model (LLM) software can benefit the informed user.

A. AI in Engineering Education

Several studies have explored integrating Artificial Intelligence (AI) with engineering education, as well as its application in machine design. AI in engineering education can be categorized into Deep Generative Models (DGMs) and AI-driven Gamification:

Deep Generative Models (DGMs): Regenwetter et al. [1] conducted a study focusing on the potential of Deep Generative Models (DGMs). These models aim to replicate datasets. However, the authors highlighted the limitations of DGMs in addressing engineering design challenges. Through a case study on bicycle frame design, they demonstrated that while DGMs can generate new frames resembling past designs, they often fall short of meeting engineering performance standards and requirements. The findings underscored the importance of engineering-centric considerations in AI modeling, suggesting that purely similarity-focused approaches may not effectively translate to engineering tasks [2]. The researchers emphasized the potential of AI models as design "co-pilots" with appropriate task-oriented metrics in place. Ngai et al. [3] introduced a learning AI-based platform for wearable computing, consisting of a user-friendly construction platform, a hybrid text-graphical programming environment, and a sample syllabus guiding students through basic concepts in wearable computing. The study observed heightened attention and engagement in the learned topics, attributed to participants' interest in the AI-based platform.

Wong and Looi [4] investigated swarm intelligence education employing a feedback-based AI model. Their study delved into how feedback mechanisms can enhance participants' learning experiences. In particular, they found that feedback provided by the AI model enabled learners to receive immediate correction, reinforcement of correct behaviors, identification of weaknesses, adaptation of strategies, and reflection on their learning process. The AI model dynamically recommended personalized learning pathways based on students' progress. This multifaceted feedback approach contributed to a more effective and engaging learning environment, ultimately leading to improved understanding and mastery of swarm intelligence concepts and algorithms. Williamson et al. [5] revealed how Internet of Things (IoT) technology can replicate brain functions within physical settings, enabling the sensing and comprehension of human cognitive behaviors. They also showed how this innovation enhances human cognition and performance.

AI-driven Gamification: AI-driven gamification in education represents a paradigm shift, where artificial intelligence is seamlessly integrated into gaming experiences to enhance learning outcomes. Successful educational games seamlessly blend pedagogical design, domain knowledge, and emotional elements with gameplay. Moon et al. [6] introduced a computer-assisted exponential learning model using digital game-based learning approaches. By incorporating student feedback throughout their research process, they stimulated emergent learning. This means that AI-driven gamification can personalize the learning experience, adjusting game difficulty, content presentation, and feedback based on individual learner preferences and performance.

B. AI in Machine Design

In the field of machine design, we can categorize AI methods into two groups, namely strong and weak AI, based on the problem-solving approach utilized:

Strong AI: Strong AI methodologies aim to replicate the cognitive and heuristic mechanisms employed by humans in problem-solving tasks [7]. The objective of these systems is to endow computers with the capacity for thinking, perception, and reasoning. Salehi and Burgueno [8] emphasized the potency of AI methodologies, such as genetic algorithms and machine learning (ML), in optimizing design processes. Current strong AI studies exemplify how AI-driven algorithms efficiently navigate complex design spaces, identifying innovative solutions that satisfy diverse constraints and objectives. Additionally, AI accelerates decision-making processes, reduces error rates, and enhances computational efficiency [3]. Notably, ML and pattern recognition are attracting substantial interest as cutting-edge intelligent techniques in engineering applications.

Weak AI: Weak AI typically refers to computational software proficient in executing specific tasks without relying on heuristic processes [7]. Despite the term "weak" implying otherwise, these AI methods often outperform humans in efficiency and success for a particular task. Caldas and Norford [9] utilized the principles of generative and object-oriented design. They proposed a computer tool aiding designers in generating and evaluating various facets of a solution, particularly focusing on environmental performance aspects of buildings. Their study primarily centered on optimizing energy efficiency in office buildings through the strategic placement and sizing of windows. Simulations were conducted to assess different configurations, guiding a genetic algorithm (GA) search towards identifying low-energy solutions. In general, weak AI methodology can be extended to address a diverse array of design challenges, including the selection of construction materials, the design of shading elements, and the optimization of lighting and mechanical systems for buildings. The majority of AI methods currently in use are categorized as weak AI [10]. However, there has been a recent emergence of the concept of super AI, which denotes software with superior processing and cognitive capabilities compared to humans, albeit still in its early stages of research.

For our case study on the impact of interactive artificial intelligence, we chose a standard machine component problem of designing a power transmission shaft. Writing a computer code that provides applicable shaft design parameters while offering multiple flexibility to users is a relatively complicated engineering task. The design parameters depend on several factors, including applied mechanical loads, choice of material, physical conditions of operation and shaft, statistical considerations, and the selected safety factors that the user enters. Some aspects of this code development demand a deep understanding of engineering mechanics. While valuable, mere proficiency with algorithms and programming are insufficient to address the involved intricacies. In the following, we address important facets and interpretations of this case study.

II. The Project Procedure and Method

The full version of the shaft design assignment statement is available in Appendix A. The class was almost entirely composed of senior students with advanced footing in engineering mechanics and materials. For our comparative analysis, we first asked the machine design class to develop a MATLAB code capable of taking user inputs such as mean torque (T_m), alternating torque (T_a), mean bending moment (M_m), alternating bending moment (M_a), choice of material, and safety factor (n) (Appendix B). The program was expected to return estimated shaft dimensions for selected material. Figure 1 provides a schematic representation illustrating the key geometric components involved. It is crucial that the user inputs are chosen carefully, and the program itself should determine certain parameters, such as stress concentrations, based on the provided inputs. Additionally, the students were expected to validate the functionality of their MATLAB code by comparing results with the previous hand calculations in the proof of concept section. Numerical examples with varied inputs and initial guesses should be included to demonstrate the code's versatility and reliability.

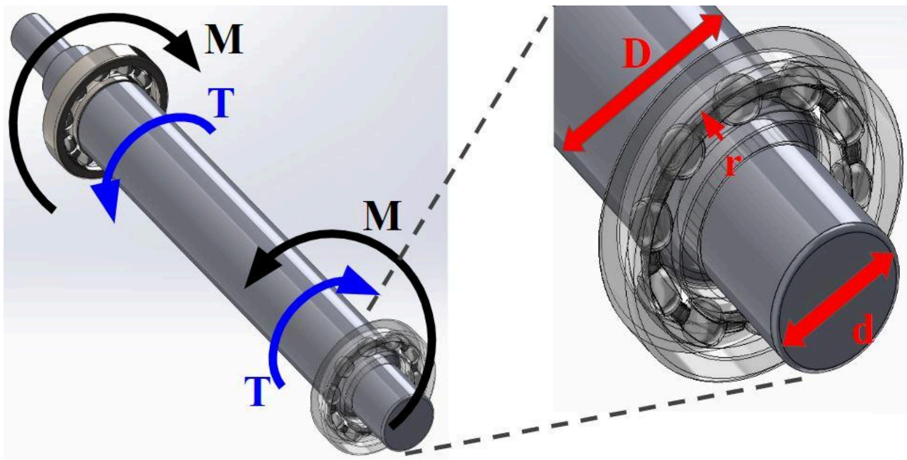
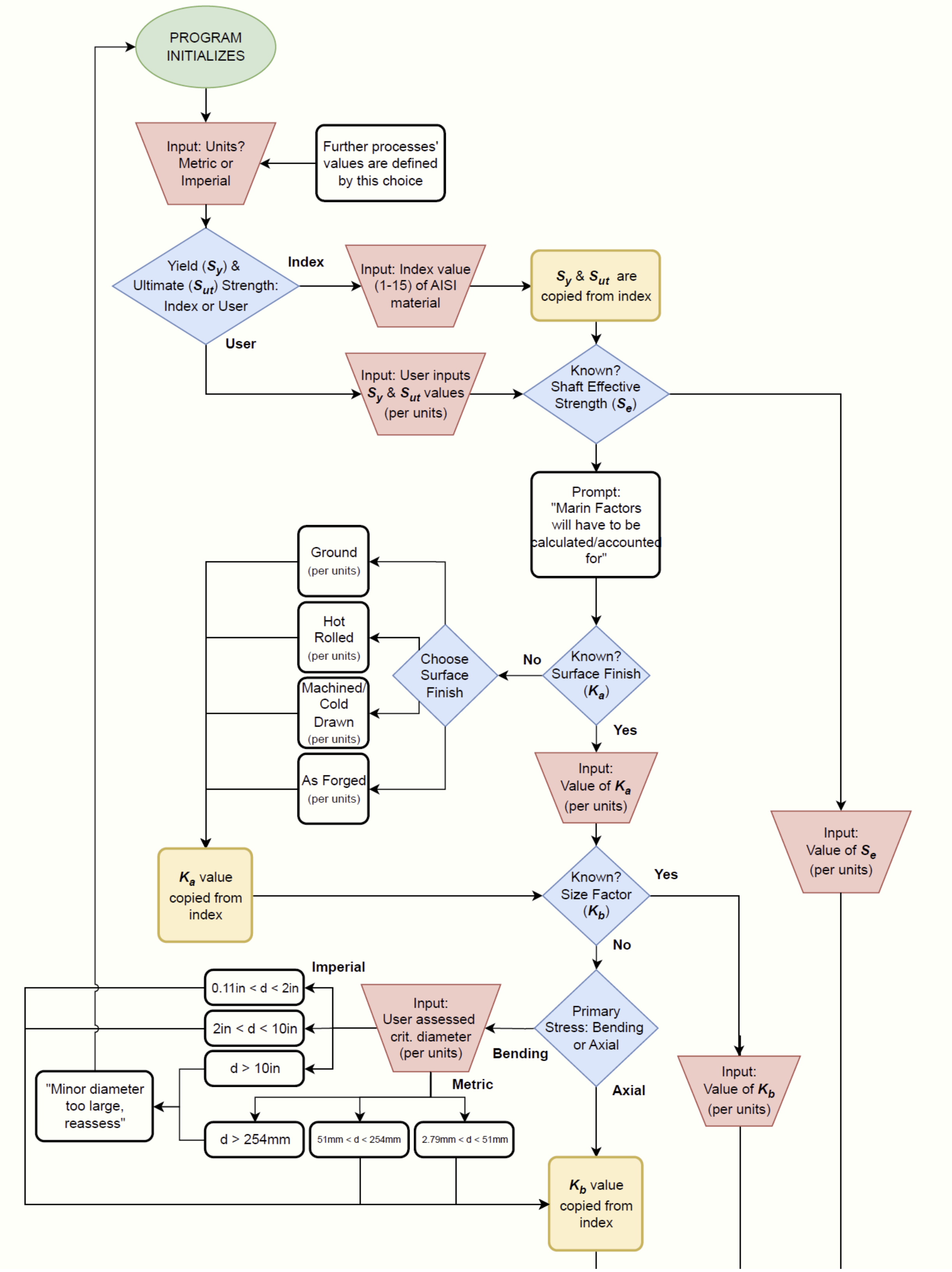


Figure 1: Schematic of a circular shaft under alternating loads. In this illustration, D represents the larger diameter, d the smaller diameter, r is the fillet radius, while T and M are the torsion and bending moment, respectively.

Figure 2 presents an algorithmic flowchart detailing the iterative steps involved in the design computational process. The instruction team did not disclose the second part of the design program assignment involving Generative AI to develop a similar design code. This is mainly to ensure students independently develop their design code without outside influence. After the students completed the first stage of the design assignment, the instructor posted the second part, followed by a briefing session for the students. We explained that the project's objective is to explore the applicability of artificial intelligence interfaces (such as ChatGPT) to produce a multi-step design code and how to use these tools to enhance our engineering range. The students were particularly asked to investigate the possibility of using well-thought prompts in ChatGPT to acquire a code capable of providing shaft design parameters based on the entered loads, material selection, and geometric considerations. The assignment directed the students to address the following areas in their critical analysis: (i) the role of appropriate prompting to get applicable skeleton codes and how a knowledgeable designer iterates on a prompt to enhance the sophistication of the obtained code, (ii) the reproducibility of the AI-assisted code, based on prompt patterns, (iii) the comparison between students' code syntax and structures, and those of AI-assisted codes, and (iv) developing the general techniques that a knowledgeable engineer needs to verify AI-generated codes (critical constraints and pitfalls).


PROGRAM INITIALIZES
Input: Units? Metric or Imperial
Further processes' values are defined by this choice
Yield (S_y) & Ultimate (S_{ut}) Strength: Index or User
Index
Input: Index value (1-15) of AISI material
S_y & S_{ut} are copied from index
User
Input: User inputs S_y & S_{ut} values (per units)
Known? Shaft Effective Strength (S_e)
Prompt: "Marin Factors will have to be calculated/accounted for"
Ground (per units)
Hot Rolled (per units)
Machined/ Cold Drawn (per units)
As Forged (per units)
Choose Surface Finish
No
Known? Surface Finish (K_a)
Yes
Input: Value of K_a (per units)
Input: Value of S_e (per units)
K_a value copied from index
Known? Size Factor (K_b)
Yes
No
Imperial
0.11in < d < 2in
2in < d < 10in
d > 10in
Input: User assessed crit. diameter (per units)
Primary Stress: Bending or Axial
Bending
"Minor diameter too large, reassess"
Metric
Axial
Input: Value of K_b (per units)
d > 254mm
51mm < d < 254mm
2.79mm < d < 51mm
K_b value copied from index

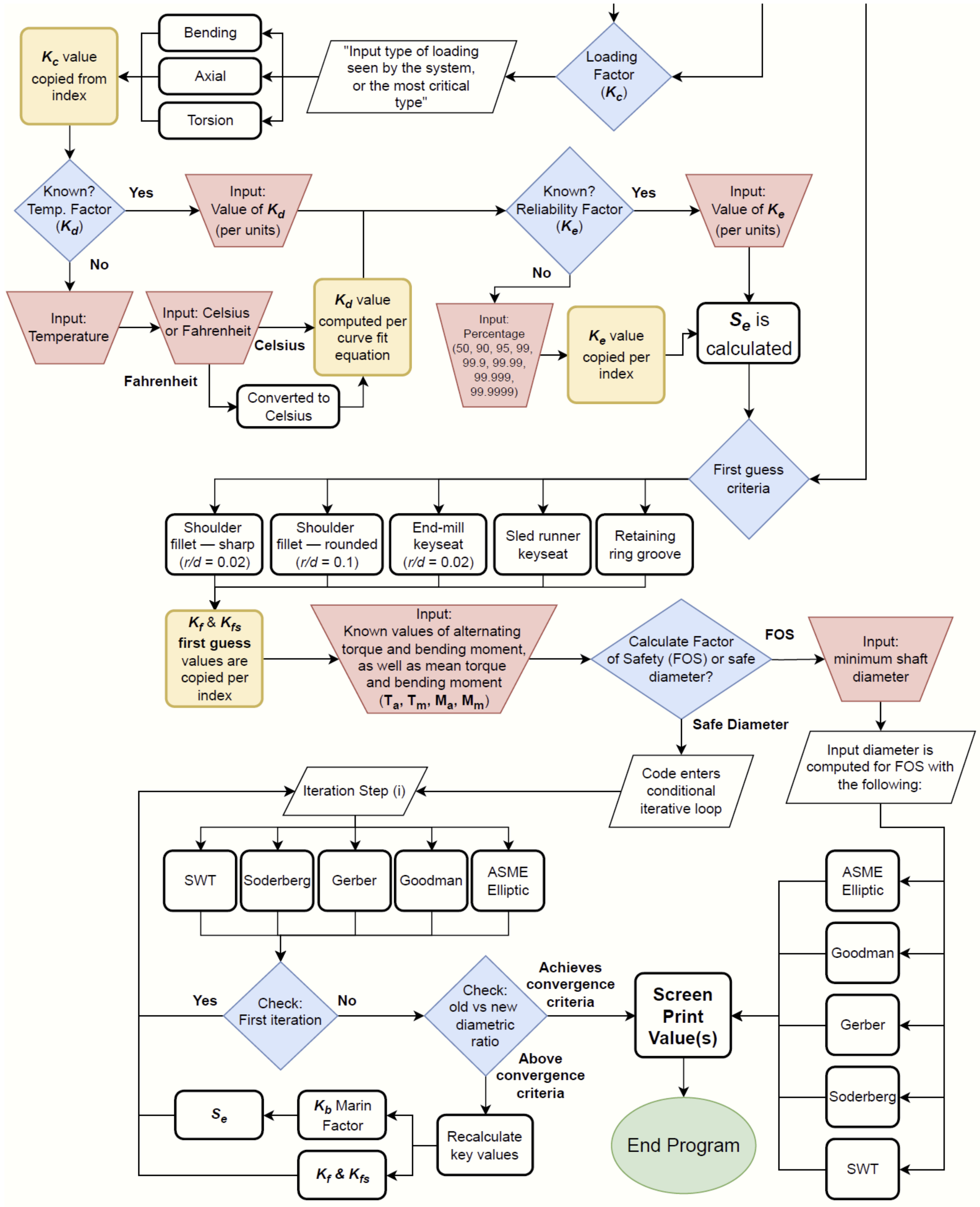


Figure 2: The figure illustrates the iterative steps involved in MATLAB code through an algorithmic flowchart.

III. Results & Discussion

Upon concluding the machine design class, we conducted a survey to collect feedback from students regarding the course project. Our survey involved 11 participants, with a distribution of 8 males and 3 females. The survey and results are available in Appendix C. The subsequent analysis of their responses provides an understanding of their viewpoints on ChatGPT (and by extension, AI LLM generated engineering practice) and the applications to shaft design.

One of the quickly rising skill trends in the human-computer interface is the capability of insightful prompting to achieve the intended outcome from AI chatbots. Here, our students showed great versatility in getting applicable code segments from AI. However, they found it difficult when they had to iterate on the prompt to enhance the sophistication of the obtained design code (Question 7). 91% of respondents mentioned encountering *some difficulties* or *significant difficulties* when using ChatGPT to develop the shaft design code. Most design students stated that the Generative AI was incapable of including all vital facets of design despite their efforts to prompt the platform in the right direction. Major difficulties mentioned were (i) AI programs using incorrect design formulas, (ii) interpreting datasets (charts and figures) containing necessary reference values, and (iii) flaws in developing iterative correction loops.

Naturally, our design students gravitated toward using the platform to obtain applicable "blocks" of code that they (knowledgeable users/designers) can incorporate into a functional overarching structure. This strategy appeared to be effective. Using a shaft design as a case study, when we asked our design students how confident they were in using ChatGPT-generated codes to meet critical design requirements, 63% said they were somewhat not confident or not confident at all, and 36% said they were somewhat confident. This outlook was reflected in the strategies adopted by our budding designers, as they gradually leaned towards using ChatGPT for generating code blocks while building the "big picture" design code themselves (Figure 3). Moreover, the students shared similar views when we asked them what level of machine design knowledge they think is necessary to ensure that the ChatGPT-generated code for shaft design is accurate and applicable (Question 10). 63% answered, "An intermediate knowledge of machine design, including understanding different machine elements and their functions." is necessary. The remaining respondents (36%) stated, "An advanced knowledge of machine design, including expertise in designing and optimizing complex machine systems," is needed to ensure the ChatGPT-generated design code is functional. We also asked students to what degree the code generated by ChatGPT needed to be modified to be functional for shaft design (Question 3). The majority (72%) reported that major modifications were required.

Reproducibility is central to scientific and engineering endeavors. However, the replicability of design codes based on similar prompts can prove challenging due to the dynamic nature of Generative AI algorithms. We tried to address this topic by asking our machine design students about the replicability of the codes generated by ChatGPT for prompts that were conceptually similar but expressed with different wording (Question 4). Interestingly, we received a broad spectrum of responses, with 9% entering “not at all reproducible”, 36% “somewhat not reproducible”, 18% “neutral”, 27% “reproducible”, and 9% mentioning “very reproducible”. Anecdotally, our students’ oral feedback changed throughout working on the assignment (over 3 weeks), mainly suggesting that they could get better reproducibility and functionality from the ChatGPT with similar prompts. While the evolving nature of the AI could be responsible for this, we think the improving dexterity of students with the interactive platform might also have played a role.

Moreover, we asked students if ChatGPT offered valuable insights into the design process that they hadn't previously contemplated (Question 6). The students stated that ChatGPT usefulness was primarily geared towards code syntax/structuring rather than providing valuable insights into the design process that hadn’t been considered before. In particular, our students found the platform inefficient in applying different physics-based design intricacies. As a result, they moved towards using the platform for code blocks and programming efficiency.

Recognizing the emerging *divide-and-conquer* and *generate-and-revise* strategies in our students’ approach, we asked them if they would consider using ChatGPT-generated code as a base that they would edit to become functional for future design tasks (Question 8). The students were predominantly positive, with 63% answering they would “definitely consider using ChatGPT-generated code as a base for future design assignments”. In comparison, 27% mentioned they would consider using ChatGPT-generated code as a base for future design assignments but would also like to have the option to complete the design without ChatGPT assistance. In their descriptive responses, students foresaw the widespread application of AI-generated codes, stating that learning how to generate, check, and modify the provided code will become a necessary skill.

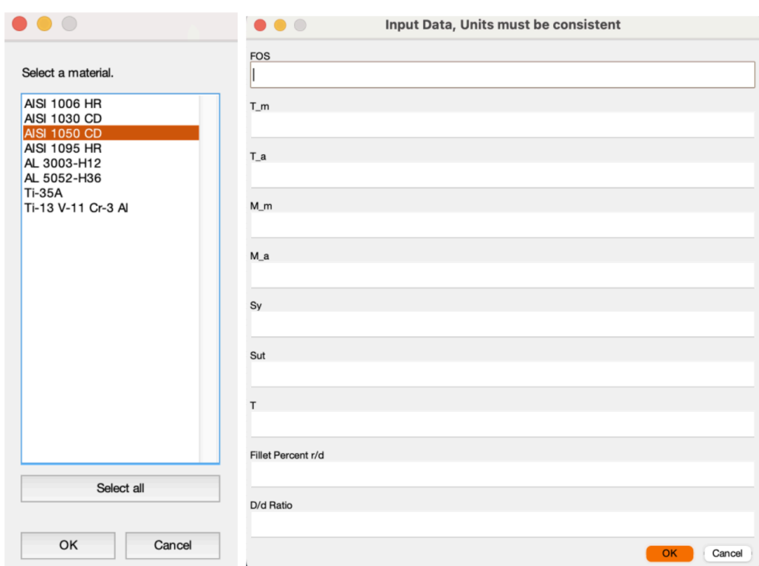


Figure 3: Sample MATLAB user interface generated by an MAE machine design student.

IV. Conclusion

A profound understanding of the physics of engineering problems is still critical in achieving applicable analysis and design. Our case study and survey suggested without such expert knowledge, most AI-generated design codes will face dire roadblocks that render them impractical. One approach that organically shaped up during our case study is to break down a compound, layered engineering case into simpler sub-problems that AI platforms can address more accurately. Such a *divide-and-conquer* strategy will still require the supervision of an insightful engineer who could configure these blocks into a cohesive and sophisticated design solution.

In the following stages of this research, our goal is to expand the educational framework, presenting MAE students with the challenge of designing and manufacturing the shafts. This hands-on integration of digital design and physical production aims to provide students with a comprehensive understanding of the design-to-manufacturing workflow. We also believe more focused research is needed to study the role of human grammar and the growing interplay between human language and prompt syntax in engineering design. On a broader level, prompt engineering can gradually become a formal topic in engineering fields as it evolves into a high-level computer language.

All things considered, it is prudent to develop techniques that a knowledgeable engineer may apply to verify AI-generated codes, an outlook that our budding designers share. Such strategies can involve a curated host of tests by an experienced engineer to detect common pitfalls, including logic errors, lack of robustness, performance issues, and other potential challenges. Given the inevitability of AI involvement in future designs, such assessment instruments will become increasingly essential.

V. Acknowledgements

The authors express their gratitude for the survey responses provided by students and the support received from UCSD's Course Development and Instructional Improvement Program (CDIIP).

References

1. Regenwetter, Lyle, et al. "Beyond statistical similarity: Rethinking metrics for deep generative models in engineering design." Computer-Aided Design, Volume 165 (2023).
2. Zhai, Xuesong, et al. "A Review of Artificial Intelligence (AI) in Education from 2010 to 2020." Complexity 2021 (2021): 1-18.
3. Ngai, Grace, et al. "Deploying a wearable computing platform for computing education." IEEE Transactions on Learning Technologies 3.1 (2009): 45-55.

4. Wong, Lung-Hsiang, and Chee-Kit Looi. "Swarm intelligence: new techniques for adaptive systems to provide learning support." Interactive Learning Environments 20.1 (2012): 19-40.
5. Williamson, Ben, Jessica Pykett, and Selena Nemorin. "Biosocial spaces and neurocomputational governance: brain-based and brain-targeted technologies in education." Discourse: Studies in the Cultural Politics of Education 39.2 (2018): 258-275.
6. Moon, Man-Ki, Surng-Gahb Jahng, and Tae-Yong Kim. "A computer-assisted learning model based on the digital game exponential reward system." Turkish Online Journal of Educational Technology-TOJET 10.1 (2011): 1-14.
7. Yüksel, Nurullah, et al. "Review of artificial intelligence applications in engineering design perspective." Engineering Applications of Artificial Intelligence 118 (2023): 105697.
8. Salehi, Hadi, and Rigoberto Burgueño. "Emerging artificial intelligence methods in structural engineering." Engineering structures 171 (2018): 170-189.
9. Caldas, Luisa Gama, and Leslie K. Norford. "A design optimization tool based on a genetic algorithm." Automation in construction 11.2 (2002): 173-184.
10. Lu, Huimin, et al. "Brain intelligence: go beyond artificial intelligence." Mobile Networks and Applications 23 (2018): 368-375.
11. R. G. Budynas and J. K. Nisbett. *Shigley's Mechanical Engineering Design*. Vol. 9. New York: McGraw-Hill, 2011.

Appendix A: Shaft Design Project

Design of a Shaft Subjected to Alternating Loads - Part 1

1. [Proof of Concept, Hand Calculation] Consider a machined steel circular shaft subjected to minimum and maximum bending moment of 800 and 5,000 lb .in, respectively; and a minimum torque of 0 and maximum torque of 2,000 lb.in. Consider the factor of safety to be 1.5, the yield strength of the materials 60 ksi, and its ultimate strength to be 80 ksi. Determine the minimum diameter of the shaft for the case where D is 20% larger than d, the fillet radius is 10% of d, and reliability is 99.9%. You may use DE-ASME formula.

2. Write a computer code that can take T_m , T_a , M_m, M_a, choice of material, safety factor, and other necessary assumptions from the user and return an estimated diameter for the shaft. The choice of user inputs is important. Note that many of the required quantities are themselves functions of the shaft diameter. The program cannot ask the user to find different parameters (such as stress concentrations) from the charts and enter them into the program (the code itself should determine these parameters, based on the user inputs). You may check the functionality of your code based on the previous hand calculation in the Proof of Concept section. Your code should be submitted with a report.

Report Guidelines. Similar to other projects that you complete, you have to submit a brief report explaining different parts of the project and discuss your results. In this case, your report should discuss your shaft design code and the results that it produces with different inputs. Your report should typically include the following sections:

1- Abstract (optional) a maximum of 200 word abstract of your report.
2- Introduction: You explain the nature of the problem, the origin of the equations used, the challenges to be met in the projects, a review of the other/similar methods used, and your approach to resolving the challenges.
3- Results: in case of this project, you discuss the algorithm of your code, different sections of the code (material selection, stress concentration and geometry consideration, endurance limit determination, trial and correction procedure, fatigue failure design criteria, check for yielding, …), and some numerical examples of your code results with different inputs and first guesses.
4- Discussion: here you discuss the results you get from the code, that includes how your code converges to final answer, how the converging to final diameter changes with different input, how your code exit from the trial and correction loop, the strength of your code and the advantages that it will give to user in shaft design.
5- Conclusion and closing remarks. Please adopt a professional and technical tone for your report.

Design of a Shaft Subjected to Alternating Loads using an Open AI Interface - Part 2

As this project's second and last segment, we want to explore if we can use an open artificial intelligence interface (such as ChatGPT) to produce a multi-step design code, and how we can use these tools to enhance our engineering range.

1. Investigate the possibility of using well-thought prompts in ChatGPT to acquire a code capable of providing shaft design parameters based on the entered loads, material selection, and geometric considerations. Your task is to obtain a functional shaft design code and verify it. You may evaluate the accuracy of the acquired code with the (i) hand calculation of Part 1 and (ii) the code you wrote in part 1.

Tip: Start with simplified conditions and add levels of complexity gradually. For example, start with prompting a basic shaft layout and then iterate to add more details and design parameters.

2. Provide a comparative analysis between your own code and the acquired one. In your analysis, please address the following areas. - The role of appropriate prompting in getting applicable codes and how to iterate on the prompt to enhance the sophistication of the obtained code. - The reproducibility of the code, based on similar prompts. - The comparison between your code's syntax and structure and the code you received from the Chatbot. - Developing the general techniques that a knowledgeable engineer needs to verify AIgenerated codes (critical conditions and pitfalls)

Appendix B: DE-ASME Elliptic Method

The DE-ASME Elliptic Method is used in fatigue analysis to determine the minimum critical diameter d for rotating shafts under cyclic loading, where d is a function of S_e (endurance limit), S_y (yield strength), K_f (fatigue stress concentration factor), M_a (alternating bending moment), T_a (alternating torque), M_m (mean bending moment), T_m (mean torque), and n (safety factor). Mean torque T_m represents the average torque exerted on a shaft over a specific timeframe, while alternating torque T_a captures the fluctuating component of torque during a cycle [11]. Similarly, mean bending moment M_m reflects the average bending effect applied to a structure, and alternating bending moment M_a signifies the dynamic fluctuations in this effect. The safety factor n serves as a multiplier applied to the calculated strength to accommodate uncertainties and ensure a reliable and secure design, taking into account material properties, manufacturing processes, and loading conditions. Based on the Von Mises stress theory, DE-ASME Elliptic Method considers both normal and shear stresses to assess material failure. The analysis involves calculating alternating and midrange Von Mises stresses, representing cyclic variations and average stress over the loading cycle. These values determine the equivalent stress amplitude and mean stress experienced by the shaft.

Endurance Limit: The endurance limit - S_e - is determined by employing Marin factors as:

$$S_e = \frac{1}{2} K_a \, . \, K_b \, . \, K_c \, . \, K_d \, . \, K_e \, . \, S_{ut} \tag{1}$$

where,

K_a: Factor considering the effect of surface finish. The formula is given by:

$$K_a = a \, . \, S_{ut}^{b} \tag{2}$$

in which S_{ut} is the ultimate tensile strength of the material. The specific values of the constants a and b depend on the material and surface finish.

K_b: Factor accounting for the influence of size on endurance limit.

K_c: Factor addressing the impact of load conditions (e.g., bending, axial, or torsional). K_c is equal to *1* for bending and torsion loads.

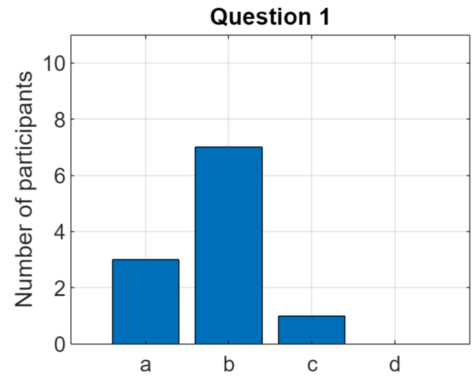
K_d: Factor incorporating the effect of temperature on the endurance limit. K_d is equal to *1* for design at room temperature.

K_e: Factor considering the impact of reliability and statistical variations on the endurance limit. K_e is equal to *0.753* for *99.9%* reliability.

Appendix C: The Survey and Results

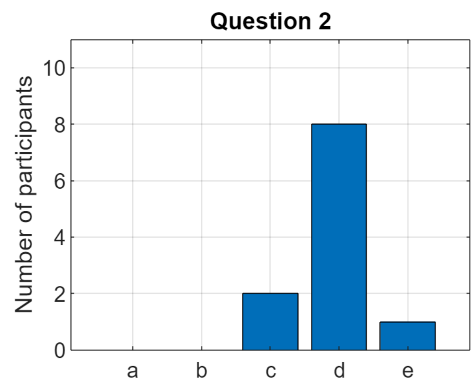
1. How much experience did you have with ChatGPT prior to this assignment:

a) No experience
b) Very little experience
c) Moderate experience
d) Extensive experience



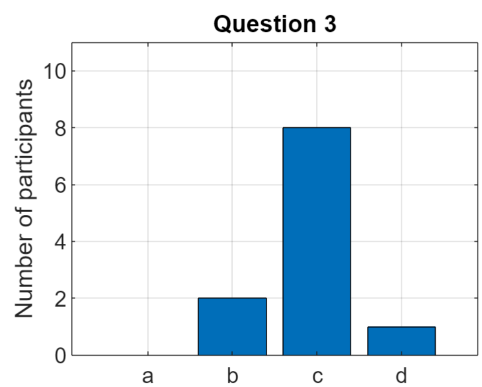
2. How easy was it to understand the syntax and structure of the code generated by ChatGPT for shaft design?

a) Very difficult
b) Difficult
c) Neutral
d) Easy
e) Very easy



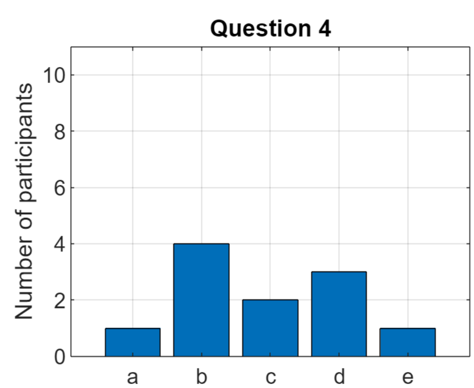
3. To what extent did the code generated by ChatGPT require modification to become functional for shaft design?

a) No modification was required.
b) Minor modifications were required.
c) Major modifications were required.
d) The code generated by ChatGPT was not functional for shaft design and required significant reworking.

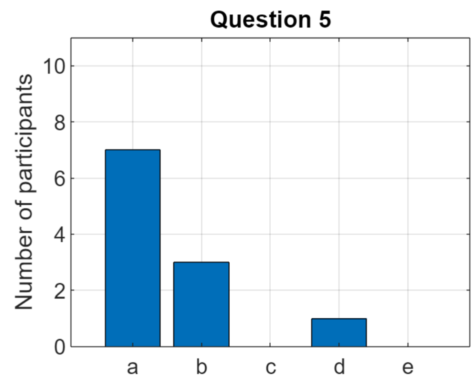


4. How reproducible were the codes generated by ChatGPT for different but similar prompts for shaft design?

a) Not at all reproducible
b) Somewhat reproducible
c) Neutral
d) Reproducible
e) Very reproducible

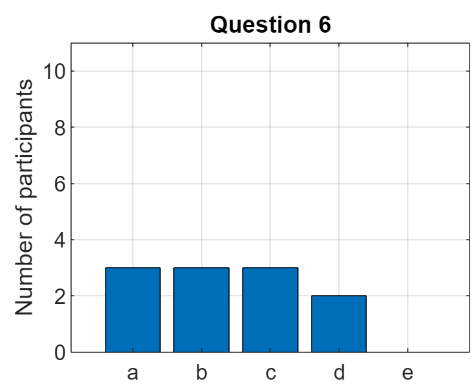


5. How would you compare the length of the code generated by ChatGPT for shaft design with your own code?



a) The ChatGPT-generated code was significantly shorter than my own code.
b) The ChatGPT-generated code was somewhat shorter than my own code.
c) The ChatGPT-generated code was of similar length to my own code.
d) The ChatGPT-generated code was somewhat longer than my own code.
e) The ChatGPT-generated code was significantly longer than my own code.

6. Did ChatGPT provide useful insights into the design process that you hadn't considered before?



a) Not at all useful
b) Somewhat useful
c) Moderately useful
d) Very useful
e) Extremely useful

Please elaborate on your answer:

- ChatGPT was primarily useful as a way to compress code and computation time. In terms of mechanical design, you need to prompt the AI to give the user choices, such as material, temperature, etc. which influence the diameter of the shaft. It was generally incapable of making those decisions on its own.

- Utilizing Chat GPT gave me an alternate perspective on how some elements of shaft design could be implemented via Matlab code. There were approaches, syntax, loop structures and equations that Chat GPT came up with that I did not.

- Insights were more useful on the code syntax/structuring side rather than the conceptual side.

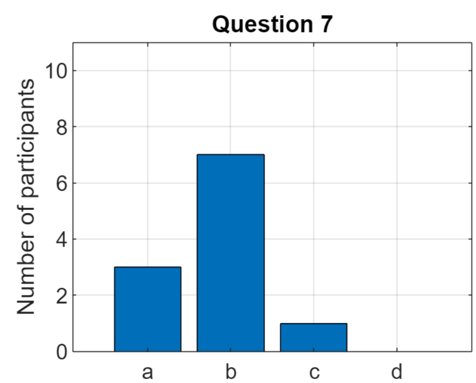
- Not in particular, it just used different forms of the same formula and coded more efficiently. There was no additional insight provided into other methods.

- I mostly prompted ChatGPT to create a general framework for my code, and I'd fill in all the details myself. ChatGPT was decent at creating that framework and from what I remember, it made that process much easier.

- ChatGPT was able to generate a more efficient block of code to open and read spreadsheet material data and for factors a and b, although this required running this prompt 3 times for a successful output. It was also able to generate code for a pop-up window that allowed user input for reliability, but I had to copy this block to generate the other functioning user input pop-up windows necessary for this project.

7. Did you encounter any difficulties or limitations when using the ChatGPT for developing the shaft design code?

a) Yes, I encountered significant difficulties/limitations
b) I encountered some difficulties/limitations
c) Neutral
d) I did not encounter any difficulties/limitations



Please elaborate on your answer:

- I was unable to have ChatGPT correctly include all the marin factors, it was generating extraneous equations for the stresses involved, it used the wrong formula for SE' and DE-ASME Elliptic, it often indexed by different initial variables in the same output generating many off-by-one errors, forgot to initialize variables, and when I attempted to correct these errors it would often correct the error I asked for while overwriting previously correct sections and generating new errors despite me asking to preserve previous work.

- ChatGPT is currently unable to generate code which can accurately compute what you need. It will always require modification and revision before it is usable. Plus it is very bad at pulling equations from textbooks and writing code based on that.

- Chat GPT had trouble interpreting certain charts and figures containing reference values necessary for specific approaches. There were also many instances where the generated code either did not return a fictional code.

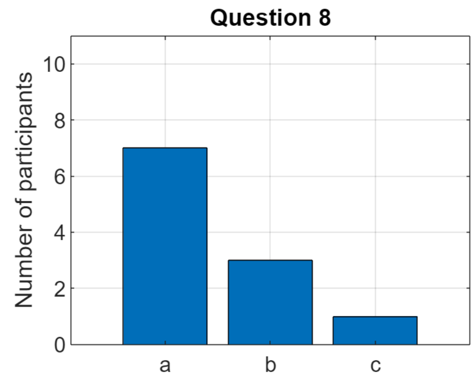
- In my project, I didn't focus on forcing "good" code out of ChatGPT as I did not have enough knowledge about the AI to reap its full benefits. That being said, typing various prompts into the chatbot wasn't too difficult.

- ChatGPT would consistently produce code filled with errors or frequently contradict itself. Since it was prone to corrections, it was difficult to grasp what was right and what was incorrect. It also has issues remembering what was asked so if I wanted to ask it to fix a part of generated code, I had to do this within the next prompt.

- The code often left out key elements of shaft design even if I clearly stated the requirements in the prompts. It could only handle so many prompts at once.

- ChatGPT was not very successful in implementing specific functions or the iterative loop. It also did not choose correct values of stress concentration and other things that are typically found in literature.

8. Would you consider using ChatGPT-generated code as a base for future design assignments in your engineering coursework, which you would edit or modify to become functional for specific design tasks?



a) Yes, I would definitely consider using ChatGPT-generated code as a base for future design assignments.
b) I would consider using ChatGPT-generated code as a base for future design assignments, but I would also like to have the option to complete assignments without ChatGPT assistance.
c) No, I would not consider using ChatGPT-generated code as a base for future design assignments.

Please elaborate on your answer:

- Eventually, AI-generated code will become prevalent and it will be important to learn how to generate, check, and modify the provided code.

- The end result was surprisingly elegant after some tweaks and adjustments.

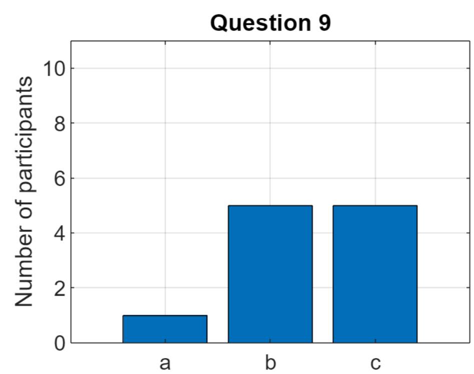
- Practically, I feel like using ChatGPT to devise the "skeleton" of my code is not something I would do, despite it being particularly effective at that. Every person has their own style of coding, and I feel like constraining myself with an arbitrary structure can only lead to frustration and less familiarity with the code. I also think that ChatGPT uses functions that I am not familiar with which can lead to problems with debugging the code. I thought this project was pretty interesting and unique — I had never done anything similar in any other class at UCSD — however the ChatGPT aspect of it seemed sort of orthogonal to the point of the Machine Design class. This also seemed to be the consensus of my friends who were not in the class after I showed them the project prompt. I think I would have preferred more time spent in class on gear design for example, as it made up a decent portion of the final exam, or another topic pertinent to professional engineering, such as GD&T.

- Very often, the generated code either has errors or doesn't do what I want it to do. It would be the same amount of time if I just wrote the code.

- It can be a good way to start an assignment if you don't know where to start, but it is pretty unreliable.

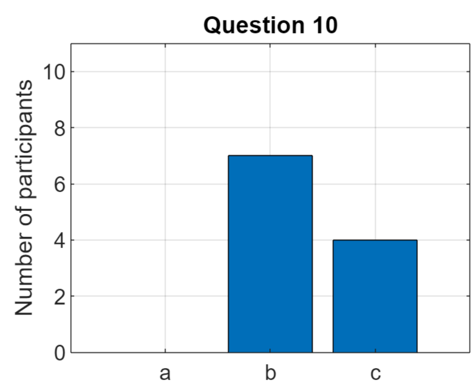
- ChatGPT made it very easy to get started on the code by creating the general framework. All I had to do was fill in the details and write specific functions. It definitely made the process easier and faster, and also helped with debugging.

9. Do you think a person with inadequate knowledge of machine design could use ChatGPT to design a shaft?



a) Yes, ChatGPT is user-friendly enough that someone with little knowledge of machine design could use it successfully.
b) Maybe, with some guidance and support from a knowledgeable engineer or instructor.
c) No, ChatGPT is not user-friendly enough for someone with little knowledge of machine design to use successfully.

10. What level of machine design knowledge do you think is necessary to ensure that the ChatGPT-generated code for shaft design is accurate and applicable?



a) Basic knowledge of machine design concepts and principles.
b) Intermediate knowledge of machine design, including an understanding of different machine elements and their functions.
c) Advanced knowledge of machine design, including expertise in designing and optimizing complex machine systems.