Towards Designing an Interactive System for Accelerated Learning and Assessment in Engineering Mechanics: A First Look at the Deforms Problem-solving System

Mr. Arinjoy Basak, Virginia Polytechnic Institute and State University

Arinjoy Basak is a PhD student in the Department of Computer Science at the Virginia Polytechnic Institute and State University, advised by Prof. Clifford A. Shaffer. He obtained his Bachelors in Computer Science from the Indian Institute of Engineering Science and Technology, Shibpur in 2016, and his Masters in Computer Science from Virginia Tech in 2019.

Mr. Todd Patrick Shuba, Virginia Polytechnic Institute and State University

Todd P. Shuba is a New Horizon Graduate Scholar in the College of Engineering, as well as a Graduate Research and Teaching Assistant in the Department of Engineering Education, at the Virginia Polytechnic Institute and State University. His research interests include transfer of learning, collaborative learning, and student motivation and engagement. He holds a Bachelor of Science in Engineering with a concentration in Environmental and Ecological Engineering and a minor in Mechanical Engineering, as well as a Master of Science in Education with a concentration in Educational Psychology and Research Methodology, from Purdue University-West Lafayette.

Mr. Jianqiang Zhang, Virginia Polytechnic Institute and State University

Dr. Sneha Patel Davison, Virginia Polytechnic Institute and State University

Sneha is an instructor in the Biomedical Engineering and Mechanics Department at Virginia Tech. She earned her Bachelor's of Science, her Master's of Science, and her Doctoral degree from the Engineering Mechanics department also at Virginia Tech. Her research interests include exploring the most effective methods to teach students introductory level mechanics, especially in the large classroom environment.

Prof. David A. Dillard, Virginia Polytechnic Institute and State University

David Dillard is the Adhesive and Sealant Science Professor in the Biomedical Engineering and Mechanics Department at Virginia Tech. He has worked extensively in the field of adhesive bonding, having experience in structural adhesives for aerospace, automotive, and infrastructure applications; adhesives and coatings for microelectronic applications; pressure sensitive adhesives; elastomeric adhesives and sealants; and polymeric membranes. He has authored or co-authored over 195 refereed publications and regularly teaches courses in adhesion science, polymer viscoelasticity, and sustainable energy solutions. With nearly 40 years of experience as an educator, he is interested in new ways to leverage technology to have a positive impact on student learning and assessment.

Dr. Jacob R. Grohs, Virginia Polytechnic Institute and State University

Jacob Grohs is an Assistant Professor in Engineering Education at Virginia Tech with Affiliate Faculty status in Biomedical Engineering and Mechanics and the Learning Sciences and Technologies at Virginia Tech. He holds degrees in Engineering Mechanics (BS, MS) and in Educational Psychology (MAEd, PhD).

Dr. Nicole P. Pitterson, Virginia Polytechnic Institute and State University

Nicole is an assistant professor in the Department of Engineering Education at Virginia Tech. Prior to joining VT, Dr. Pitterson was a postdoctoral scholar at Oregon State University. She holds a PhD in Engineering Education from Purdue University and other degrees in Manufacturing Engineering from Western Illinois University and a B.Sc. in Electrical and Electronic Engineering from the University of Technology, Jamaica. Her research interest is eliciting conceptual understanding of AC circuit concepts using active learning strategies.

Prof. Clifford A. Shaffer, Virginia Polytechnic Institute and State University

Clifford A. Shaffer received his PhD in Computer Science from University of Maryland, College Park in 1986. He is currently Professor of Computer Science at Virginia Tech, where he has been since 1987. He directs the OpenDSA project, whose goal is to provide a complete online collection of interactive tutorials for data structures and algorithms courses. His research interests are in Digital Education, Algorithm Visualization, Algorithm Design and Analysis, and Data Structures.

Towards Designing an Interactive System for Accelerated Learning and Assessment in Engineering Mechanics: A First Look at the Deforms Problem Solving System

Abstract

Repeated deliberate practice has been shown to be vital to developing mastery in engineering problem solving. Online tutoring systems have enhanced learning experiences, and delivered content tailored for specialized fields. Motivated by the aim of improving students' problemsolving skills, we created an interactive system for use in an undergraduate introductory engineering mechanics course required for many engineering disciplines. Our system provides an intuitive, visual framework that allows students to rapidly solve problems that require building systems of equations in multiple steps. Built within the OpenDSA eTextbook system, these exercises can be served directly through a learning management system such as Canvas, allowing the exercises to be integrated seamlessly with other content. In this paper, we describe the key design choices for our system, present important features and the student workflow, and describe support for targeted feedback and analysis for the instructors. We present our plans to evaluate the system, and discuss the results of a preliminary usability study.

1. Introduction

Recent studies show problem-solving ability is being increasingly prioritized as a core aspect of engineering curriculum and a fundamental competency demanded by employers. However, not only are problem-solving activities time consuming for students, they are also often difficult to assess beyond simply checking for correctness of the final answer. In addition, it is difficult in a classroom setting to deliver useful feedback [1]. While popular online tutoring frameworks exist [2], [3] that support structures for mathematical problem solving and some elementary level of feedback, most suffer from significant drawbacks in terms of design and usability that introduce a significant overhead from learning to use the system. Additionally, they lack flexibility for supporting a variety of problem-solving approaches, and underlying support for analyzing solution approaches. We aim to create a learning framework that can be used to deliver interactive, automatically assessed problems for engineering mechanics. The aim is to capture students' problemsolving processes, quickly assess correctness, and provide accurate feedback through targeted hints or explicit feedback. This approach accelerates and enhances learning by reducing algebraic tedium and emphasizing deliberate practice [4], which has been shown to enhance problem-solving accuracy and motivation to learn more [5], [6].

The Deforms problem-solving system provides an intuitive, visual framework that allows students to rapidly solve problems that require building systems of equations in multiple steps. Students can create systems of equations from palettes of existing equations and create variable and value associations from the problem texts and figures as necessary to solve single-step problems, or multistep problems with intermediate solutions. The students see exercises containing prose, figures, and submission boxes, as well as workspaces and equation palettes that students can interact with through simple click-and-drop actions. Built within the OpenDSA [7] eTextbook system, these exercises can be served directly through learning management systems such as Canvas, allowing the exercises to be integrated seamlessly with other courses. We present the details of the current iteration of our system.

As proposed in [8], one of the primary goals of our project is to systematically study how technology-rich environments can enhance the learning, teaching and assessment of complex

knowledge. Inadequate development of conceptual knowledge, cognitive overload while learning, and the limitations of individualized feedback in large classes are widespread. A key aspect of learning sciences is to understand how students interrelate and organize knowledge in the domain, and the critical factors that help or affect such active construction, as this distinguishes experts and novices. According to cognitive load theory (CLT), for learning to occur, working memory needs to accommodate the additive needs of intrinsic, extraneous, and germane cognitive loads [9]. From this perspective, interactive exercises empower the user to optimize their own learning through the ability to decrease intrinsic cognitive load of the problem, allowing identification of what they know and what they don't, as well as provide opportunity for metacognitive reflection – all of which has been shown to increase development of more complex knowledge [10]. When done properly, educational technologies and e-learning environments can greatly optimize the elements of CLT for effective learning [11]. Correspondingly, our system also logs student interactions such as click events that can be analyzed to provide data-driven reflections for both students and instructors to focus their efforts on understanding misconceptions.

We also present the results of a usability study that we conducted in Fall 2020 with students from an undergraduate course in engineering mechanics of materials at a major R1 university. Feedback from this preliminary study shows positive results, indicating that the students found many of the features helpful, and that the system mapped well to their traditional pen-and-paper experiences of solving problems. We also obtained valuable feedback to guide design choices for future iterations of our software.

2. Prior Work on Relevant Tutorial/Interactive Exercise Systems

The system that is closest to ours in terms of functionalities provided and approach is Andes [3] by Van Lehn et. al. Like our system, Andes captures equations entered by students, supports variable associations, and reduces algebraic tedium. We drew from the idea of providing short hints at the student's request to design a preliminary version of our guidance system. Despite the positives of the Andes system and the lessons learned from it over the years on student interactions, it has drawbacks. The interface is cumbersome and involves entering definitions of terms directly, as well as constructing equations by typing them in - which while helpful, requires adherence to syntax rules. In contrast, we used a palette-entry approach that is popular in commercially available software. This approach helps reduce the complexity of verifying solutions [12].

Another approach to creating similar tutorial systems uses ideas from mathematical model construction and exploration - the oldest such system being Stella [13], [14], which was a graphical stock-and-flow notation-based editor used in mathematical model exploration tasks. Similar systems are Model-It [15], also an editor, and NetLogo [16], which focused on agent-based modeling. Co-Lab [17] worked as a tutor by introducing feedback and hints as students asked for help when constructing models, but it's learning gains were never evaluated. Dragoon [2], also developed by VanLehn et. al., focuses on teaching students mathematical modeling through construction of computational models of dynamical systems for general science topics. The interface draws from Stella, by creating and connecting nodes in a directed graph to create flows and dependencies among named variables, and observe how they affect each other. Dragoon offers four different modes: Editor, Test, Coached, Immediate feedback. This is an idea we intend to incorporate into our system as our feedback support matures, such that the instructor can adjust how much feedback, and what kind of feedback they want to provide to the students.

For our context, the approach used by Dragoon would not be ideal. Dragoon requires learning the formal language of notations, which is somewhat detached from the typical mathematical notation used for teaching concepts. Van Lehn [18] mentions achieving notational mastery in translating model descriptions into notation, which we feel places an additional cognitive load on the student. This approach makes it difficult to construct complex systems of equations required in our problems. In our case, the learning curve with respect to notation is small since the users already know how to construct systems using mathematical notations, so our system provides a fast alternative for construction that allows for rapid deliberate practice and feedback.

We provide a fast, efficient method to construct systems of equations to represent physical quantities. This allows students to rapidly build and test their understanding while receiving targeted feedback, potentially on significantly more and varied problems than in comparable time with pen and paper.

3. System Overview

Exercises are delivered through a learning management system like Canvas, or individually as standalone exercises. Each exercise consists of an HTML file containing the formatted problem prose and a reference to the solution file corresponding to the current problem. The core of the system (comprising the interface interactions and the solvers) is implemented in JavaScript and built on top of existing frameworks used by the OpenDSA eTextbook system such as JSAV [19] for visualizations and the KhanAcademy exercise framework [20] for delivering individual exercises. Open-source libraries for unit conversions and performing complex mathematics with physical quantities and mathematical solvers are used, but apart from these, the codebase entirely consists of libraries already part of OpenDSA.

Figure 1 shows an example exercise. We can divide the interface into three main parts – the problem prose (together with the submission boxes, and check answer buttons), the workspace/solver area, and the notifications section.

We use click-and-drop as the main interaction idiom [21]. Point-and-click interactions are generally known to be better than drag and drop [22], and drag and drop has been shown to be more prone to errors, especially at the beginning and ending of an interaction [23]. We opted to keep all direct functionalities in the open (that is, not hidden behind cascades of menus). To aid newcomers and beginners, interactive items are designed to highlight on hovering, and display text snippets summarizing their functionality and how to interact. We also provide in-system help/tutorial material for every interface element (submission boxes, workspaces, equations, palettes, etc.), accessible by clicking on the appropriate question-mark icon for that element (7). Clicking this opens up a dialog box with a quick text overview of the element, its common features, and is accompanied by animations to show how they work.

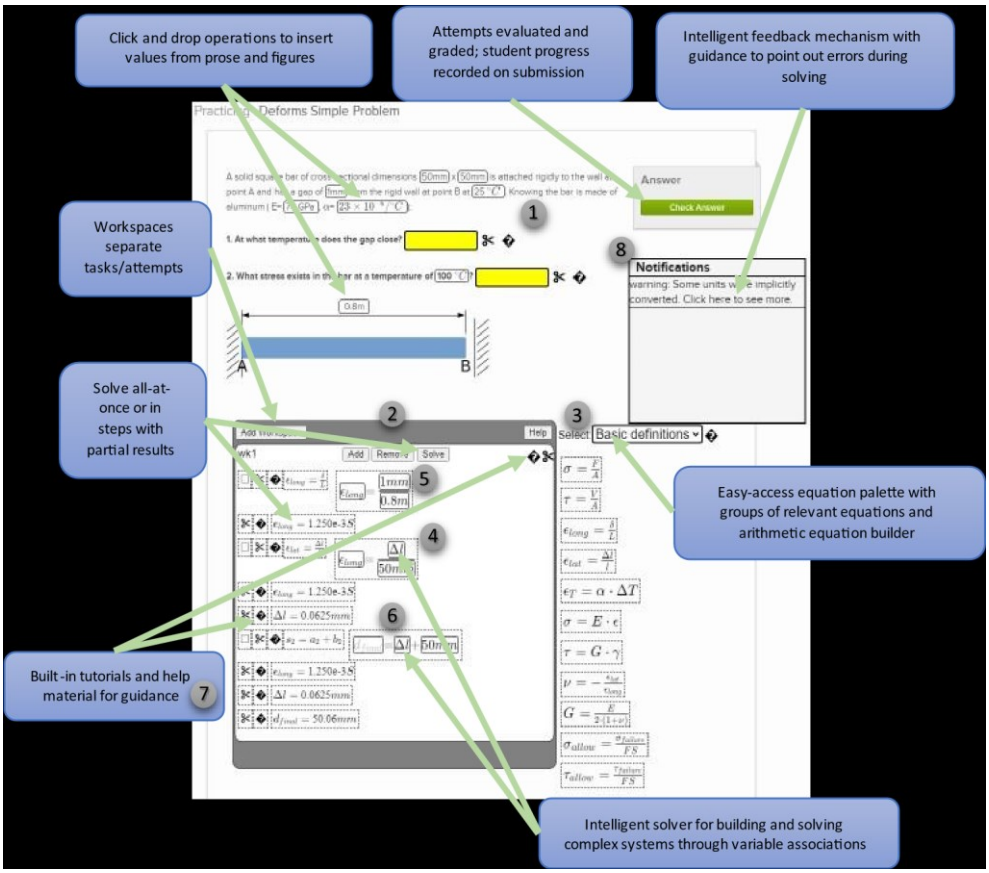

Figure 1: An outline of the problem-solving interface with the different features offered

The problem prose (1) contains the text describing the problem setting (including sub-parts and diagrams). The interactive parts of the problem prose include the physical quantities embedded in the prose, which can be used to solve the problems by adding them to equations (4) in the workspace (2). In addition, each subpart has a yellow box next to it which accepts candidate answers calculated inside the workspaces. The boxes can be filled by clicking on a value calculated in a workspace and then clicking on the box. The boxes can be cleared by clicking on the scissor icons. Once candidate answers have been added, the user clicks on the "Check Answer" button, which validates whether the answers were right or wrong.

The workspaces (2) are the main areas where equations are entered, known values associated with parameters from equations, and relationships defined to enable solving for unknown variables. Their purpose is to logically separate different sets of problem-solving steps (such as subparts of the same problem), or alternative approaches for the same problem. The workspaces are variable in size, and expand and contract to accommodate the equations being solved and the solutions computed (value boxes, see bottom of workspace in Figure 1). Additionally, deleting a workspace removes all the equations and solutions contained in it, effectively creating a clean slate. Each workspace has an Add, Remove, and Solve button. The Add button is used to add selected equations from the equation bank (3) into respective workspaces, doing which expands a newly added equation into a collection of elements as shown in Figure 1. The Remove and Solve buttons, respectively, work with equations in a workspace as we describe later.

The equation palette (3) is a persistent, drop-down list of equations segregated by topics covered by the subject that the exercises belong to. In addition to equations that define relationships between physical quantities, a group of equations for defining algebraic relations (such as sums, differences, divisions, etc.) is also provided. Additionally, we provide a "Favorites" list to

bookmark equations that the student has already used in the current session. Palette-based entry has gained popularity in modern software interfaces since they not only improve user experience by reducing algebraic tedium, but also make it easier to identify student intent when analyzing problem-solving behavior later, compared to more open-entry methods.

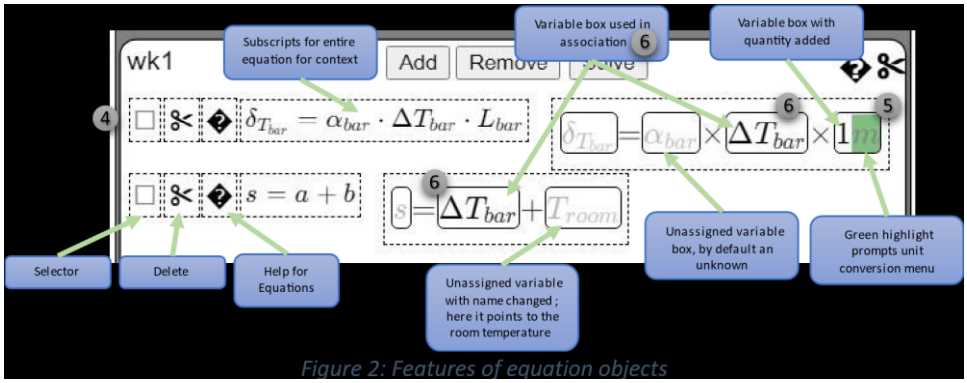

Figure 2: Features of equation objects

Equations (4) form the main workhorse for our system. Each equation added from the equation palette expands into a collection of icons and representations as shown in Figure 2. The checkboxes are used to select equations to be solved as a set (click on "Solve"), or deleted ("Remove"). This allows the user to solve multiple steps in the same workspace, or evaluate smaller sets of equations as part of a bigger problem. Each equation has an enlarged interactive section with greyed-out variable boxes, which can accept quantities from the problem prose. Adding a quantity to this box also enables converting quantities to different units on prompt (5).

We finally talk about creating unknowns and variable associations (6). Variable associations involve selecting multiple boxes in different equations to create systems of equations in more than one unknown. By default, we create equations in one unknown by populating all but one box in a single equation. However, to create a system, click on an empty box to start an association from its context menu, and then click on a second box. Once created, more variables can be added to this association from the context menu of any variable box. Moreover, students have the option to customize the names of these variables in an association, single unknowns, or all variables in an equation to allow for readability and context (as shown in Figure 2).

We now discuss briefly the underlying solver engine for the system, which relates to our guidance system (8 in Figure 1), as outlined by the Notifications panel in Figure 3. We use an opensource solver library called Nerdamer [24] to solve systems of equations constructed in the interface, and math.js [25] for working with physical quantities. Our underlying solver code involves tying these two together through preprocessing, with our guidance code using this preprocessing phase and the post-solve results to give feedback to the students about their attempts. We designed our notifications system – the current iteration of our guidance system – to provide targeted, brief feedback, which is shown to be more useful than long feedback [26], directing the students to address specific issues that lead to incorrect/erroneous computations. However, hints have been criticized in the past for offering more "just-in-time learning". To that end, our system currently provides only feedback on parts of the system that are blatantly wrong. We aim to account for these precautions when designing our guidance system in the future to offer in-depth feedback about steps taken.

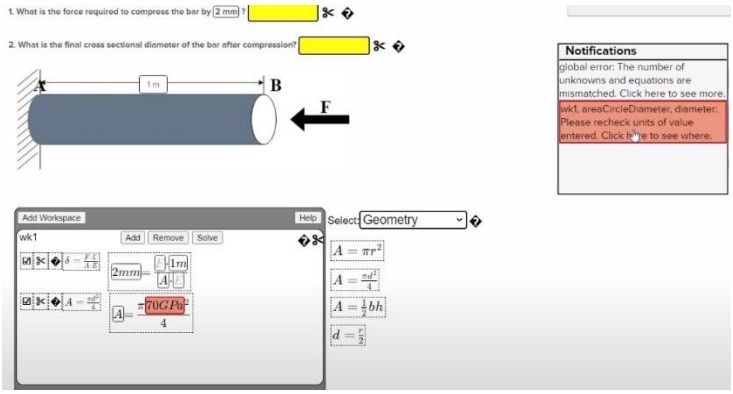

Figure 3: Example of the guidance system in action

The guidance system works by preprocessing the equations, quantities entered and the associations made to feed into the numerical solver. It currently checks for inconsistencies in the equivalence of units of the numerical quantities on both sides of equations, whether quantities are in right places in an equation from the palette involving physical quantities, and whether the number of unknowns and equations match. If this phase passes properly, then the equations are sent to the solver, following which the results are output. Any errors which occur at any phase of these two phases are immediately reported in the notifications panel, where clicking on individual errors points the user to where the error occurred (which could be associations, a variable box with quantities, an equation itself, or a general error message).

In addition to error checking, our solver preprocessor also handles implicit unit conversions for quantities in equations having inconsistent orders of magnitude (for example, m vs. mm in SI units), and handles inferring units of unknown quantities computed from the system. Any errors in attempting to infer the domain of an unknown quantity (due to inconsistencies in creating associations) are flagged as errors and immediately reported, although implicit conversions are reported as warnings at the moment. However, in future iterations, these would be placed under the control of the instructor, who can choose how much guidance to provide, and whether implicit conversions should be allowed to support the student or not.

We provide an example of how a student would work with this system in a video at https://youtu.be/YTVuf2ahdm0. A typical workflow, considering a student is familiar with the interface, would start with reading the problem prose, followed by browsing the equation palette to find the appropriate equations and adding them to a workspace. These equations would be populated with quantities either from the prose, or custom entered from the context menu for an empty box. Associations can be created as outlined earlier. Finally, once all the equations are created, they can be selected and solved. This is repeated as required to create candidate solution boxes in the workspace, which are then selected and added to submission boxes in the question.

4. Usability Study

4.1. Study description

We conducted a preliminary study of the problem-solving system with students in an undergraduate engineering mechanics course at a major R1 university. The topic of the course was ‘Mechanics of Deformable Bodies’ (hence the name Deforms problem-solving system). Two studies were conducted with the same system in the Fall and the Spring semester. The studies presented four problems and six problems respectively. Problems were served through Canvas. These problems covered topics of increasing difficulty from throughout the semester (mechanical properties of materials, axially loaded members, statically indeterminate problems).

The problems used in Spring included the four from Fall and two additional problems of easy and medium difficulty, covering similar topics.

A short one-page tutorial on how the system works and a video demonstration of the system on a sample problem (that is not part of our problem set) was provided to the students to help them get familiar with the system. After solving the problems, students were asked to complete a survey to provide feedback on the usability of the system. The survey tells us about students' experiences with the system, and suggests directions for future development. We also collected student interaction data from tracking their interactions with the system (button clicks, submitted attempts, etc.) to help obtain a preliminary understanding of the problem-solving processes.

The survey asked a total of 26 questions, 18 of which were 5-point Likert scale questions, two of which required them to choose between two alternatives, and six qualitative questions to allow students to provide detailed feedback. The Likert scale questions are presented together with the responses in Figure 4. The qualitative questions were:

- Q1. Please include any additional guidance material that you would like to see to make it easier to learn or find one's way around the system.
- Q2. Please include any specific bugs encountered that hindered your progress while using the system.
- Q3. Please include any other comments about the design of the interface and if there are any new features or changes that would be of value.
- Q4. Please include any other feedback that you would like to see to guide you on how you solved the problem.
- Q5. Please include other interactions that you would like to see. (If student answered "No" to the question "Were the click and drop interactions enough to perform the necessary tasks efficiently?")
- Q6. Please include any special features that would help make the system faster and more efficient to use.

The survey was offered to the students as an extra credit assignment, whereby the students had the option to receive extra credit by solving problems using the system and optionally completing the survey, or by turning in a solution of the problems on pen and paper as part of a practice problem set. The students were also informed of their choice to not participate in the study or request that their data not be used, and efforts were taken to maintain the students' anonymity at all points. A total of 162 people responded to the study by engaging with the system, as recorded by interaction events. We further assumed that a student who completed at least one problem successfully, or registered at least 200 events in the first study or 300 events the second study, respectively, exhibited earnest engagement by spending enough time with the system, which gave us 96 out of 162. Among these, n=80 students completed the survey, whose responses are tallied and discussed next.

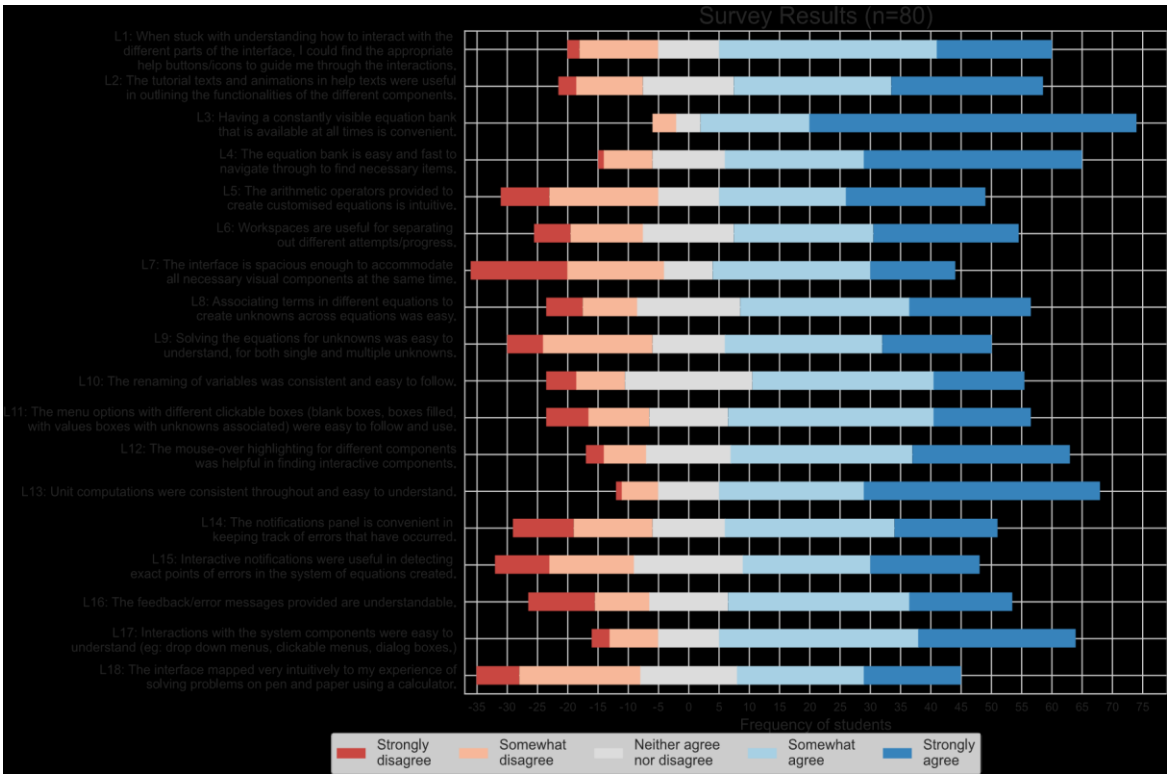

Figure 4: Summary of usability survey responses to Likert-scale questions

4.2. Results and Observations

The Likert-scale responses showed generally positive results, with most people responding with either “Somewhat Agree” or “Strongly Agree” (shown by the light and dark blue bars on the right side of the plot) on most items under questions about the system, indicating that they found the interface usable and the exercises helpful. We summarize the responses in Figure 4.

When asked about tutorial elements provided to guide newcomers and beginners (L1, L2 in figure), users knew they were accessed through help icons in different parts of the interface when needed. In response to Q1 however, there were many requests to provide help material that was more comprehensive in guiding students on how to perform tasks such as entering values in boxes and performing algebra, preferably as a document or a video, which leads to questions about the intuitiveness of the system. Other comments (9/80) included requests to improve the explanations for how the feedback system worked, and to redesign the tutorial to be easier to search for answers. The video in Section 3 was also provided as part of the tutorial in the second iteration of the study, and received generally positive reviews. 59 out 80 students either did not report anything for this question, or reported positively with regards to the help material provided. Overall, the in-system tutorial features and the additional tutorial material provided were found to be helpful.

Several questions pertained to interface design (L3-L10). The persistent equation bank was received favorably by students (L3), which is confirmed by L4 that it made it easier to find and add equations. However, 11/80 responses to Q3 requested reorganizing the workspace to optimize work area, which shows that it competes for space with other elements on-screen. Similarly, while the concept of using workspaces to separate out work done received moderately positive reviews (L6), the results were varied when we asked about the interface being spacious enough (L7) - and the responses to Q3 and Q6 supplement this further. Thus, we must consider trade-offs in future designs to maximize on-screen work area while keeping features that received favorable responses. Several students reported their work being much better organized

when using our system, which indicates success for our core design choices for the workspace environment.

We received mixed reviews about the equations supported in the palette (L5, Q3, Q6). Q3 received 12/80 responses and Q6 received 18/80 responses that explicitly asked for more flexibility in constructing algebraic equations. Many exercises not only require specific equations with physical quantities that can be entered through palettes, but also simple algebraic manipulations to combine intermediate solutions that require more traditional text-entry approaches. This becomes another trade-off to consider in future design choices. When asked about features used to create systems of equations (L8-10), there was almost unanimous agreement that creating associations as well as solving for individual variables was intuitive and helpful, indicating that this design choice worked. Our qualitative responses (Q3) show reports on how certain functionalities such as adding values, adding multiple equations, or the default behavior of the solve button could be improved, as well as requests for supporting drag-and-drop functionality (10/80). We also noted requests for optimizing workspace area to fit everything on one screen (11/80). Support for computations with units were considered useful (L13), which speaks a lot of positives for our underlying solver.

The notifications panel and the guidance system received generally favorable reviews (L14-16). Most students had nothing specific to say or indicated it was useful in Q4 (46/80). 15/80 responses to Q4 indicated that the notifications provided were ambiguous and not specific towards the mistakes made. This was to be expected, considering that we are on our first iteration of the notifications feature, which itself is a precursor to the future guidance system. We aim to improve this feature to support variable instructor-controlled support, while remaining independent of problems. Along those lines, we also received several suggestions in Q4 (19/80) on how to improve the visuals and the functions of the notifications system.

Together, i) increased flexibility to construct equations, and ii) improving interface design to better use screen space and accommodate elements were the top responses to open-ended questions (particularly Q6 – 18/80 and 11/80 responses respectively), requesting feature ideas the students wanted to see in the system. These concerns were raised in other questions as well, such as Q3 and Q1 (11/80 responses, with 7/11 responses asking for greater flexibility). We plan to address these issues prior to our next roll-out. We received 12/80 responses that suggested additional features, which we also plan to incorporate down the line.

Finally, we asked about overall experiences using the system. Several questions asked about new features the students would like to see included. Few bugs were reported by the users (Q2, 13/80 recorded bugs). The few issues reported were mostly related to glitches in the interface visuals and equation building, both of which we plan to fix, along with interface updates based on responses to Q3, Q4 and Q6. Users generally agreed that the components and operations were intuitive (L11-13). Context menus were helpful, and supportive features (highlighting, texts, etc.) made the system behavior intuitive, echoing L1 and L2. Click-and-drop interactions received favorable reviews (58/80 students reported favorably when asked if this was helpful, 22/80 reported against), although there are some requests for drag and drop operations (Q5), which we also intend to add. Responders also requested the ability to manually enter quantities, much like the other qualitative questions (see responses to Q6). When asked about the layout, students indicated a need for extra navigation, but they would rather have functionality hidden behind menus (51/80 students supported this) to create a compact workspace, rather than have everything out in the open (29/80 students supported this).

Overall, the system received positive reviews when asked how similar their experiences were to that of using a calculator and pen-and-paper format (L18), with the graph skewing towards the positive side (37/80 positive, 27/80 negative, 16/80 neutral; with 16/80 strongly agreeing and 7/80 strongly disagreeing), indicating the success of our first iteration.

4.3. Event log data

We collected clickstream data based on the students' interactions. A total of 111,328 events were collected that recorded values clicked in the prose, equations added, associations created, unit conversions, etc. Most importantly, we capture the following important milestones in the problem-solving process: i) systems of equations solved, ii) errors generated and suggestions provided, and iii) answers submitted their correctness. Figure 5 shows a sample student interaction.

161	26762	"2020-12-07 21:08:37"	deforms-equation-created	""wk1_PoissonsRatio_1_1""
161	26762	"2020-12-07 21:09:12"	deforms-equation-created	""wk1_axialMemberNormalStrainAvgLongdelta_2_1""
161	26762	"2020-12-07 21:09:23"	deforms-association-created	"{"name":"a_d","source":"a_d","target":"a_b"}"
161	26762	"2020-12-07 21:09:23"	deforms-association-name-changed	"{"name":"a_d","newName":"\\epsilon_{long_{ }}"}"
161	26762	"2020-12-07 21:09:38"	deforms-equation-created	""wk1_axialMemberNormalStrainAvgLatDiameter_3_1""
161	26762	"2020-12-07 21:09:45"	deforms-association-created	"{"name":"a_g","source":"a_g","target":"a_c"}"
161	26762	"2020-12-07 21:09:45"	deforms-association-name-changed	"{"name":"a_g","newName":"\\epsilon_{lat_{ }}"}"
161	26762	"2020-12-07 21:10:18"	window-focus	""User looking at 00.01 Deforms module""
161	26762	"2020-12-07 21:10:19"	window-blur	""User is no longer looking at 00.01 Deforms module""
161	26762	"2020-12-07 21:11:07"	deforms-variable-enter-value	""wk1_axialMemberNormalStrainAvgLatDiameter_3_1_changeLength""
161	26762	"2020-12-07 21:11:07"	deforms-variable-add-value	"{"name":"wk1_axialMemberNormalStrainAvgLatDiameter_3_1_changeLength","value":"0.13","unit":"mm"}"
161	26762	"2020-12-07 21:11:39"	deforms-variable-enter-value	""wk1_axialMemberNormalStrainAvgLatDiameter_3_1_length""
161	26762	"2020-12-07 21:11:39"	deforms-variable-add-value	"{"name":"wk1_axialMemberNormalStrainAvgLatDiameter_3_1_length","value":"20","unit":"mm"}"
161	26762	"2020-12-07 21:12:05"	deforms-variable-add-value	"{"name":"wk1_axialMemberNormalStrainAvgLongdelta_2_1_deform","value":"3","unit":"mm"}"
161	26762	"2020-12-07 21:12:05"	deforms-variable-enter-value	""wk1_axialMemberNormalStrainAvgLongdelta_2_1_deform""
161	26762	"2020-12-07 21:12:25"	deforms-variable-add-value	"{"name":"wk1_axialMemberNormalStrainAvgLongdelta_2_1_length","value":"200","unit":"mm"}"
161	26762	"2020-12-07 21:12:25"	deforms-variable-enter-value	""wk1_axialMemberNormalStrainAvgLongdelta_2_1_length""
161	26762	"2020-12-07 21:12:28"	deforms-solver-engine-error	""Unexpected error during solving, check solveSet or debug""
161	26762	"2020-12-07 21:12:28"	deforms-solve-errors	""{\"error\":{\"global\":{}},\"warning\":[]}""
161	26762	"2020-12-07 21:12:28"	deforms-solve-started	""{}""
161	26762	"2020-12-07 21:12:42"	deforms-equation-selected	""wk1_PoissonsRatio_1_1""
161	26762	"2020-12-07 21:12:43"	deforms-solve-errors	""{\"error\":{\"global\":{\"unequalUnknownsEquations\":{\"description\":\"The number of unknowns and equatio
161	26762	"2020-12-07 21:12:43"	deforms-solve-started	""{\"0\":{\"name\":\"wk1_PoissonsRatio_1_1\",\"repr\":{\"equations\":[\"a_a = - a_g / a_d\"],\"unknowns\":{\"a_a
161	26762	"2020-12-07 21:13:16"	deforms-submit-answer-check	""{\"0\":{\"solution\":null,\"decision\":false},\"1\":{\"solution\":null,\"decision\":false}}""
161	26762	"2020-12-07 21:13:26"	window-unload	""User closed or refreshed 00.01 Deforms module""
161	26762	"2020-12-07 21:13:52"	document-ready	""User loaded the 00.01 Deforms module""
161	26762	"2020-12-07 21:14:45"	deforms-equation-created	""wk1_axialMemberNormalStrainAvgLongdelta_1_1""

Figure 5: Student interaction data as captured from our Fall 2020 pilot study

Each solve (deform-solve-started) event captures the system of equations, custom variable names, quantities used, and errors/warnings generated if any – making them the most important events to analyze, as they provide summative snapshots of progress. Looking at the sequence of events provides a more dynamic view of the student's thinking process. This can be helpful towards understanding: i) how a student gets familiar with the interface, and ii) once they're familiar, areas that can be improved to speed up workflow. We defer a more detailed analysis of the log events and student behavior based on this log data to a future paper.

5. Conclusion

In this paper, we introduced the current iteration of a problem-solving system aimed at accelerating learning and enhanced assessment for teaching engineering mechanics topics. The results from preliminary usability studies show generally positive reviews for the system, indicating that further work on improving the system would make this more appealing for students to use as an alternative to traditional pen-paper-calculator setups in practicing problems. Such a system would support increased deliberate practice, allow objective assessment of engineering problemsolving skills, and provide targeted feedback to improve understanding of core concepts.

There are several limitations to our current study. The questionnaire needs to be updated to extensively cover user experiences in more detail, as opposed to just feedback on design decisions, following a formal formative analysis. For example, we do not have concrete answers on whether features for customizing equations and variables through renaming and subscripting were useful or not. More user surveys are required to better understand the pros and cons of the current design, and new features needed. We also need to evaluate how this system improves student proficiency in core concepts for each topic that they solve exercises for. This would need a detailed study in parallel with the usability studies that we conduct. We also need a better idea

of student activity events to capture for better analysis of problem-solving activities and student progress. We aim to address all of these issues in future studies and successive iterations of the system.

6. Acknowledgements

Supported in part by the National Science Foundation under Grant No. DUE 1841980. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Soledad and J. Grohs, "Understanding Faculty Experiences in Teaching Large Classes: A Pilot Study on Faculty Perceptions of Teacher-Student Interaction in Foundational Engineering Courses," in *The 2nd Annual Teaching Large Classes Conference*, 2016.

[2] K. VanLehn, J. Wetzel, S. Grover, and B. Van De Sande, "Learning how to construct models of dynamic systems: an initial evaluation of the dragoon intelligent tutoring system," *IEEE Trans. Learn. Technol.*, vol. 10, no. 2, pp. 154–167, 2016.

[3] K. VanLehn *et al.*, "The Andes physics tutoring system: Five years of evaluations," 2005.

[4] K. A. Ericsson, R. T. Krampe, and C. Tesch-Römer, "The role of deliberate practice in the acquisition of expert performance.," *Psychol. Rev.*, vol. 100, no. 3, p. 363, 1993.

[5] J. R. Grohs, T. Kinoshita, B. J. Novoselich, and D. B. Knight, "Exploring learner engagement and achievement in large undergraduate engineering mechanics courses," in *ASEE Annual Conference and Exposition*, 2015.

[6] J. R. Grohs, D. K. Maczka, M. Soledad, and K. K. Bagalkotkar, "Exploring the feasibility of an educational computer game as novel means of assessing problem-solving competencies," *ASEE Comput. Educ. J.*, vol. 7, no. 4, p. 95, 2016.

[7] C. A. Shaffer, "Opendsa: An interactive etextbook for computer science courses," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 2016, p. 5.

[8] N. P. Pitterson *et al.*, "Accelerated learning and assessment in engineering mechanics: Designing an interactive tool to support students' learning," in *ASEE Annual Conference and Exposition, Conference Proceedings*, 2020, vol. 2020-June.

[9] J. Sweller, J. J. G. Van Merrienboer, and F. G. W. C. Paas, "Cognitive architecture and instructional design," *Educ. Psychol. Rev.*, vol. 10, no. 3, pp. 251–296, 1998.

[10] F. Paas, A. Renkl, and J. Sweller, "Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture," *Instr. Sci.*, vol. 32, no. 1/2, pp. 1–8, 2004.

[11] C. J. Egelhoff and K. L. Burns, "A Heuristic to Aid Teaching, Learning and ProblemSolving for Mechanics of Materials," in *American Society for Engineering Education*, 2011.

[12] R. Ranganathan, K. Vanlehn, and B. Van de Sande, "What do students do when using a step-based tutoring system?," *Res. Pract. Technol. Enhanc. Learn.*, vol. 9, no. 2, pp. 323–347, 2014.

[13] H. M. Doerr, "Stella ten years later: A review of the literature," *Int. J. Comput. Math. Learn.*, vol. 1, no. 2, pp. 201–224, 1996.

[14] B. Richmond, "STELLA: Software for bringing system dynamics to the other 98%," in *Proceedings of the 1985 international conference of the System Dynamics Society: 1985 International system dynamics conference*, 1985, pp. 706–718.

[15] S. J. Metcalf, J. Krajcik, and E. Soloway, "Model-It: A design retrospective," *Innov. Sci. Math. Educ.*, pp. 77–115, 2000.

[16] S. Tisue and U. Wilensky, “Netlogo: A simple environment for modeling complexity,” in *International conference on complex systems*, 2004, vol. 21, pp. 16–21.
[17] C. Bravo, W. R. van Joolingen, and T. de Jong, “Using co-lab to build system dynamics models: Students’ actions and on-line tutorial advice,” *Comput. Educ.*, vol. 53, no. 2, pp. 243–251, 2009.
[18] K. VanLehn, G. Chung, S. Grover, A. Madni, and J. Wetzel, “Learning science by constructing models: can dragoon increase learning without increasing the time required?,” *Int. J. Artif. Intell. Educ.*, vol. 26, no. 4, pp. 1033–1068, 2016.
[19] V. Karavirta and C. A. Shaffer, “JSAV: The JavaScript Algorithm Visualization Library,” in *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 2013, pp. 159–164.
[20] “Khan/khan-exercises: A (deprecated) framework for building exercises to work with Khan Academy.” .
[21] W. Barendregt and M. M. Bekker, “Children may expect drag-and-drop instead of pointand-click,” in *Conference on Human Factors in Computing Systems - Proceedings*, 2011, pp. 1297–1302.
[22] K. M. Inkpen, “Drag-and-Drop versus Point-and-Click Mouse Interaction Styles for Children,” *ACM Trans. Comput. Interact.*, vol. 8, no. 1, pp. 1–33, Mar. 2001.
[23] A. Donker and P. Reitsma, “Drag-and-drop errors in young children’s use of the mouse,” *Interact. Comput.*, vol. 19, no. 2, pp. 257–266, Mar. 2007.
[24] “Nerdamer | Symbolic Math for Javascript.” .
[25] “math.js | an extensive math library for JavaScript and Node.js.” .
[26] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, “Cognitive tutors: Lessons learned,” *J. Learn. Sci.*, vol. 4, no. 2, pp. 167–207, 1995.