

# **Developing a Program to Assist in Qualitative Data Analysis: How Engineering Students' Discuss Model Types**

## **Abstract**

This Research paper discusses the opportunities that utilizing a computer program can present in analyzing large amounts of qualitative data collected through a survey tool. When working with longitudinal qualitative data, there are many challenges that researchers face. The coding scheme may evolve over time requiring re-coding of early data. There may be long periods of time between data analysis. Typically, multiple researchers will participate in the coding, but this may introduce bias or inconsistencies. Ideally the same researchers would be analyzing the data, but often there is some turnover in the team, particularly when students assist with the coding. Computer programs can enable automated or semi-automated coding helping to reduce errors and inconsistencies in the coded data.

In this study, a modeling survey was developed to assess student awareness of model types and administered in four first-year engineering courses across the three universities over the span of three years. The data collected from this survey consists of over 4,000 students' open-ended responses to three questions about types of models in science, technology, engineering, and mathematics (STEM) fields. A coding scheme was developed to identify and categorize model types in student responses. Over two years, two undergraduate researchers analyzed a total of 1,829 students' survey responses after ensuring intercoder reliability was greater than 80% for each model category. However, with much data remaining to be coded, the research team developed a MATLAB program to automatically implement the coding scheme and identify the types of models students discussed in their responses.

MATLAB coded results were compared to human-coded results ( $n = 1,829$ ) to assess reliability; results matched between 81%-99% for the different model categories. Furthermore, the reliability of the MATLAB coded results are within the range of the interrater reliability measured between the 2 undergraduate researchers (86-100% for the five model categories).

With good reliability of the program, all 4,358 survey responses were coded; results showing the number and types of models identified by students are presented in the paper.

Keywords: first-year engineering, modeling, qualitative survey data, data analysis, MATLAB tool

## **Introduction**

Many educational researchers agree that qualitative data analysis provides important insights necessary for curriculum and institutional changes. It is especially important to ensure underrepresented students' perspectives are not lost or muted within large data sets. Although qualitative research is critical, it can be extremely time consuming. A common practice in analyzing qualitative data is to develop a coding scheme or framework to analyze data, train research assistants (RAs) to apply the framework, ensure sufficient inter-rater reliability, and then have RAs analyze the data [1-3]. Some researchers also discuss the use of machine learning

or artificial intelligence to help throughout the qualitative data analysis process as another pathway to analyzing data [4-9]. This paper explores the idea of developing and using a computer program to assist in coding open-ended survey responses.

While the use of human coders for qualitative data is relatively commonplace, there are some challenges involved. With large sets of qualitative data, often multiple researchers are used to code quicker. However, this introduces the possibility of one researcher's bias skewing results. Typically, there are inter-coder reliability measures that are put into place to prevent this with an assumed numerical threshold dictating that reliability has been achieved [1-2]. However, a high intercoder reliability rating, such as 70-80%, does not mathematically imply that the coders are performing with both high accuracy and high agreement [1-2]. Additionally, there is an issue with consistency over longer periods of time, where researchers typically re-code data to ensure similar coding trends later into a study [3].

The use of machine learning techniques in qualitative analysis has had mixed results. Verleger (2015) used decision trees to numerically evaluate the quality of students' written work [4]. Using a hand-coded historical data set, he was able to train an algorithm to classify student work, but only achieved an accuracy between 57% and 80% on new data, depending on the rubric item being assessed. This accuracy was only after applying an algorithm built on training data which took a substantial amount of time to generate. He and Schonlau (2021) compared the accuracy of automated coding using 2 types of statistical learning models (support vector machines and random forests) to human coding [10]. They found that the accuracy of the human coders (82-97%) was significantly greater than the automated coding (58-78%) in 3 large datasets (over 1000 responses each). However, they did find correlations between the auto- and human-coded results suggesting that they both found the same responses easy or hard to code. The same authors have also attempted semi-automated coding methods to improve accuracy using machine learning to identify and code easy responses, leaving the more difficult ones to be manually coded [11] or to identify responses in a dataset with a high probability of error for further analysis via double-coding [8].

These examples highlight two challenges with automated coding algorithms: namely the need for training and the accuracy of the results. As discussed, the struggles with utilizing automated computer coding and manual human coding for data analysis are essentially the same. One of the goals of this study was to try to find a balance of time and accuracy, combining both of these methods to exploit their benefits, while minimizing their shortcomings.

Despite challenges in achieving sufficient accuracy, the benefits of computer-assisted coding for analysis of large datasets appear useful given sufficient time to train the program or develop the framework. In this study, the research team collected data consisting of over 4,000 students' responses to a Modeling Survey administered at three universities over four semesters. The longitudinal nature of the data collected increases the potential errors in data analysis by researchers, as Elliott (2018) discussed [3]. Thus, we wanted to see if we could get sufficient consistency and accuracy using a computer program to assist in automatic coding of the data using an already established coding scheme. The team developed a coding scheme to categorize identified model types in students' open-ended responses about their ideas of models in STEM fields [12]. Previously, two researchers independently coded portions of the data after over 80%

intercoder reliability was attained. After each survey was administered, there were concerns about consistency with long periods of time in between coding students' responses. Eventually the team used the developed coding scheme and the previously coded data to develop a MATLAB tool to analyze all the data collected throughout this research project.

One of the major goals of this research project was to implement modeling interventions requiring varying levels of change and investigate their impact by assessing students' awareness of model types before and after the intervention. Models are fundamental to engineering, but many engineering students cannot identify more models than prototypes [12-13]. There are well-researched modeling interventions to help students learn about different models, such as model-eliciting activities (MEAs) that help students learn how to develop mathematical models [14]. Some interventions like MEAs require too much effort to implement and assess for others to find them adoptable [15]. There are also sometimes institutional barriers that limit the amount of change that can be implemented in courses that have multiple sections with multiple instructors, which is the typical structure of many first-year engineering courses. This research involved the investigation of modeling interventions in first-year engineering courses that required varying levels of change and buy-in. The details of various interventions implemented are further discussed in other works [12,16].

## **Research Purpose**

The purpose of this paper is to discuss the development and utility of a MATLAB Program to analyze students' written responses to open-ended survey questions about models based on an established modeling framework. The process, as well as the benefits and concerns of implementing similar approaches are discussed and compared to analysis completed by undergraduate research assistants. Preliminary results based on all the multiple years of data collected throughout this study and analyzed with the MATLAB Program are also presented and discussed.

## **Methods**

### ***Setting and Participants***

In Fall 2018, the research team began developing and testing various modeling interventions, as well as methods to collect data regarding the impact of the interventions in first-year engineering courses across three universities. At Embry-Riddle Aeronautical University (ERAU), there are three engineering courses (i.e. design course, MATLAB course, CAD course) that are typically taken by students in their first year. These were substantially revised to incorporate modeling language, concepts, projects, and problems throughout. The modeling interventions implemented in the MATLAB course included an iterative modeling problem similar to MEAs and model-adaptation activities (MAAs) [17,18]. The other pedagogical interventions were further discussed in other publications [12,16]. At San Jose State University (SJSU), the lab portion of the only first-year engineering course consisted of two design projects. One of these projects was recreated to incorporate physical, CAD, mathematical, computational, and financial models throughout [12]. At University of Louisville (UofL), there are two consecutive first-year engineering courses. The first course provides instruction in key engineering skills areas, such as

programming and graphics. The second course requires students to apply those skills in a semester-long team design project, which was not revised. Instructional materials in the first course were reviewed to find where modeling was implicitly included, and modeling language was explicitly added throughout the course to identify physical, graphical, mathematical, computational, and business/financial models where relevant [12].

**Data Collection**

Over the course of the academic year (Fall 2018 - Spring 2019) and Summer 2019, the team used their findings, anecdotal evidence, and reviews of relevant literature to develop a survey about models [12,19]. The survey was also revised based on a review by an external group of four modeling education experts. The purpose of the survey was to investigate students’ awareness of different types of models and how to apply different models to solve engineering problems.

The complete modeling survey was published by Rodgers et al. [12]. The previous study and this study focus on the first three survey questions about modeling (Q3-Q5) which are:

- Q3. What is a model in science, technology, engineering, and mathematics (STEM) fields?
- Q4. List different types of models that you can think of.
- Q5. Describe each different type of model you listed.

In Fall 2019, the modeling survey was first administered to first-year engineering students at two universities. In Fall 2020, it was administered to students at the original two universities along with a third. The surveys were implemented in the courses as a class assignment at the beginning (pre) and end (post) of the semester. The number of students that completed each semester in the four courses are shown in Table 1. There were a total of 4,358 survey responses collected across the two years.

Table 1. Summary of Students that Completed Modeling Survey

	Fall 2019	Spring 2020	Fall 2020	Spring 2021
Embry-Riddle Aeronautical University, FYE CAD Course	Enrolled: 437 Pre: 247 Post: 165	Enrolled: 309 Pre: 188 Post: 66	Enrolled: 377 Pre: 211 Post: 99	Enrolled: 357 Pre: 220 Post: 60
Embry-Riddle Aeronautical University, FYE Programming Course	Enrolled: 375 Pre: 364 Post: 201	Enrolled: 431 Pre: 411 Post: 238	Enrolled: 432 Pre: 393 Post: 287	Enrolled: 419 Pre: 271 Post: 288
San Jose State University, FYE Design Course	Enrolled: 25 Pre: 23 Post: 22	N/A	Enrolled: 25 Pre: 22 Post: 22	N/A
University of Louisville, FYE Fundamentals Course	N/A	N/A	Enrolled: 570 Pre: 311 Post: 249	N/A

Students’ responses to the first three questions about modeling in the survey (Q3-Q5) were previously analyzed by undergraduate research assistants [12]. The Fall 2019 and Spring 2020

data that was analyzed for ERAU (n = 1,829 students' responses; shown highlighted in Table 1) was used in this study to test the reliability and validity of the developed MATLAB Program. Some responses were excluded due to incomplete or duplicate responses.

### ***Coding Scheme***

Rodgers et al. (2021) developed a coding scheme to identify types of models that students discussed in their responses to open-ended survey questions [12]. To ensure validity of the coding scheme, relevant modeling literature was reviewed to ensure the types of models in engineering were accurately reflected. Throughout the development of the coding scheme, the team used inductive and deductive approaches to ensure both the literature and the data were reflected.

Once intercoder reliability was obtained, the two undergraduate researchers coded a total of 1,829 survey responses. This consisted of data collected at University 1 in two different first-year engineering courses across two semesters. Throughout the development of this framework and analysis of students' responses, the researchers documented patterns and developed a list of keywords. Based on the analysis of students' responses, it was found that students presented different types of models in their responses by labeling them, explaining them, and/or providing examples (see Figure 1).

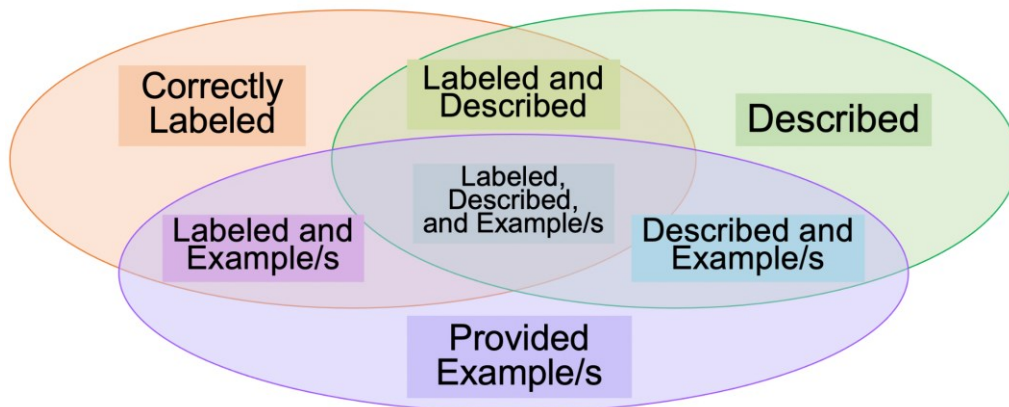


Figure 1. How Models were presented in Students' Responses

These ideas and the list of keywords were then used to develop a MATLAB Program that could analyze all the data to identify the established five types of models (i.e. Physical, Graphical/Virtual, Mathematical, Computational, and Financial/Business). The code was written to search for keywords, combinations of keywords, and established patterns to code types of models that students identified in their responses.

### ***MATLAB Program Development***

The developed code started with cleaning the students' responses to minimize concepts missed because of formatting or importing data. All data that was not a character or number was removed (e.g., spaces, apostrophes, hyphens). The students' responses to all three questions were combined to be analyzed as one data point.

Once the data was ready for analysis, each student's response was analyzed to code the types of models identified. The general flow of the code is shown in Figure 2.



Figure 2. Order of Models Analyzed in Students' Responses in MATLAB Program

The order was important to enable certain codes to be based on both keywords and if a certain type of model was already coded. For example, if a student's response included the word "digital" it was coded as a Graphical/Virtual Model as long as it was not already coded as a Computational or Mathematical Model. In addition to this, a student's response was coded as Computational Model, when it included "computerprogram" and was not already coded as discussing a Graphical/Virtual Model. This required Computational Models to be analyzed both before and after Graphical/Virtual Models (refer to the orange symbols in Figure 2).

#### *Physical Models*

Students' responses that contained the following words were coded to identify Physical Models: physical, prototype, build, tactile, and tangible. Students' responses that also contained "scal" were coded for Physical Models as well, as long as they were not already coded as Graphical/Virtual Models. This was to capture students' responses that included the words, scale, scaled, and scaling. Students' responses that contained the word print, but not blueprint were also coded as Physical Models. In addition to this, common examples of scaled physical models that students discussed (i.e., model cars, model planes, and model rockets) were added to the list of keywords that led to responses being coded to identify Physical Models.

#### *Graphical/Virtual Models*

Responses where students discussed CAD or engineering drawings were coded to identify Graphical/Virtual Models. These responses were identified by at least one of the following keywords: CAD, "computeraideddesign", "computerdesign", "virtualmod", 2d, drawing, sketch, symmetric, illustrate, blueprint, and diagram. Based on the CAD programs commonly used in the different courses and the three universities, if students wrote CATIA, SolidWorks, SolidEdge, or Inventor, they were coded as Graphical/Virtual Models. In addition to the use of these keywords, students' responses were searched for "3d"; these were only coded if they did not also include the word "print", "plot", or "structure". If they included the words print or structure, it was assumed they were more likely discussing Physical Models when they wrote about 3d. If they included the word plot, it was assumed they were more likely discussing a Computational Model or some type of graphing. Also if the students included the word "digital" in their responses and they were not already coded as Mathematical or Computational Models, then they were coded as Graphical/Virtual Models.

#### *Mathematical Models*

There were a few keywords that easily led to Mathematical Models being coded, including math, equation, formula, and "calculat" for calculating, calculation, calculate/d, etc. If students wrote process or data model, these were also coded as identifying Mathematical Models. There were a

few keywords that were coded as Mathematical Models, if the students' responses were not already coded as Computational Models. These included algorithm and data. In order for data to be coded as a Mathematical Model, the students' responses also could not include graph, plot, or table. Also, if students wrote about a flow chart in their survey response and did not write about graphing representations, it was coded as a Mathematical Model.

### *Computational Models*

To code Computational Models, the concept of simulations and computer program models based on mathematical models were captured through various keywords. The students' responses that stated code, coding, and programming were coded as Computational Models. Based on the context of the courses at the three universities, the coding language programs used in the courses (i.e., MATLAB, Python, and java) were coded as Computational Models. Also based on the context of the courses, if students wrote about Excel or a spreadsheet, they were coded as a Computational Model. If students wrote computer program and they were not already coded as a Graphical/Virtual Model, then they were coded as a Computational Model. This dependence on their code for Graphical/Virtual Model was incorporated because many students discussed computer programs along with CAD programs. Also, students' responses that included "simulat" for simulation/s, simulating, simulate/d, etc. were coded as Computational Models. Students' responses that included both input and output were coded as Computational Models. Along with a core feature of simulations, the keyword "interactive" was also considered, but this alone was not coded as Computational Models because it led to too many false positives in the coding. The last keyword that led to a response being coded as a Computational Model was computational.

### *Financial/Business Models*

In the Fall 2019 and Spring 2020 analyzed data, there were not many examples of Financial/Business Models; this was the least coded type of model. Since there were not as many data points to pull from, it had the least amount of keywords. The keywords "financ" (for finance/s, financial, etc.), business, management, and risk were coded as Financial/Business Models.

### ***Data Analysis - MATLAB Program Reliability***

To assess the reliability of the MATLAB program, program outputs were compared to the researcher-coded data. The researcher-coded data consisted of 1,829 responses (from ERAU). Researcher codes were determined by two undergraduate research assistants that independently coded part of the data set. To ensure consistency throughout, there were weekly meetings while they were coding students' responses where any codes they were unsure about were brought up and discussed amongst the two undergraduate research assistants and one of the primary researchers.

The research team went through a process of adding, removing, and modifying keywords in the MATLAB Program, as well as modifying the logic, to increase the reliability of the tool to accurately code the types of models that students discussed. This process consisted of reviewing students' responses for codes that were missed and adding content based on these missed items. This process also consisted of looking for responses where items were coded incorrectly. A report of the correctly identified, missed, and false positive codes for each of the types of models

was utilized throughout the refinement process in developing the MATLAB Program to understand which model categories needed improvement. The accuracy of the MATLAB program, compared to human-coded results, was checked after each major change. If the accuracy increased, the change was kept. If the accuracy decreased, the change was removed or revised until the change improved accuracy. This process continued until all the identified keywords were exhausted and there were no more positive changes.

To further check the reliability of the MATLAB Program in identifying types of models in students' survey responses, the accuracy was compared to that of multiple human coders for a subset of the data. Four sets of 30 students' randomly selected survey responses were used to train the researchers (human coders) in our previous research. Acceptable percentage agreement between the 2 researchers was attained in the final (fourth) round of coding. This data and their agreed upon analysis was used to compare the percentage agreement between the research team's codes and the MATLAB Program's resulting codes (see Table 3).

***Data Analysis - MATLAB Program Application***

Once the MATLAB program was deemed suitable, it was applied to the full dataset of student responses from all three universities (n = 4,358 Table 1) to determine the number and types of models students identified. Results were compared at the beginning and end of the semester for each of the four courses.

**Results**

***MATLAB Program Reliability***

After many rounds of modifications, the MATLAB Program reached over 80% percent agreement for each of the five types of models when comparing the MATLAB Program's codes to the researcher team's codes for the final round of intercoder reliability analysis (n = 30 students' survey responses) [12]. The team came to a consensus on all the codes then used this to compare to the MATLAB results. The second row shows the percent of agreement between the research team and the MATLAB Program. The percent of agreement for each category was considered acceptable since all were over 80% [1].

Table 2. Intercoder Reliability Measurement – Percentage Agreement for N=30 Survey Responses

	<b>Physical</b>	<b>Virtual/ Graphical</b>	<b>Mathematical</b>	<b>Computational</b>	<b>Financial</b>
Comparing two Researchers	93.3%	86.7%	96.7%	93.3%	100.0%
Research Team vs. Program	86.7%	90.0%	90.0%	90.0%	100.0%

The MATLAB Program also reached over 80% percent agreement for each of the five types of models when comparing the MATLAB Program's codes to the student researchers' codes across the 1,829 Survey Responses (see Table 3). Overall the MATLAB Program and Researcher coded



the same for four or all five model types for 86% of the responses. The number of responses coded by the researchers and MATLAB Program for each type of model are also presented in the last two rows of Table 3. The number of coded responses for each model ranges from 56 to 1,128 coded responses.

Table 3. Reliability of MATLAB Code Compared to Researchers in Identifying Model Types for N=1,829 Survey Responses

	Physical	Virtual/ Graphical	Mathematical	Computational	Business/ Financial
Percent Agreement between MATLAB code and Researcher code	84.3%	81.0%	84.1%	87.2%	98.9%
Researcher-coded responses containing model type (n)	982	1,075	664	512	65
MATLAB-coded responses containing model type (n)	1,128	1,026	776	448	56

Figure 3 shows the number of times each type of model was coded accurately (student researchers also identified the same model) or inaccurately (student researchers did not identify the model, but the MATLAB Program did) and was not coded, but should have been (student researchers identified, but MATLAB Program did not). Physical Models and Mathematical Models were falsely coded much more than missed. The other three types of models (Virtual/Graphical, Computational, Business/Financial) were missed more frequently than falsely coded in comparison to the researchers' codes.

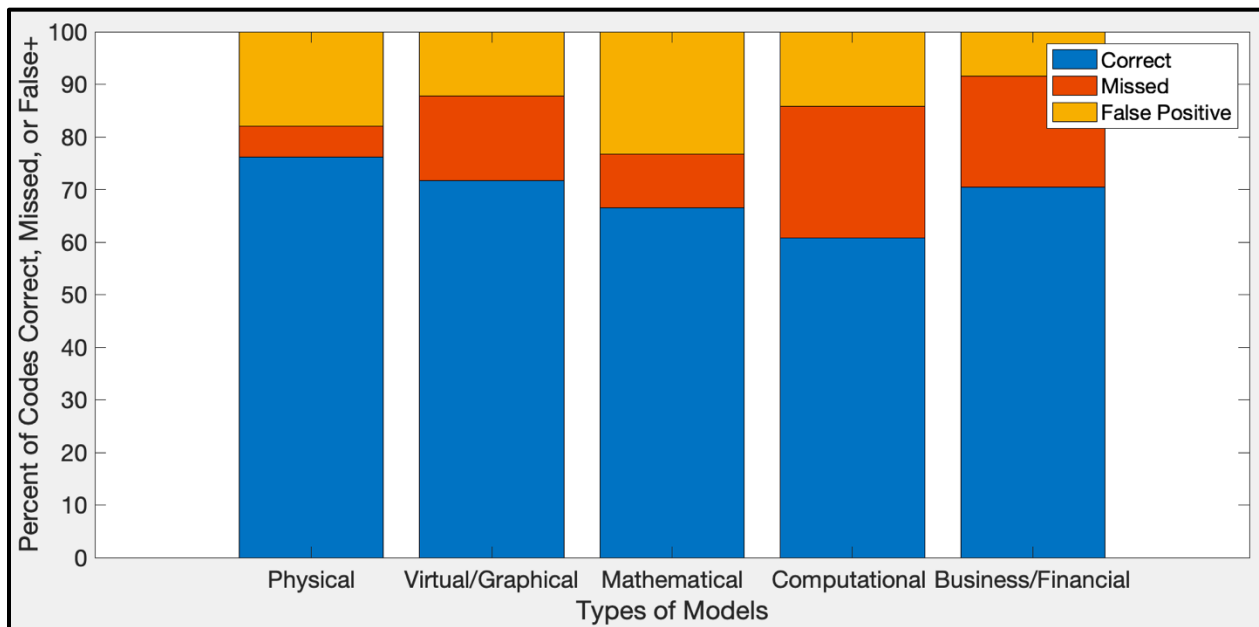


Figure 3. Comparison of Types of Models Coded (MATLAB Program vs. Researchers)

*Sample Student Survey Responses and Codes*

Out of all 1,829 Students’ Survey Responses, there were seven responses that were coded as all five model types by both the MATLAB Program and Researchers. Upon further analysis of the responses, the team agreed that four of them clearly demonstrated understanding of all five model types, but three of them do not (see Table 4). The first two students’ responses listed in Table 4 are examples where students clearly state “Graphical” models, but based on their description, they are not discussing 3D models; ideally this should have not been coded as Graphical Models. The last student response in Table 4, clearly states “Computational Model”, but based on their description they are discussing CAD models – Graphical Models. This should not be coded under Computational Models even though they said the correct word because they do not demonstrate awareness of Computational Models. These three students’ responses are examples where human interpretation would ideally code based on the full picture, but the MATLAB Program will always miss these.

Table 4. Sample Student Responses – Researcher and MATLAB Program coded all model types

	<b>Q3 – What?</b>	<b>Q4 – List</b>	<b>Q5 – Describe</b>
1	It’s used to represent something. It could be used for testing, comparison, evaluation.	financial, computational, physical, graphical	financial - shows the cost of the thing computational - shows the possible outcomes of the thing physical - scale model of the thing graphical - math/numbers of the thing
2	A model is something that is a representation of data.	Physical model, computational model, mathematical model, financial model, graphical	Physical model = 3d representation, computational based on mathematical formulas. mathematical, models based on formulas. financial models model financial situations.
3	A model is a representation of an idea or object that can help an observer understand a concept or information better.	Physical, mathematical, computational, graphical, financial	Physical is actual objects in the real world you can see, such as scale models or prototypes. Mathematical models are used to show theories. Computational models are usually created in a CAD software. Graphical displays information in a format that is easier to understand. Financial shows different ways money and resources can be manipulated.

Out of all 1,829 Students’ Survey Responses, there were no examples where the MATLAB Program and Researchers had opposite codes for all five types of models. There were two examples that the MATLAB Program and Researchers had opposite codes for four types of models (see Table 5). Based on evaluation of the first student’s response in Table 5, the researcher missed codes for mathematical and computational model; the MATLAB Program coded this as a demonstrating a Physical Model, but their description of schematic more clearly points to Graphical Model (like the Researcher correctly coded). Based on the second student’s response in Table 5, the MATLAB Program again miscoded Physical Model and missed Graphical Model, and the Researcher again missed Mathematical Model.

Table 5. Sample Student Responses – Only one matching code (shown by *AGREE*)

	Q3 – What?	Q4 – List	Q5 – Describe	Codes (correct bolded)
1	A computational problem with two parts - calculations and physical data	mathematical, system(s), and schematic	Mathematical can be broken up into empirical, optimization, and structural used to show correlations. systemic models are used mainly in business applications, and schematic models are used to show structure and early stages of planning before decisions on final design are made.	MATLAB Program: Physical, <b>Mathematical</b> , <b>Computational</b> , <b>Financial</b>  Researcher: <b>Graphical</b> , <b>Financial</b>  <i>AGREE: Financial</i>
2	models are portrayals of thoughts, objects and frameworks, which are focal in science, innovation, building or potentially arithmetic.	visual mathematical computer models	visual models are like flowcharts and charts used to represent ideas. mathematical models use math concepts to explain situations computer models use computer graphics to portray what the phenomenon should look like	MATLAB Program: Physical, <b>Mathematical</b>  Researcher: Computational, <b>Graphical</b>  <i>AGREE: not Financial</i>

### ***MATLAB Program Application***

Based on analysis utilizing the MATLAB program of all the data collected, there was an overall decrease in the number of responses with no or one model/s identified and increase in the number of responses with three or more models identified (see Figure 4).

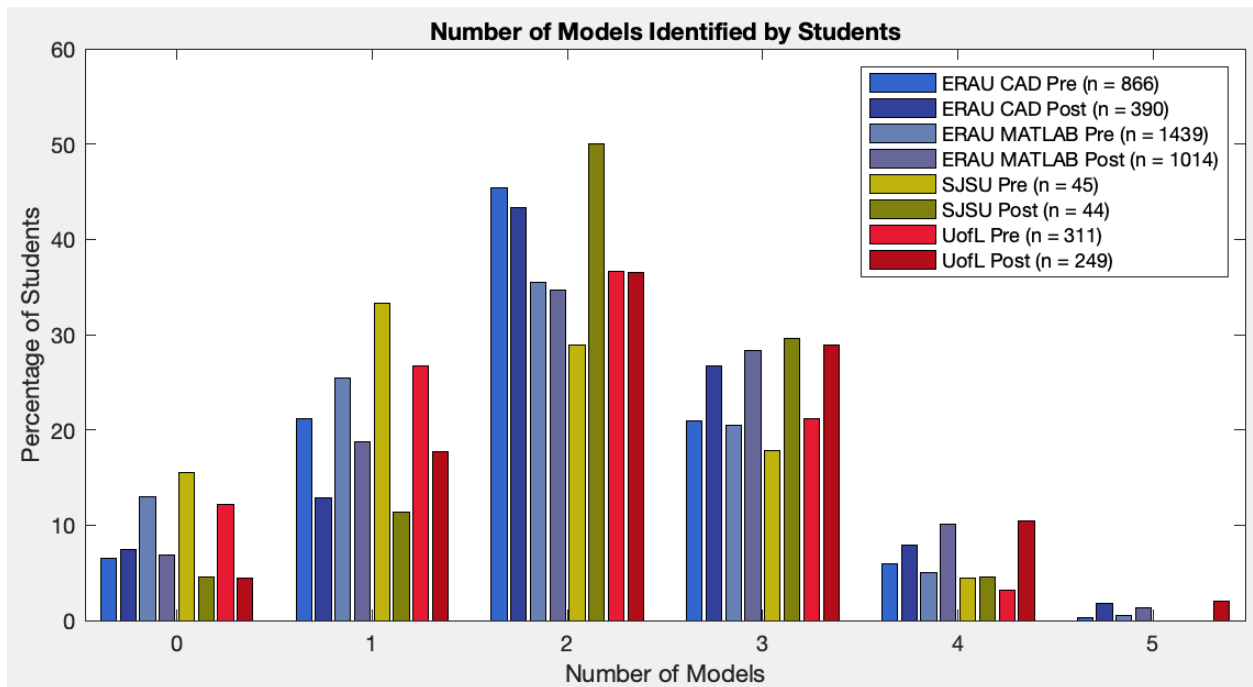


Figure 4. Number of Models Identified in Students' Responses (percentages)

Based on a more detailed analysis, the most commonly identified models in the pre-survey were physical models (Figure 5). This was still the most frequently identified model on the post-survey for three of the courses - all except the MATLAB course at University 1. In the MATLAB course, mathematical models were identified by more students than any other model on the post-survey. Across all four courses, mathematical models were identified by more students in the post-survey than the pre-survey (with increases ranging from 16.3% to 29.3%). Across three of the four courses computational and financial/business models were identified by more students in the post-survey. There was only a large increase in students identifying physical and graphical/virtual models in the Design course at University 2. Out of the model types, financial/business models were identified the least by students (ranging from 0.0% to 7.0% of students).

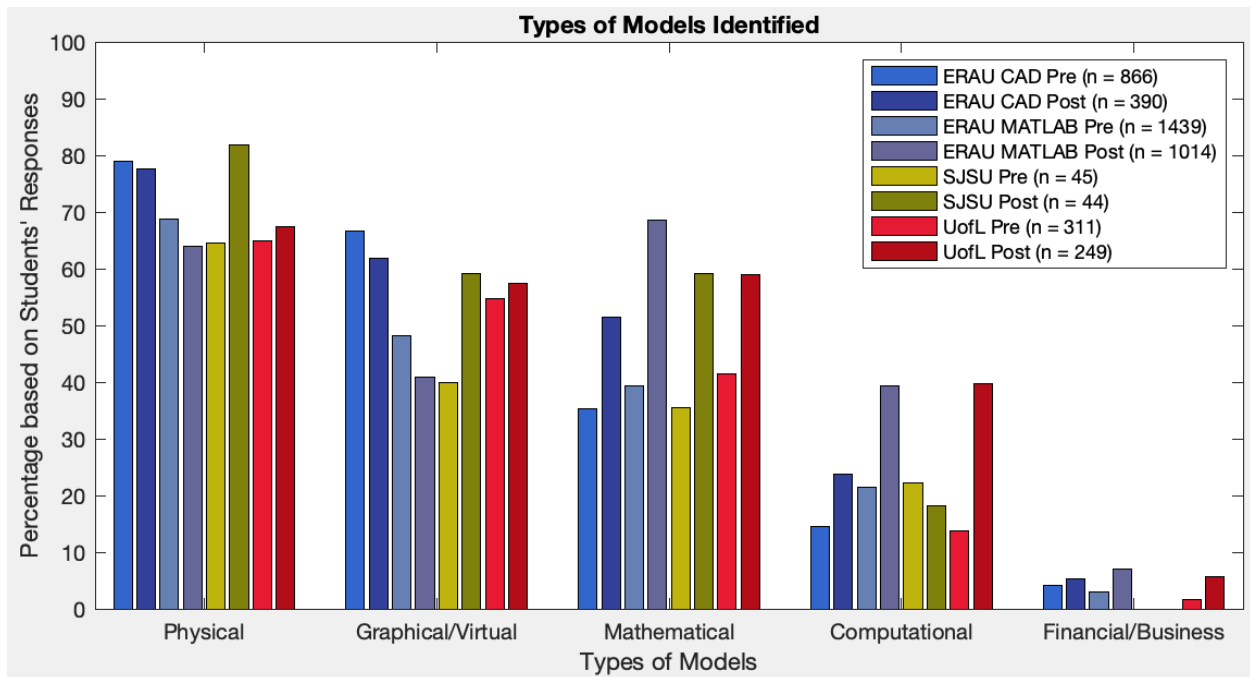


Figure 5. Percentage of students in each course who identified each type of model.

## Discussion

In this study, a computer program was developed to enable automated coding of open-ended survey responses. The program utilized a previously developed coding scheme to identify five types of models (physical, graphical/virtual, mathematical, computational, financial/business) students identified in their open-ended survey responses. The percent agreement between the MATLAB Program and human-coded responses was high (mean 87% for the five model categories) and within the range of variation of human coders. The reliability of this program was higher than that reported by other studies using automated coding [4,9,10]. He and Schonlau (2021a) reported accuracies ranging from 58-78% (compared to double-coded human data where disagreements were resolved through group discussion) using statistical learning models to code large sets of open-ended response data [10]. Giorgetti and Sebastiani (2003) reported 58-63% accuracy of a fully-automated machine learning approach compared to expert human coders [9]. Verleger (2015) attained a mean accuracy of 68%. These studies all used machine learning

algorithms where the model was initially trained on a small subset of the data [4]. In contrast, in our study, the coding scheme was initially developed by the researchers and refined through manual coding of a subset of the data. Refinement of the coding scheme involved four rounds of coding 30 student responses until sufficient interrater reliability was attained. Subsequently, through regular meetings of the research team reviewing items where coders were unsure, the scheme was further revised. Then the coding scheme was implemented through the MATLAB program by identifying keywords in student responses, similar to the process used by human coders.

The use of a highly refined coding scheme enabled good reliability in the use of the MATLAB program. This refinement process took a substantial amount of time and should not be discounted. However, given the size and longitudinal nature of the dataset in this study, development and use of the MATLAB Program was well worth the effort for time savings and consistency in data analysis. Implementation of this automated coding program enabled years worth of coding to be done instantaneously, and without the need to train additional coders. For those doing qualitative analysis on smaller datasets or over a short duration, the additional effort in developing a tailored computer algorithm may not be of value.

Machine learning was not employed in the development of the MATLAB Program, but this could be something that is utilized to go beyond keywords. With the explosive growth in AI research in recent years, new techniques for using machine learning in qualitative research are being developed, with a lot of potential to change how qualitative research is developed and used [6]. Machine learning methods are evolving and new techniques may improve accuracy and reduce the need for development of a refined coding scheme. Current efforts at using machine learning for qualitative analysis generally focus on recommending qualitative codes; still relying on researchers to develop a code book and to approve (or reject) the recommended codes [7]. Machine learning techniques have become good enough that the “training” process can occur throughout the natural coding process without any precoded “training data”, with analysis becoming more accurate as researchers assign codes or reject recommendations [5]. This process still requires coders to initiate and iterate through much of the coding process.

### ***Limitations***

Certainly, errors exist in the MATLAB-coded data. The program compares keywords and cannot account for all possible variations in the open-ended text responses. The MATLAB program is not able to interpret the context in which the keywords were written, unlike a human coder which can use the full text response to make a determination. As illustrated in Tables 4 and 5, however, human coders can also misjudge and fail to fully contextualize survey responses, particularly when coding a large amount of data. The lack of contextualization with the MATLAB Program can be both positive and negative. This can be beneficial in eliminating assumptions based on an individual’s own subjective perspective; for instance, “cell diagram” could mean physical or graphical/virtual model depending on one’s own experience with this concept like a human-coder may. It can also be a negative, when a collection of ideas are presented without keywords, but with sufficient context to understand (a) type/s of model/s students are discussing. Additionally, some codes may be missed due to spelling errors or typos (like “comptuer” instead of “computer”). It may be possible to correct spelling errors prior to coding to increase reliability

of the MATLAB program. With large datasets and longitudinal studies, there is often a trade-off between time and consistency, and researchers must strike a balance between sufficient reliability and use of available time and financial resources.

## **Conclusions**

The computer program used for this study is an effective tool to save researchers time when coding high volumes of qualitative survey data while retaining reliability. Comparisons of results from both the computer program and human coders indicates that the computer program is over 80% reliable. Additionally, this automated coding tool helps maintain consistency over a longer analysis time as different researchers work on the project. The computer program is adaptable in structure and the concept could be adjusted for most open-ended survey questions where distinct classifications occur. Based on this process and these results, we suggest a similar process of researchers analyzing existing literature and/or the data to develop the initial coding scheme to ensure validity and then use a small set or a few small sets of data to ensure reliability of the program (e.g., around 30 coded sample responses). Based on these findings, we do not feel a large set of already coded survey responses is necessary for the development of this kind of analysis tool. In fact, the large set of data that is more likely to contain errors/inconsistencies, due to long periods of time or one researcher coding compared to a team ensuring agreement, could potentially be more of a hindrance than help. Researcher fatigue or burnout when analyzing qualitative data may be an important area for future investigation.

The results of the survey itself point towards an overall increase in student model identification from the beginning to end of the semester (pre to post survey). The depth of intervention in each course, alongside more particular course curriculum, appears to play a role in how much of an increase specific models saw. This will be investigated further in future work since this is not the focus on this study.

## **Acknowledgements**

This work was made possible by a collaborative research grant from the National Science Foundation (DUE 1827392; DUE 1827600; DUE 1827406). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

## **References**

1. A. Wilson-Lopez, A. Minichiello, and T. Green. An inquiry into the use of intercoder reliability measures in qualitative research. ASEE Annual Conference & Exposition proceedings, Tampa, FL. 2019.
2. Anne Margueri McAlister, Dennis M. Lee, Katherine M. Ehlert, Rachel Louis Kajfez, Courtney June Faber, and Marian S. Kennedy. "Qualitative Coding: An Approach to Assess Inter-Rater Reliability". *2017 ASEE Annual Conference & Exposition, Columbus, Ohio, 2017, June*. ASEE Conferences, 2017. <https://peer.asee.org/28777>

3. Victoria Elliott. "Thinking about the Coding Process in Qualitative Data Analysis". *The Quantitative Report 2018 Vol 23 Num 11 2850-2861*.
4. M. Verleger, "Pilot study on applying natural language processing techniques to classify student responses to open-ended problems to improve peer review assignments," in *ASEE Annual Conference Proceedings*, Seattle, WA, Jun 14-17, 2015
5. Nan-Chen Chen, Margaret Drouhard, Rafal Kocielnik, Jina Suh, and Cecilia R. Aragon, "Using Machine Learning to Support Qualitative Coding in Social Science: Shifting the Focus to Ambiguity," *ACM Trans. Interact. Intell. Syst.* 8, 2, Article 9 (June 2018), 20 pages.2018.
6. L. Longo, "Empowering Qualitative Research Methods in Education with Artificial Intelligence," In: Costa A., Reis L., Moreira A. (eds) *Computer Supported Qualitative Research. WCQR 2019, Advances in Intelligent Systems and Computing*, vol 1068, Springer, Cham. 2020. [https://doi.org/10.1007/978-3-030-31787-4\\_1](https://doi.org/10.1007/978-3-030-31787-4_1)
7. T. Rietz and A. Maedche, "Towards the Design of an Interactive Machine Learning System for Qualitative Coding," in *Proceedings of the Forty-First International International Conference on Information Systems (ICIS '20)*, India 2020.
8. Z. He and M. Schonlau, "A model-assisted approach for finding coding errors in manual coding of open-ended questions," *Journal of Survey Statistics and Methodology*, vol. 00, pp. 1-12, 2021. doi: 10.1093/jssam/smab022
9. D. Giorgetti and F. Sebastiani, "Automating survey coding by multiclass text categorization techniques," *Journal of the American Society for Information Science and Technology*, vol. 54(14), pp. 1269-1277, 2003.
10. Z. He and M. Schonlau, "Coding text answers to open-ended questions: human coders and statistical learning algorithms make similar mistakes," *methods, data, analyses*, vol. 15(1), pp. 103-120, 2021. DOI: 10.12758/mda.2020.10
11. M. Schonlau and M. Couper, "Semi-automated categorization of open-ended questions," *Survey Research Methods*, vol. 10(2), pp. 143-152, 2016. doi:10.18148/srm/2016.v10i2.6213
12. Rodgers, K. J., Thompson, A., Verleger, M. A., Marbouti, F., Shah, N., & Thaker, P. (2021, July), *Types of Models Identified by First-Year Engineering Students* Paper presented at 2021 ASEE Virtual Annual Conference Content Access, Virtual Conference. <https://peer.asee.org/37943>
13. A. R. Carberry and A. F. McKenna, "Exploring student conceptions of modeling and modeling uses in engineering design," *Journal of Engineering Education*, vol. 103, no. 1, pp. 77-91, 2014.
14. J. S. Zawojewski, H. A. Diefes-Dux, and K. J. Bowman, *Models and modeling in engineering education: Designing experiences for all students*. Sense Publishers, 2008.
15. K. J. Rodgers, "Development of first-year engineering teams' mathematical models through linked modeling and simulation projects," Purdue University, 2016.
16. Authors. (2022). Manuscript under review. *IEEE Transactions on Education*.
17. Rodgers, K. J., Verleger, M. A., & Marbouti, F. (2020, June), *Comparing Students' Solutions to an Open-ended Problem in an Introductory Programming Course with and without Explicit Modeling Interventions* Paper presented at 2020 ASEE Virtual Annual Conference Content Access, Virtual On line . 10.18260/1-2--34311

18. Rodgers, K. J., McNeil, J. C., Verleger, M. A., & Marbouti, F. (2019, June), Impact of a Modeling Intervention in an Introductory Programming Course Paper presented at 2019 ASEE Annual Conference & Exposition , Tampa, Florida. 10.18260/1-2--32918
19. Marbouti, F., Rodgers, K. J., & Verleger, M. A. (2020, June), Change in Student Understanding of Modeling During First-year Engineering Courses Paper presented at 2020 ASEE Virtual Annual Conference Content Access, Virtual On line . 10.18260/1-2--34269