121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014

Paper ID #9623

Effective Educational Methods for Teaching Assistants in a First-Year Engineering MATLAB® Course

joshua jude heeg
Kyle Flenar, University of Cincinnati
Jordan Alexander Ross
Mr. Taylor Okel, University of Cincinnati

Taylor Okel is a Computer Engineering major at the University of Cincinnati. A sophomore at the time of this paper, he had already worked on another research paper, while maintaining a high grade point average and managing to stay active in the community through service projects.

Mr. Tejas Abhijit Deshpande, University of Cincinnati

Sophomore at the University of Cincinnati

Dr. Gregory Warren Bucks, University of Cincinnati
Dr. Kathleen A. Ossman, University of Cincinnati

Dr. Kathleen Ossman is an Associate Professor in the School of Engineering Education in the College of Engineering and Applied Science at the University of Cincinnati. She teaches courses to freshmen engineering students that require the application of mathematics and physics to solving applied problems from a variety of engineering disciplines and utilize MATLAB for solving computationally intensive problems and analyzing data. She earned a BSEE and MSEE from Georgia Tech in 1982 and a Ph.D. from the University of Florida in 1986. She is a member of IEEE and ASEE.

Effective Educational Methods for Teaching Assistants in a First-Year Engineering MATLAB® Course

Abstract

At the University of Cincinnati, two courses were introduced during the 2012-2013 school year to provide first-year students with hands-on experience in engineering and form a link between engineering and the required mathematics and science courses**.** These interdisciplinary courses form a two-semester sequence in which students apply fundamental theory from algebra, trigonometry, calculus and physics to relevant engineering applications. MATLAB is introduced as a programming tool to enable students to explore engineering concepts, investigate solutions to complex problems, analyze and present data effectively, and to develop an appreciation for the power and limitations of computing tools.

In an effort to improve the learning of students in the course, three sets of two teaching assistants each (TAs) tested a different educational method in the lab portion of the courses. These methods were identified by the TAs after teaching students for several weeks and analyzing the common pitfalls encountered by the students. In the first method, the TAs focus on complimenting students on their performance and encouraging them to break up the problem into smaller, more manageable steps. This fosters an engineering mindset that hopes to assist the students intimidated by programming or those that struggle with starting the problem. The next method tested involved the TAs asking questions about the coding concepts the students were using to encourage a deeper level of thought and understanding with the assignment. Lastly, the TAs questioned the students on the commands used in the lab activity and provided some initial guidance in starting the code for the assignment.

The effectiveness of the different methods was assessed in two ways. In the first, qualitative data, in the form of observations and informal interviews, were collected by the TAs as they implemented the different methodologies. In the second, quantitative data, in the form of student grades on exams and lab assignments are compared between the experimental groups and a control group. The quantitative data was analyzed to support the results of the qualitative analyses.

Background and Motivation

At the University of Cincinnati, a series of courses called Engineering Models I and II were introduced to help students make the connection between the math and science courses typically taken during the first year to engineering applications. This is done by introducing students to computing, through MATLAB, to allow the students to work on a variety of complex problems. For the purposes of this paper, computing is defined as a combination of computer programming and problem solving.

However, computing, and particularly computer programming, is widely viewed as a difficult topic for many students to grasp.[1,2] As stated by McCracken et al., the average student performance witnessed in their study was even lower than they expected.[1] This issue is exacerbated within the context of the Engineering Models courses due to the additional subject matter intended to provide context within engineering, mathematics, and the sciences included in the course. In addition, the Engineering Models I and II courses are predominately taken by first-year students so they are also dealing with the sudden change in school environment. Therefore, methods need to be developed to help students develop an understanding of computing concepts to allow them to focus on the mathematics, science, and engineering content.

The Engineering Models courses consist of a 55 minute lecture and a 120 minute lab each week. The lab activities are designed so that students are given a description of a real-world problem and asked to develop a MATLAB script to solve it. During lab, groups of 10 students sit at a table with one teaching assistant who has taken the class before. The TA stays with this group of students for the whole semester. Many students run into errors while attempting to complete the lab activities and the TAs are the main support for students during lab. Since TAs are integral to helping students understand the material and relate it to practical engineering applications, a group of six TAs decided to explore different teaching techniques to enhance their effectiveness during lab sessions and improve the understanding and performance of their students. Supported by the faculty involved in the delivery of the course, the TAs researched and implemented specific teaching styles with their students and observed the results on the students' performance. The goal of this study is to gain insight into effective methods of presenting new material to students and to mentoring them.

At the beginning of the semester, the TAs associated with the experiment brainstormed a variety of methods to apply to their assigned groups. To do so, the group enumerated the issues beginning programmers experience that a specific method could possibly address. After some research into previous studies, the main issues were decided to be:

- Poor programming practices[1]
- Lack of confidence,[1] TA-student relationships
- Lack of interest[3]

Within McCracken et al's experiment, a group of students were taught introductory programming techniques, and tested using standardized testing. They expressed concern about "bad programming practices," as well as mentioning an apparent "mathematical anxiety" in some students. The TAs appreciated these concerns and adopted them into the study. First, they differed in type; that is, one dealt with addressing "tangible" issues in the student's logic, the other dealt with improving a student's confidence in a more subtle, "intangible" way. Secondly, of the other concerns discussed by McCracken, these were possible to test given the class structure. For example, the language taught *must* be MATLAB, and therefore could not be changed to Java or similar.

The remaining issues decided upon by the TAs address common concerns seen in classroom settings. The obvious issue in almost every classroom is the students who cannot maintain interest in the subject. At this university, all engineers must take Engineering Models I and II. This means the course is comprised of students from all disciplines including those that are non-programming based; therefore, the common mindset among this group is "I do not care; this does not pertain to my major." If a student loses interest, they are less likely to pay attention in lectures, complete assignments thoroughly and on time, etc.[3] Additionally, it was also discussed that students who were either shy or soft-spoken, perhaps even "too cool" to talk to professors and TAs, do not get a chance to voice their difficulties. The idea was brought up to work on TA-student relationships and try to form friendships to possibly discover these problems. As this was similar to the confidence bolstering issue in the research study, the two issues were combined.

After the issues were compiled, methods were developed to address the issues. These methods, discussed in detail later, are summarized as:

- Freely give compliments to students. Work on camaraderie and be especially careful of harsh criticism.
- Quiz the students about random, pertinent material seen in class, with emphasis given to the troublesome aspects.
- Have energy when teaching or tutoring; give examples of how MATLAB or coding in general can be extremely useful outside of the classroom.

The first method of tutoring is aimed at helping students who struggle with programming by fostering the students' confidence with programing and to teach them how to break coding assignments into easy, manageable portions. The TA gives compliments to students, and, when asked very general questions (i.e. "How do I write a program to simulate a Taylors series?" - a whole assignment), the TA will respond with a question prompting the student to focus on smaller portions of the lab (i.e. "How would you use a for loop to perform a simple series?"). This has been shown to be an effective tutoring method.[4]

In the second method, students are subjected to questions regarding the material they learned that week in lecture and lab. These questions serve to supplement the material they are given, and pose challenges to them that may lead to a better understanding of the material. The questions also sever to remind students of the concepts pertinent to the lab activity. This is a variation of just-in-time teaching that is gaining popularity in engineering education.[5]

The final method for this study is to approach teaching the students about the assigned labs at a high level. Essentially, this is an approach to breakdown the problem at hand and to make the students ask themselves what they need to do to make the code solve the assigned problem. A common way of attacking these sorts of programming challenges is to construct flow charts of the logistics of the program. Flow charts offer an opportunity for novice (and even expert) programmers to plot the logical flow of the program to achieve the desired task. This has been shown to help students in solving programming problems.[6] Specifically, the TA will sit down with the students prior to them beginning their programming assignment and talk about the logical flow of the program. The student would then draw a flow chart detailing how they would solve the task.

In the following sections, feedback from the teaching assistants and performance data from student assignments are analyzed to assess the effectiveness of each method. Some suggestions for training TAs who teach in similar classes are also offered.

Data Collection

The data presented in the following sections includes the scores on course assignments and final course grades for several class sections of Engineering Models I offered in the Fall 2013 semester. The students in the experimental groups (one group per method) consisted of those who were directly impacted by the TAs involved in this study. There were 9 students for method 1, 7 students for method 2, and 9 students for method 3. These students came from two sections taught by the same faculty instructor. The control group consisted of 187 students from three sections taught by a different instructor. The TAs of these students received only the basic training provided to all TAs of the Engineering Models course and were unaware of the teaching methods being implemented by the 6 TAs in the experimental group.

The final data presented is the final percentages achieved by the students in the two sections. These grades take into account the lab assignments (20%), homework assignments (20%), midterm exam (20%), final exam (20%), final project (10%), and attendance/quizzes (10%). Lab assignments are scored out of 50 points, and the data presented is normalized to a percentage of the total available points.

The graphs of student performance for each method below on laboratory assignments and in the overall course were creating using a normalized frequency. The students in the experimental groups were normalized against the other students in their method. The students in the control group were normalized against the control group population. The normalized frequencies were computed as follows:

$$Normalized\ Frequency = \frac{Raw\ Frequency}{Total\ Number\ of\ Students}$$

This approach was taken due to the significant difference in the size of the groups. A simple frequency distribution would make it virtually impossible to compare the control and experimental groups. By normalizing the frequency distribution, the scale is adjusted so that the groups can be compared.

In addition to quantitative data, qualitative data was also obtained. Each of the TAs enacting a teaching method recorded observations throughout the semester detailing their thoughts on the method and the affect it had on their students. This data type is important because it addresses factors beyond a sample of student grades, such as attitude and enthusiasm towards course material.

Finally, an end-of-semester, anonymous course survey was administered to students in all 19 sections of the course. The survey asked students to describe their experience with various aspects of the course, and included both Likert scale and open-ended response questions. Of particular interest in this study was a pair of questions addressing the effectiveness of the TAs. Students were asked to state their degree of agreement with the statement “I think that my teaching assistant (TA) for Models I did a good job” on a 5 point Likert scale ranging from Strongly Agree to Strongly Disagree. Students were then given the opportunity through an open-ended response question to explain their response about their TA. These results are discussed in each section to show the concerns expressed by the total student enrollment in the course and serve to verify the specific concerns identified by the TAs with their own students.

Method 1

Method 1 consisted of bolstering student confidence by providing compliments throughout the lab period and encouraging students to break down the problems into smaller, more easily managed pieces. What follows are the observations of one of the TAs who implemented this strategy, performance data from the students on lab and homework assignments and on the midterm and final exams, and results of an end-of-semester survey that contained questions relating to the performance of TAs in the class.

Qualitative Observations:
Observations for week of 10/14/2013:
"With the encouragement, I typically did this anyways as my own way of motivation for the students. This week, I made sure to pay compliments to see everyone's reactions. At one instance, the student was caught off guard and just stared at me. Then I noticed he did have a renewed sense of enthusiasm and had no problem coming to me when he had issues, when typically he waited until he was completely lost to ask me questions. It also was just general reinforcement for the students to show they were going in the proper direction."

Observations for week of 10/21/2013:
"My group started off pretty well today. One of my students I could tell was very hesitant to start today and while the rest of the group was working away on the first problem, he was sitting on Facebook on his phone and just kept looking around. This student tends to struggle a lot with programming and always has issues figuring out where to start. Today, I really focused on breaking down the project for him.

To do this, I explained that the first table was just to use the formulas provided and learn what an iterative process is (based off the lab's flow chart). His main worry was having to code in Mat lab, so by breaking it down and giving him a good starting point to just understand the concept, he seemed much more confident and started right away. When it came to coding while loops, he was completely off as to how to begin. I kept breaking down the problem into its basic components (initial conditions, a conditional statement/loop, and output) and that kind of helped him out. I made sure to compliment him as he figured it out and it seemed to put him more at ease and more relaxed, because he had previously said he felt stupid (and I'm pretty sure he thought I was judging him which obviously wasn't true). I told him that he was doing great and that I was there to help him.

One student said he was terrible with MATLAB at the beginning of the class. He is basically one of the star students at my table. I told him that he knew that wasn't true and that he was typically one of the first people to finish a lab at the table and he knew most of it on his own. He thought about it and mentioned that he guessed I was right and that he did know more than he thought.

I don't have many other examples of specific reactions to the compliments. I made a few to everyone at my table and just tried to maintain a positive atmosphere. I think that's what compliments mostly does - create a positive, friendly atmosphere for learning and increase the student's self-esteem just a bit. "

Observations for week of 11/11/2013:
"I've been continuing to implement the methods of positive reinforcement and encouragement. I think it's actually been fairly effective, though the continued growth of a relationship with the kids could be a factor.

Nothing has really changed much since past weeks, but I've noticed the group dynamic is improved with me. Almost every week a few people tell me they have no idea where to start. I reminded each of them of what they were able to figure out the week before and how they broke the problem up into a smaller section before beginning. Most are daunted at seeing all of the requirements for an entire script. One student, with a language barrier, now always has me explain the purpose of the lab and the concept behind what each main command does for the lab.

I've also noticed that, in general, positive reinforcement and compliments just puts the student at ease and creates an effective atmosphere for working and also instills just a bit more confidence in them with their work."

Quantitative Results:
The graphs below show the performance of the students in the experimental group for method 1 versus the performance of the control group for both the final course average and the performance on the final 10 laboratory assignments. As can be seen from the overall course average, the students in the experimental section did not perform quite as well as the control group. However, on the laboratory assignments, after the TAs implemented the method in the middle of the semester, the performance of the students in the experimental group improved, eventually overtaking the students in the control group. This lends some credence to the qualitative results expressed above, namely that providing encouragement has helped the students to gain confidence and ultimately improve their performance on the portion of the course directly impacted by the TAs.

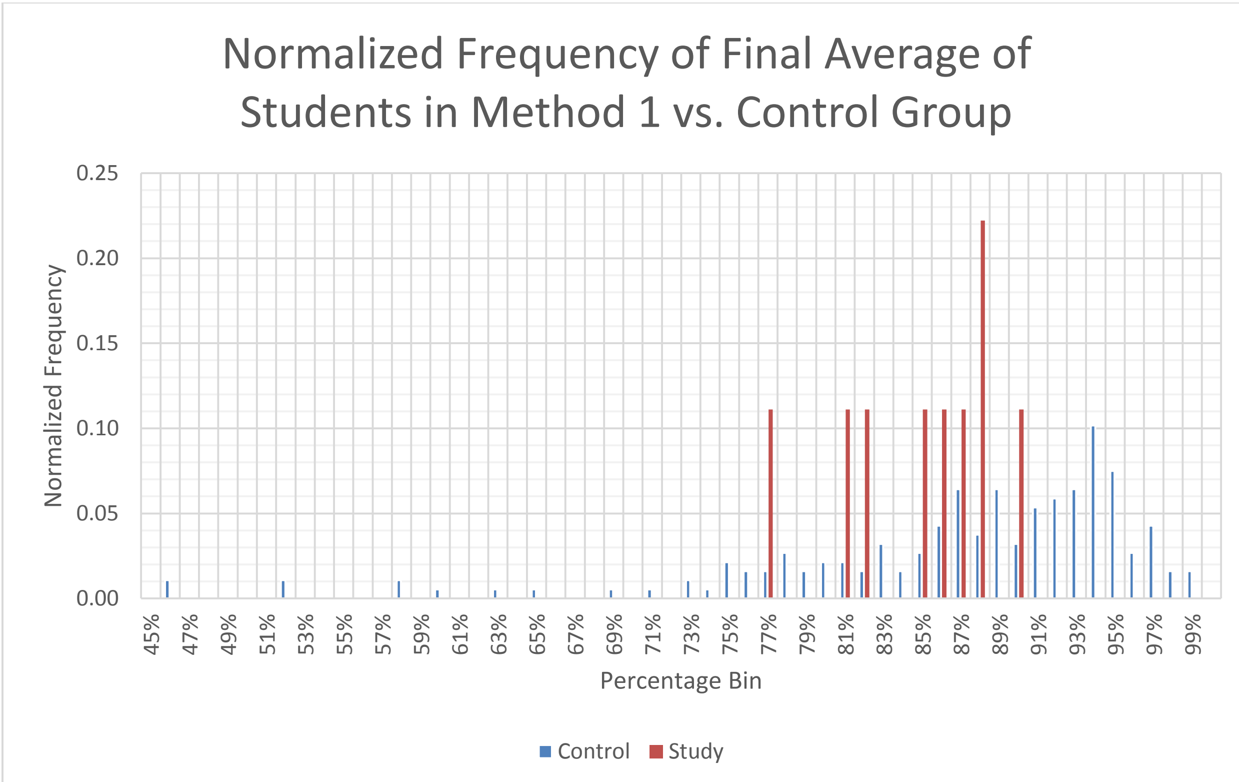

Figure 1. Normalized histogram of final averages for method 1 study and control groups.

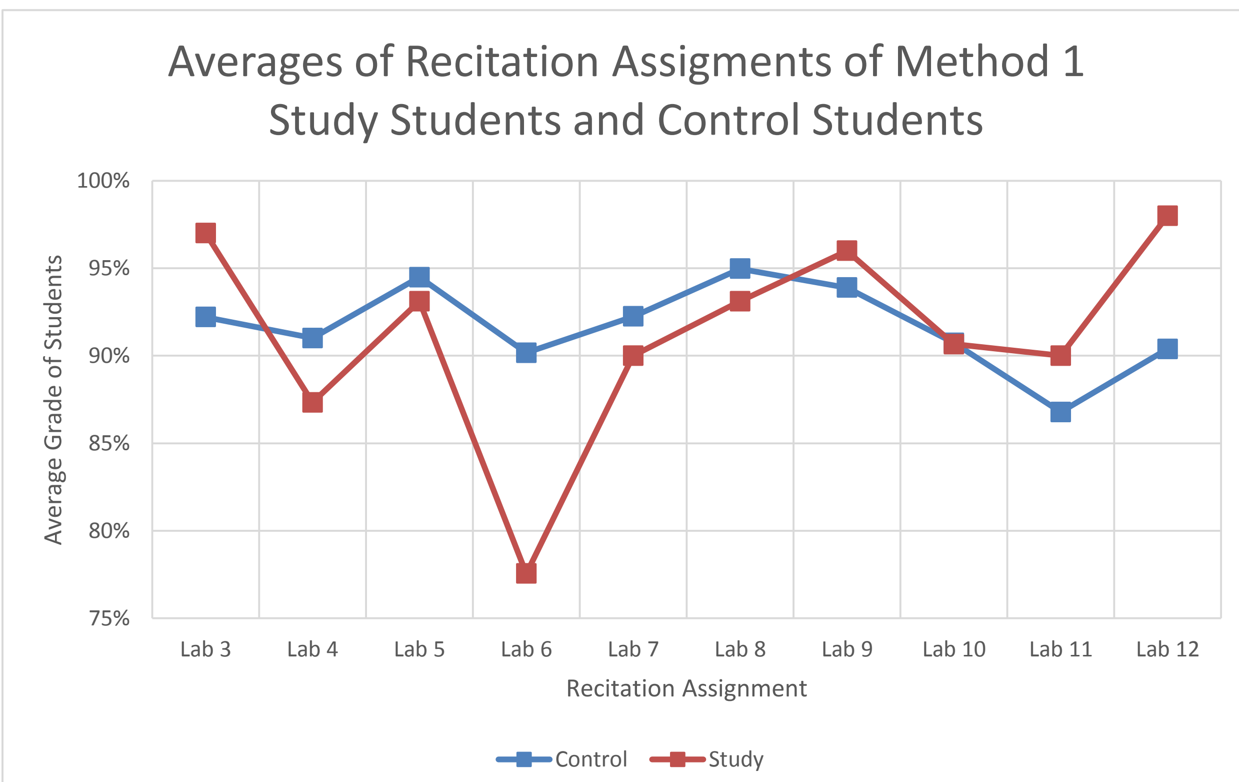

Figure 2. Scatter plot of student averages for each laboratory assignment.

Conclusions:
As mentioned earlier, at the end of the course, students completed a survey, which asked the students about different aspects of the course. One of the topics covered on the survey was the performance of the TAs. Most of the responses indicating that the TA did a poor job said the reason for the poor performance was due to the fact that they were unapproachable. Comparing this response of the overall student population enrolled in Engineering Models I to the qualitative data presented above, the bond between the students and the TA improved with the use of the method and the students were be more inclined to ask questions. Encouraging students to break up labs into smaller portions is also a good practice for solving any type of problem, and helped the students to improve their performance on the laboratory assignments.

Method 2

Method 2 consisted of questioning students before and during lab to check for understanding of the pertinent concepts, to pose challenges to them that may lead to a better understanding of the material, and to remind them to the concepts required for completing the lab activity. The following is a short analysis of this method.

Qualitative Analysis:
In a view of improvements of student performance, the TAs utilizing this method noticed a significant increase in both understanding of the material as well as an ability to apply the knowledge on the part of the students.

While the lectures material in the class served to deliver the material, the questions seemed to aid the students in grasping the material. In other words, a student may vaguely "understand" what was told to them in lecture, however they may not fully grasp what each concept puts forth. As they attempt to answer the question, it is obvious that they begin to fully understand why things are formatted in a specific way and how to use it.

Furthermore, the questions provide new methods of thinking about the material, and new tools which they can use to implement them. This aids in the application of the material, which eases the process of practicing it. Ultimately, the practicing is what will have the largest impact on their grasping of the material, so the ability to complete the lab on their own power is significant. Additionally, getting answers to the questions may also prove to increase confidence, so that they may have ideas about the material, and attempt to solve them through trial and error, as opposed to tentatively coding a few (possibly unsuccessful) lines.

In addition, the questions helped the TAs in identify the issues the group or individual students may be having with the material. This gave them the opportunity to tailor discussions with the student to specifically suit their needs. In other words, the questions both seem to improve the student's overall performance, while simultaneously improving the performance of the TAs.

Quantitative Analysis:
At the conclusion of the semester, grades were compiled and compared against class averages. As with the previous method, the weighted overall course average and the performance on laboratory assignments are compared for evidence of potential improvements.

For the control group, the students' performed at an average score of 87.27%, with the highest score of 96.8% and the lowest falling at 45.7%. The first observation to be drawn from this is the high average. Implications of such an average can mean large numbers of high scores, as well as the low scores (by normal standards) displaying less comparative comprehension. To be exact, 113 students scored at or above average, 74 scored below average. To put in perspective, Figure 3 shows the number of occurrences of each grade. It seems that the average is pulled down by several lower outliers.

Specifically observing the experimental group, the same trend is seen. The experimental average is an improvement at 89.7%. The trend also mimics that of the class, the majority of students fall above average, while lower outliers draw the average down. For the students who managed to score above average, it seems that the score tends to be highly above average, with one score tied for the highest in the class.

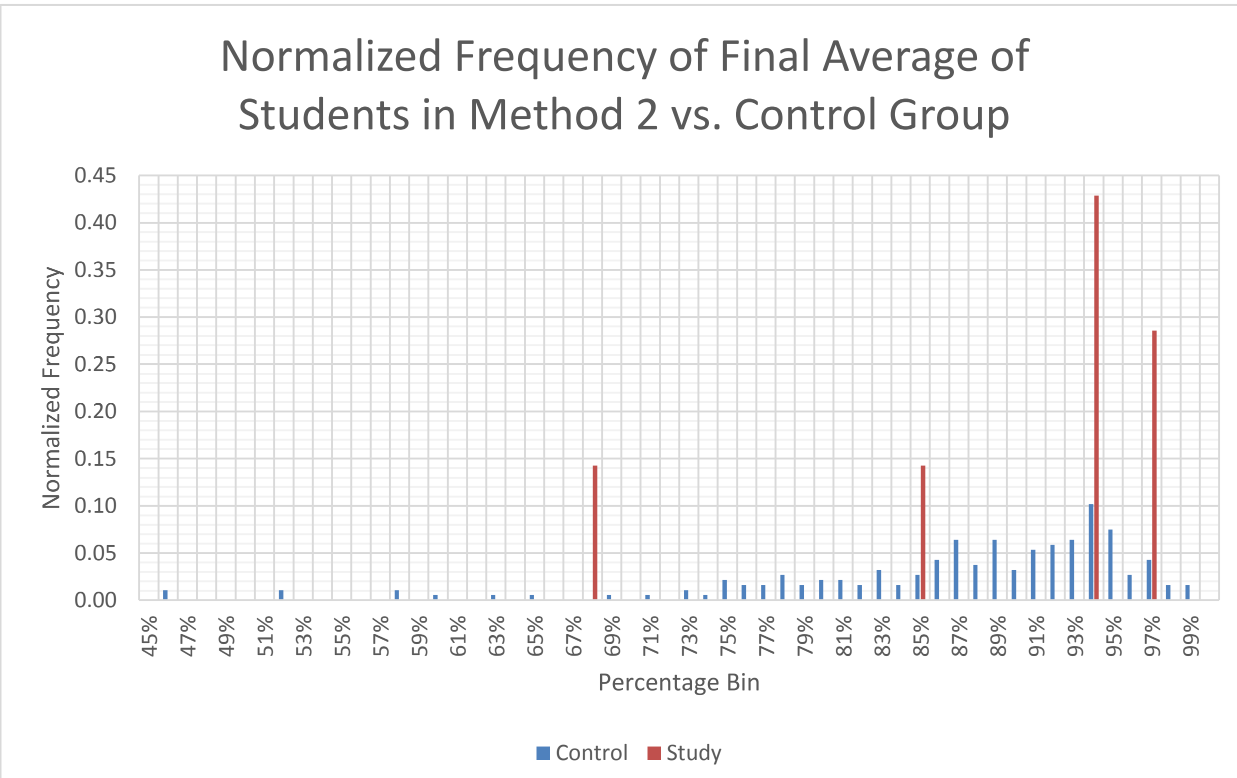

Figure 3. Normalized histogram of final averages for Method 2 study and control groups.

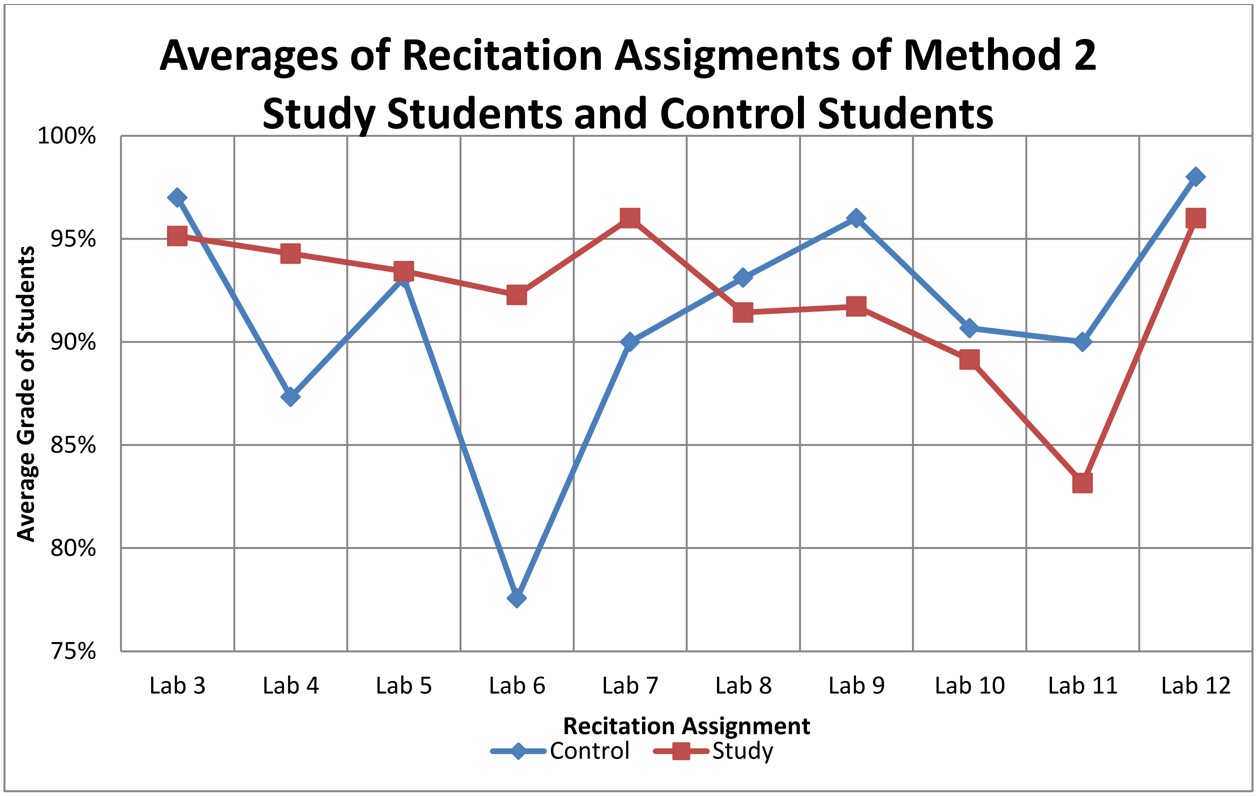

Figure 4. Scatter plot of student averages for each laboratory assignment.

It seems there was a slight improvement on class average. However, it also appears that such improvements are a fallacy. The average was very similar, yet the trends were remarkably mirrored. Overall, the conclusion drawn from the first part of analysis is that there is no change in overall performance, as the experimental average did not far exceed that of the class'; in addition, both averages seem to be dependent on the lowest score.

Conclusions:

In this experiment, students were subjected to questions pertaining to the material studied at the given week of recitation. In doing so, it was hypothesized that the student's scores would improve in respect to the scores of the other students. After collecting results, it appears that the hypothesis was refuted; experimental scores and class scores were approximately even.

However, there still may be benefits to be found in the methodology. As stated in the qualitative analysis of the approach, it was observed that the students seemed to identify and resolve issues in the logic they had with the course's subject matter. Additionally, it provides a method for the teacher to discover issues students may have, and be able to remedy the situation in an efficient manner.

Therefore, while this pedagogy does not seem to produce the desired results, as a teaching method it still seems a viable tool for instructors to evaluate the student's current performances.

Method 3

The use of flowcharts in sequential programming has been a popular tool for many years. Method three was designed to bring in the flowchart concepts in order to aid the students being studied to better understand the code they were writing. The original plan was to go over a high-level flow chart of what the program was supposed to do as a group before they began coding. This worked for many of the students, though as the semester progressed, many of the students would begin working on the assignments, and even possibly complete them, prior to the start of the recitation. The method was adapted for these students to have each student, before they left recitation, sit down with the TA and go through his/her code. In doing so the TA in charge of this study would have the student explain what there code was doing at a high-level, instead of a statement by statement basis. The point of the exercise was for the student to begin to understand the logical flow of what the program was doing.

Results:
After the conclusion of the semester the grades for the nine students were retrieved and were average according to the grading scale laid out by the syllabus. These averages were then compared to a set of 187 averages from a controlled group of students that did not take part in any of the methods discussed in this paper. The results for method 3 can be seen in the following graphs.

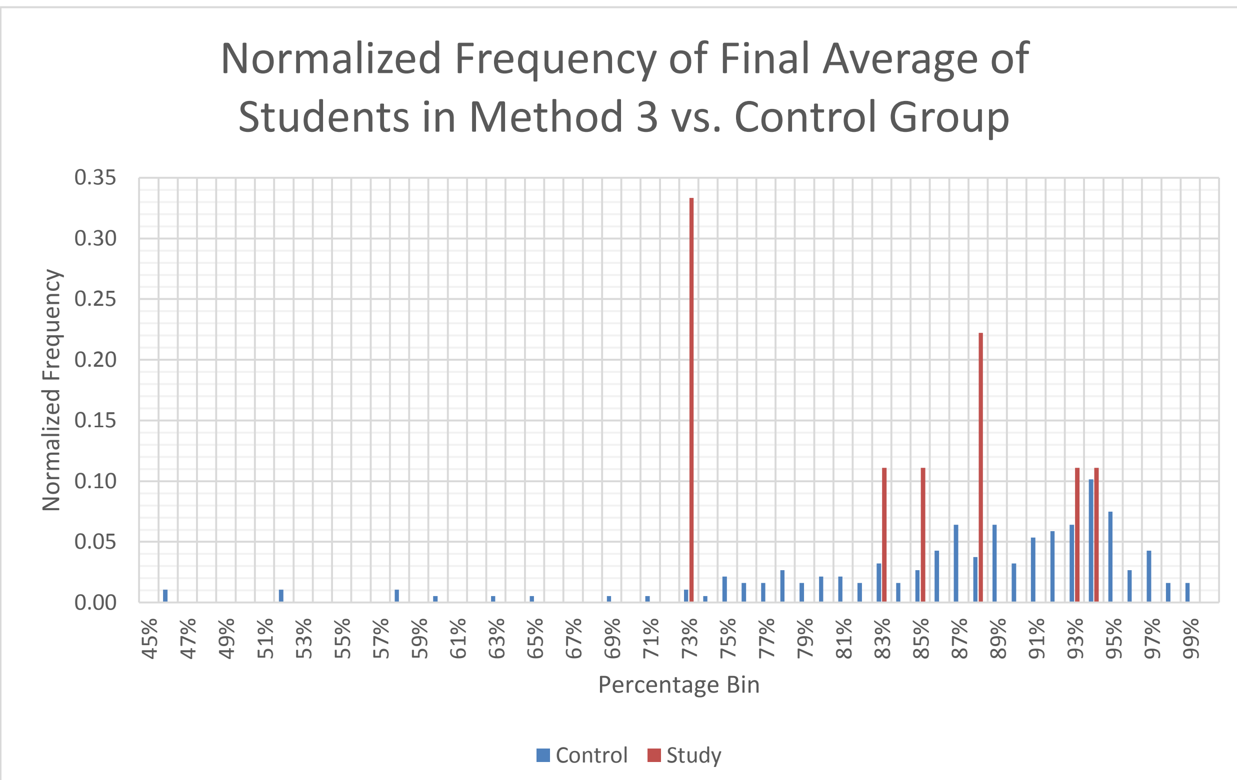

Figure 5. Normalized histogram of final averages for Method 3 study and control groups.

The average of the 187 controlled students was then calculated for their overall average as well as for the 9 students whom were a part of the study. The averages were 87.18% and 83.73%

respectively. These two averages are fairly close to one another, which suggest that no real conclusions of whether the method was successful or not.

While conducting this method on the group of nine students, some qualitative observations were made by the TAs. It was observed that there were three distinct groups of student behavior observed over the duration of the method. The first group (two students in size) consisted of the students who understood programming and improved over the semester by needing diminishing amounts of help. The second group (five students in size) was comprised of students who had no prior understanding of programming but over the course of the semester seemed to need less help understanding the fundamentals. This group however still had questions pertaining to MATLAB specific issues, such as misuse of certain commands and syntax. The last two students had no prior understanding of programming and still did not grasp the high level concepts by the end of the course.

Along with these qualitative measurements it is interesting to note the progression of the laboratory assignment grades as the semester went on. The average grade for the lab assignment was calculated for each assignment for both the 9 students being studied and the 187 students who were not. The graph below shows the results.

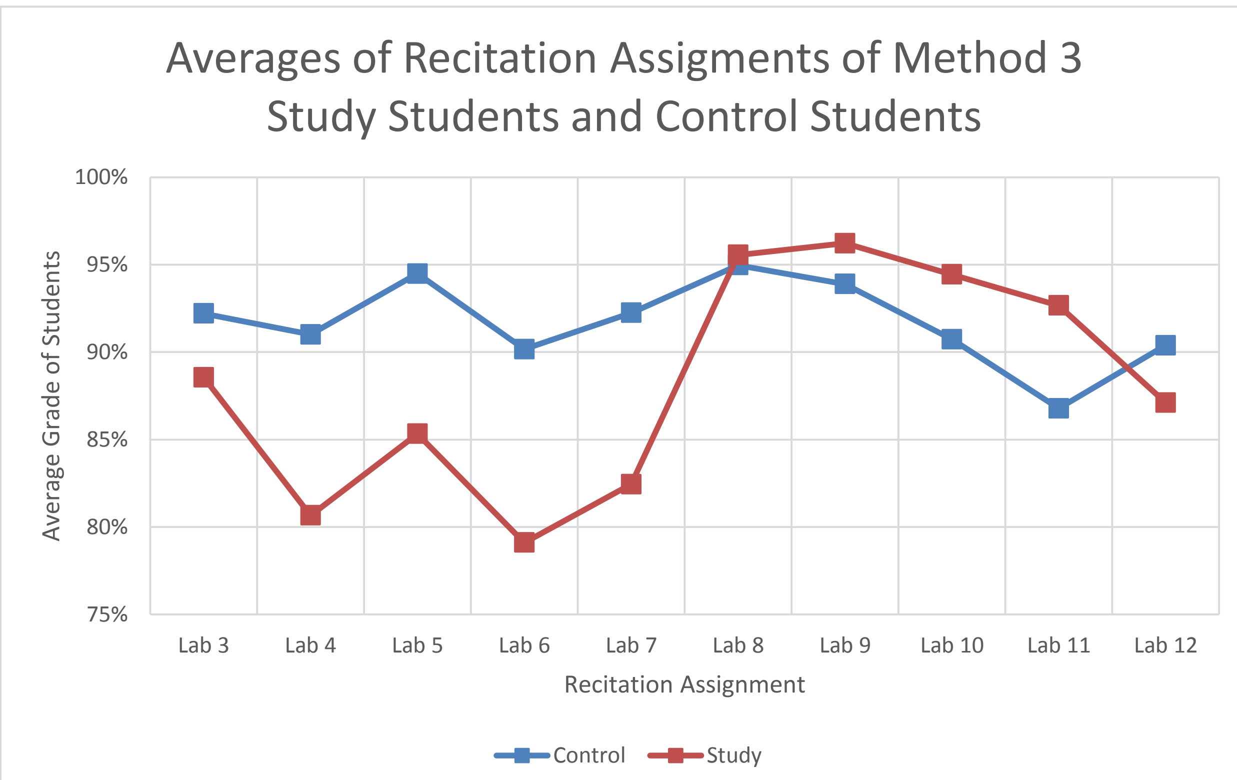

Figure 6. Scatter plot of student averages for each laboratory assignment

Conclusion:
From the above chart is can been seen that as the semester progressed so did the majority of the students who were exposed to this method. This suggests that after a few initial weeks of reviewing the code that the students understood how to solve the assignment correctly.

It was noticed qualitatively that quizzing students on the purpose of their code is a great method for students with an average and above skills at computing. These students will gain a deeper understanding of computing and will perform better. However, students below average would often get more confused when pressed with harder questions and these questions where not a benefit to them.

Discussion

Even though the quantitative data did not show much benefit to the use of these different methods, all of the TAs involved agree that the use of these methods aided the students in learning the material, in accomplishing the lab assignments, and in improving the interactions between the TA and student. As mentioned previously, one of the issues the students identified in the course survey was that many of the TAs seemed distant and it was difficult for the students to talk to them and ask questions. By purposely interacting with the students, either through questioning, guiding, or complimenting, the students felt more comfortable voicing their problems and asking the TA for help, which ultimately, allowed for the TAs to provide better help on the assignments and in clarifying misunderstandings.

One thing that many of the TAs noticed was that it was difficult to stick to the method assigned. What generally happened was that in order to help the students do well, TAs ended up using some of the other methods as well. As an example, although the method assigned to one of the TAs was Method 2, sometimes when the students had a harder time trying to understand an abstract concept or a difficult problem, the TA complimented them on every bit of progress they showed in order to motivate them to understand it better.

Although the analysis above shows the relative effectiveness of separate methods, it would be better to for future TAs to focus not just on one methods but ask them to mix and match these methods based on the situation. Sometimes all a student needs is a bit of confidence in order to complete an assignment whereas other students may need more help in understanding how to structure a problem. The best method for a TA to use is to be aware of multiple techniques and to select the methods that work best for different students and in different situations.

Summary

In order to improve the learning of students, a group of TAs in a first-year engineering course decided to explore different methods of helping students. Three methods were used through the second half of the course, which involved encouraging students through compliments and helping them break problems into smaller pieces, asking questions throughout the course of the lab period to help students refine their understanding, and taking time at the beginning of the lab period to work with the students in outlining the overall structure of the problem they were trying to solve. Overall, these methods did not have a significant impact on the scores of the students on assignments within the course, but from a qualitatively perspective, all of the TAs involved agreed that the methods did help their own interactions with the students and that the students appeared to make improvements throughout the semester. However, it is hypothesized that in order to get the most benefit out of these different methods, they should be used in conjunction with each other.

References

1. McCracken M, Almstrum V, Diaz D, et al. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*. 2001;33(4):125-180.

2. Thomas L, Ratcliffe M, Woodbury J, Jarman E. Learning styles and performance in the introductory programming sequence. *ACM SIGCSE Bulletin*. 2002;34(1):33-37.

3. Carini, R, Kuh, G, Klein, S. Student engagement and student learning: testing the linkages. Research in Higher Education. 2006; 47(1):1-32.

4. Chabay, R. Self-perception and social-perception processes in tutoring: Subtle social control strategies of expert tutors. Self-inference Processes 6. 1990:217.

5. Marrs, K, Blake, R, and Gavrin, A. Use of warm up exercises in just-in-time teaching to determine students prior knowledge and misconceptions in biology, chemistry, and physics. J Coll Sci Teach 33. 2003: 42-47.

6. Scanlan, D. Structured flowcharts outperform pseudocode: an experimental comparison. IEEE Software. 1989;6(5):28-36.