Effect of Flipping the Classroom on Student Performance in First Year Engineering Courses

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.

Dr. Gregory Warren Bucks, University of Cincinnati
Effect of Flipping the Classroom on Student Performance in First-Year Engineering Courses

Abstract

At the University of Cincinnati, three common courses were introduced during the 2012-2013 school year to provide first-year students with hands-on experiences in engineering and a link between engineering and the required mathematics and science courses. Two of these courses, Engineering Models I and II, form a two-semester sequence of interdisciplinary courses in which students apply fundamental theory from algebra, trigonometry, calculus and physics to relevant engineering applications chosen from a variety of disciplines. MATLAB® is introduced and progressively developed as a programming tool to enable students to explore engineering concepts, to investigate solutions to problems too complex for hand solutions, to analyze and present data effectively, and to develop an appreciation of the power and limitations of computer tools.

The Engineering Models sequence was required for all incoming first-year engineering and engineering technology students in 2012-2013. There were multiples sections of these courses with twelve different instructors. Common power-point presentations were provided for lecture. A student survey administered at the end of the sequence revealed that about half of the professors added in-class demonstrations using MATLAB® during lecture while the other half simply read from the power-point slides. Students indicated that it would have been beneficial to spend more time in lecture learning to use MATLAB® to solve problems prior to attending recitation. This year, 2013-2014, a flipped pedagogy is being implemented in the Engineering Models I and II courses. This paper focuses on the effect of the flipped pedagogy on student performance and on student attitudes toward an inverted classroom.

Introduction

One of the challenges in engineering education is teaching problem-solving skills. Many first-year engineering students are comfortable with the concept of exercise solving which only requires them to mimic examples provided by the instructor. However, synthesizing and applying concepts to solve a problem that is dissimilar to problems encountered before is problematic. Many authors have reported on the poor problem-solving ability of students. Heller, Keith, and Anderson1 suggested that many physics students regard problem-solving as independent of physics concepts; they claim to understand the concepts but can’t solve the problems. Many students also regard specific mathematical solutions to be the physics of interest; those students claim to understand the examples in textbooks but can’t solve test problems because they are “too different.” Woods et al2 reported that many engineering students could not solve problems if the wording or context of the problem was changed. They also could not synthesize information from various sources to solve industrial problems.

A promising avenue to explore in order to foster the development of problem-solving skills and to bridge the content areas of mathematics, science, and engineering is through computing.3 We define computing to encompass both traditional programming skills as well as the understanding the strengths of limitations of computing systems and of how to make use of the computer in
solving engineering problems. Given the extent to which computers have permeated the engineering design process, our engineering students must develop strong computing skills in addition to the traditional disciplinary skills. This sentiment has been echoed by many, including the National Academy of Engineering, who identified computing skills as one of the attributes required for future engineers in their Engineer of 2020 report.\textsuperscript{4} Computing affords instructors the ability to introduce “hands-on” projects and activities early in the engineering curriculum while requiring little disciplinary knowledge on the part of the students and no additional materials. Hands-on projects and activities have been shown to increase student motivation and interest in course content and improve retention.\textsuperscript{5-6} Through computing, instructors can bring together concepts and ideas from mathematics, science, and engineering and allow students to interact with them, helping to form the cross-disciplinary mental connections necessary for more expert-like understanding.\textsuperscript{7-8}

In the fall of 2012, the University of Cincinnati converted from a quarter system to a semester system. This conversion provided an ideal opportunity to review the first-year curriculum for the engineering and engineering technology students and make changes to help improve retention and performance of students in the College of Engineering and Applied Science (CEAS). The college faculty agreed on an almost common first year (Table 1), which would include a one year sequence called Engineering Models to try to address the issues of poor problem-solving skills and the lack of connectivity between mathematics and science courses and later engineering courses. The Engineering Models sequence was developed and piloted over the two year period preceding the semester conversion.

<table>
<thead>
<tr>
<th>Fall Semester</th>
<th>Spring Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering Models I</td>
<td>Engineering Models II</td>
</tr>
<tr>
<td>Engineering Foundations</td>
<td>Discipline Specific Engineering Course</td>
</tr>
<tr>
<td>Chemistry I</td>
<td>Physics or Chemistry II</td>
</tr>
<tr>
<td>Pre-Calculus or Higher</td>
<td>Calculus I or Higher</td>
</tr>
</tbody>
</table>

Engineering Models I and II is a two semester sequence of interdisciplinary courses in which students apply fundamental theory from algebra, trigonometry, calculus and physics to relevant engineering applications chosen from a variety of disciplines. MATLAB\textsuperscript{®} is introduced and progressively developed as a programming tool to enable students to explore engineering concepts, to investigate solutions to problems too complex for hand solutions, to analyze and present data effectively, and to develop an appreciation of the power and limitations of computer tools. Students are introduced to such ideas as interpolation, curve-fitting, and numeric differentiation and integration, through applications areas such as data analysis, image processing, communications, position tracking, basic mechanics, and system modeling. Both courses culminate with an end-of-semester team project requiring the students to use MATLAB\textsuperscript{®} to develop a solution to an open-ended design problem.

MATLAB\textsuperscript{®} was selected as the package used in this course for several reasons. First, MATLAB\textsuperscript{®} utilizes a programming language, enabling basic programming concepts to be introduced. In addition, MATLAB\textsuperscript{®} has additional features that makes it much more suited to solving engineering problems when compared with traditional programming languages, as many...
of the graphing and statistical capabilities are already built in and the programming environment is much more forgiving for novices. Students don’t need to deal with the low-level details of declaring variables ahead of time or handling vectors and arrays and instead are able to focus on the problem at hand. Lastly, MATLAB® was chosen at the request of the degree granting programs within the college, as the Engineering Models I and II courses are serves courses to the other disciplines and MATLAB® is used in upper level classes in many of the disciplines.

The objectives of the course are:

1. To explore the application of algebra, trigonometry, and calculus to various engineering disciplines,
2. To learn the fundamentals of programming and good programming practices and utilize these skills to solve numerical problems and create numerical algorithms with MATLAB®,
3. To develop good problem-solving skills by applying problem solving strategies to a variety of engineering problems, and
4. To cultivate effective team-work and communication skills through lab work and design projects.

Inverted Classroom

The Engineering Models sequence was required for all incoming first-year engineering and engineering technology students in 2012-2013. In fall 2012, there were 816 CEAS freshmen and 174 students from outside the college enrolled in Engineering Models I. Sixteen sections of the course were offered with 10 different instructors. Four of the instructors were in the department of engineering education and the remaining instructors were from the other engineering departments. During the first offering of the course, lectures, recitation activities, homework assignments, exams, and projects were common across all sections, though variation existed in how the lectures were delivered. Half of the instructors provided demonstrations using MATLAB® in addition to the PowerPoint presentations while the other half simply read directly from the PowerPoint slides. In the end of semester course surveys, students commented that more time spent on examples of using MATLAB® during lecture would have been both more interesting and helpful in preparing them for recitation activities and homework assignments.

In fall 2013, there was a significant increase in enrollment with 1029 CEAS freshmen and 123 students from outside the college enrolled in Engineering Models I. Twenty sections of the course were offered with 10 different instructors. Four of the instructors were in the department of engineering education (same four from the previous year). The remaining six instructors were from the other engineering departments. Three of the six taught Engineering Models I in fall 2012 while the other three were new to the course in 2013.

In order to address the shortcomings identified by the students from the first offering of the course, a flipped pedagogy was implemented in the Engineering Models I and II courses for the 2013-2014 school year. In a flipped pedagogy,9 traditional lecture content is assigned as homework, freeing the instructor to use the designated lecture time to focus on solving problems and addressing common misconceptions. For the Engineering Models I and II courses, videos were created from the lecture material covered previously. Students were required to watch the
videos prior to attending lecture. In order to ensure that students were watching the videos, a short multiple choice quiz was administered via Blackboard at the start of each lecture. Students brought their laptops to lecture and lecture time was spent using the concepts covered in the videos to solve problems.

The videos were created using the software program, Adobe Captivate. At the end of each video, there was a set of simple exercises for students to try in MATLAB® to reinforce the concepts followed by a multiple choice quiz so students could test their understanding before attending lecture. After completing the quiz, students could review their quiz answers and were also provided with the correct answer for any missed problems. The quizzes given in lecture via Blackboard were not identical to the quizzes in the video but were similar.

Each week, all professors teaching the course were provided with a set of suggested “hands on” activities for lecture. However, all instructors were also free to do alternative activities which covered the concepts if so desired. This added flexibility enabled the instructors to bring some of their own engineering experience to the class room.

As in 2012, all homework assignments, recitation assignments, and exams were common across all sections. The homework and recitation assignments did differ from the assignments given in 2012 in order to ensure that students did not turn in work done by other students last year. The midterm was also changed since midterm exams were returned last year. However, the first seven of the eight problems on the final exam were exactly the same in 2012 and 2013.

**Student Performance**

This paper will only address the effect on student performance in Engineering Models I since Engineering Models II doesn’t begin until early January 2014. However, student performance data from Engineering Models II will be included in the presentation at the conference. Two direct measures were used to determine the effectiveness of the flipped pedagogy on student performance in Engineering Models I: the final exam scores, which were common in both offerings of Engineering Models I, and the percentage of students with final grades of D, F, or W.

Table 2 shows the percentages of CEAS freshmen and non-CEAS freshmen that received a final grade of D, F, or W in fall 2012 and fall 2013. In spite of a significant increase in enrollment between 2012 and 2013, the percentage of students with a final grade below C- dropped from 12.1% to 9.1% for the CEAS freshmen. The non-CEAS freshmen, which consists mainly of Exploratory Studies students (undecided) and Industrial Management students (College of Business) also performed significantly better in fall 2013.
As indicated earlier, the first seven problems (out of eight problems) on the final exam were the same in fall 2012 and fall 2013. Four of the instructors (covering nine sections of the course) recorded scores for each problem on the final exam both in fall 2012 and fall 2013. The grading rubric was identical between 2012 and 2013 and the four instructors were also the same. The results are summarized in Table 3. The 2012 exam data includes 515 students and the 2013 exam data includes 538 students. Students showed significant improvement on three of the problems (P4, P5, and P7), a significant drop on one problem (P1), and no significant change on the remaining three problems (P2, P3, and P6). A description of the topics covered in each of the seven common problems is shown Table 4.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Mean 2012</th>
<th>Std. Deviation 2012</th>
<th>Mean 2013</th>
<th>Std. Deviation 2013</th>
<th>Change In Mean</th>
<th>Z Value</th>
<th>Significant α = 0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>10.95</td>
<td>2.05</td>
<td>10.48</td>
<td>2.36</td>
<td>−0.47</td>
<td>−5.17</td>
<td>YES</td>
</tr>
<tr>
<td>P2</td>
<td>9.81</td>
<td>2.41</td>
<td>9.97</td>
<td>2.34</td>
<td>+0.16</td>
<td>1.573</td>
<td>NO</td>
</tr>
<tr>
<td>P3</td>
<td>8.99</td>
<td>2.62</td>
<td>8.90</td>
<td>2.37</td>
<td>−0.09</td>
<td>−0.818</td>
<td>NO</td>
</tr>
<tr>
<td>P4</td>
<td>9.23</td>
<td>3.35</td>
<td>9.62</td>
<td>3.13</td>
<td>+0.39</td>
<td>2.624</td>
<td>YES</td>
</tr>
<tr>
<td>P5</td>
<td>9.33</td>
<td>3.39</td>
<td>9.62</td>
<td>3.24</td>
<td>+0.29</td>
<td>1.937</td>
<td>YES</td>
</tr>
<tr>
<td>P6</td>
<td>10.15</td>
<td>2.49</td>
<td>10.09</td>
<td>2.64</td>
<td>−0.06</td>
<td>−0.497</td>
<td>NO</td>
</tr>
<tr>
<td>P7</td>
<td>8.99</td>
<td>3.40</td>
<td>9.89</td>
<td>3.04</td>
<td>+0.90</td>
<td>6.027</td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 4: Description of Final Exam Problems

<table>
<thead>
<tr>
<th>Final Exam Question</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Matrix Arithmetic</td>
</tr>
<tr>
<td>2</td>
<td>Matrix Operations in MATLAB®</td>
</tr>
<tr>
<td>3</td>
<td>Array Indexing</td>
</tr>
<tr>
<td>4</td>
<td>Solving Sets of Linear Equations</td>
</tr>
<tr>
<td>5</td>
<td>Summation of Forces</td>
</tr>
<tr>
<td>6</td>
<td>Looping Structures</td>
</tr>
<tr>
<td>7</td>
<td>Looping and Arrays</td>
</tr>
</tbody>
</table>
One possible reason for the significant drop in the scores on Problem 1 is that during the 2012 offering of the Engineering Models I course, more time was spent in lecture discussing the specifics of matrix arithmetic. In the fall 2013 offering, this information was covered in the lecture video, but less time was spent during lectures on the specifics of performing arithmetic and more on applications, such as solving sets of linear equations and the summation of forces. It is encouraging to see, however, that a substantial increase in the scores on Problem 7 were observed. This was the most difficult problem, from a computing standpoint, of the common problems on the final exam, which indicates that the additional time spent during lecture periods on solving problems may have helped the students in developing a better ability to hand execute code.

**Student Survey Results**

Students were asked to fill out a course survey at the end of Engineering Models I via Blackboard. In order to encourage strong participation, the survey counted as two quiz grades (approximately 1.7% of the final course grade). There were 33 questions on the survey assessing the effectiveness of professors and teaching assistants, difficulty of assignments, topics enjoyed, difficult topics, and the reaction to the videos. 730 students responded to the survey which, discounting students that withdrew from the course, was a 66.5% participation rate.

There were four questions pertaining to the videos:

- Did you watch the videos prior to attending lecture?
  - Always
  - Often
  - Sometimes
  - Rarely

- The videos in Models I were helpful in preparing me for lecture and recitation?
  - Strongly Agree
  - Agree
  - Neither Agree nor Disagree
  - Disagree
  - Strongly Disagree

- I had to watch the videos more than once in order to grasp the material
  - True
  - False

Please explain your response on the videos (Essay)

Figure 1 summarizes the responses to the first two questions. The pie charts summarize the responses of all the students and the bar charts separate the responses by gender. As indicated in the figure, 40% of the students indicated that they *always* watched the videos while an additional 30% *often* watched the videos. There was a big difference in the responses when separated by gender. 57.5% of the female students indicated that they always watched the videos while only 35.9% of the males checked always. 78% of the students agreed or strongly agreed that the videos were indeed helpful in preparing them for lecture and recitation. This response was consistent across gender.
Figure 1: Student Survey Responses to the Videos
694 of the 730 students provided comments about the videos. Information gleaned from these comments include:

- 11.4% of the students mentioned the Blackboard quizzes as a motivational factor for watching the videos.
- 29% of the students indicated that they watched the videos more than once in order to grasp the material. Some of these students indicated that they reviewed all of the videos prior to taking the midterm and final exam and this helped them perform well on exams.
- 14.6% of the students commented that the videos were too long and an additional 3.7% of the students found the videos to be both lengthy and boring.
- 24 of the 81 students who rarely watched the videos indicated that they read through the accompanying PowerPoint slides prior to lecture instead of viewing the videos and this was sufficient for them to grasp the material (and faster).

Not surprisingly, the quizzes at the beginning of lecture were extremely important in motivating students to watch the videos and do the suggested exercises in MATLAB®. The length of the videos and remembering to allow enough time prior to lecture to watch the videos were definitely issues for some of the students. In addition, some students indicated that listening to audio just didn’t work well for them and they preferred to read the PowerPoints instead. Students with prior programming experience indicated that they just needed to look up the changes in syntax between MATLAB® and whatever language they had prior experience with.

Some sample comments from students that indicated that they *always* watched the videos:

- The videos I found to be very helpful, they discussed the concepts and then showed examples, what more could you ask for. I also like the quizzes at the end, I felt that sometimes I thought I understood something then would miss it on the quiz. I could then review just that slide (or slides) to see what i misunderstood.
- The videos I feel are a pivotal aspect in order to well in this course. I did not have prior experience to the matlab program so the videos were my bread and buter in order to understand and fully grasp the concepts that we went over in class.
- The videos contained all the required information for lecture and were very easy to follow. The examples were presented in a clear and logical manner.
- I thought that, although the videos were usually pretty long, they were well worth the watching due to the nature of their content. The material presented in the videos was very easy to grasp and helped immensely in the process of completing quizzes, homework, and labs.
- Being able to watch how a new topic is used prior to the lecture on it was helpful to me. I would already have a basic understanding of the code and lecture would allow me to put all of the pieces together.

Some sample comments from students that indicated that they *sometimes* watched the videos:

- When I watched the videos i understood most of the material we went over in class and knew what i was doing when it came to labs and homework. But when i didnt watch the videos it was tough to follow along.
I thought the videos were a great way of outside instruction. They were informative, and did a great job of covering the topic in question. However, due to the amount of time required for other assignments, I did not have time to watch the videos every week.

- sometimes they were really hard to understand and they were always really boring to watch
- When I made the time to watch the videos, the information was easy to grasp. I sometimes chose to browse through the power point instead of the videos because of their length. While this was not as effective as watching the videos with the explanations it still helped me score well on the lecture quizzes.
- I watched the videos for the first half of the semester, but I found that I could move at a much quicker pace by going through the powerpoint slides alone. So I switched to only reading the powerpoint slides halfway through the semester. However, I do believe that the short quizzes at the ends of the videos were nice. It would be great if those quizzes could be a part of the powerpoints too.

Some sample comments from students that indicated that they rarely watched the videos:

- The videos were long and hard to pay attention to the whole way through. If they weren't so long, I would have watched most of the videos.
- I neglected to watch the videos before class simply because i was too lazy. The videos were EXTREMELY helpful when I was preparing for the midterm/final exams. I would not have done as well if not for them.
- I am a procrastinator, so I usually did not have time to watch the videos before lecture. The powerpoints were very helpful in those situations. However, if I would have made time to watch the videos, it would have helped much more.
- The videos themselves were put together pretty well and did explain the material, but the few I watched I had to watch twice to fully grasp the concept. I also believe that the videos need to be more enjoyable so that people like me don't just skip them.

Several of the students that indicated that they did not watch the videos also commented that they had programming experience prior to taking Engineering Models I and therefore, did not find it necessary to watch the videos. We asked students in the end-of-course survey if they had prior programming experience. 36% of the students indicated that they did indeed have prior programming experience with the most common languages being JAVA, C/C++, and HTML. Figure 2 shows how often the students watched the videos before lecture with prior programming experience taken into account. The results (along with comments in the survey) indicate that many of the students with prior programming experience simply reviewed the Powerpoint slides rather than watching the videos.
The course survey included two questions about the in-class lecture activities:

The Lecture Activities using MATLAB were helpful in preparing me for recitation and homework assignments:

- Strongly Agree
- Agree
- Neither Agree nor Disagree
- Disagree
- Strongly Disagree

Please explain your response on the in-class lecture activities (Essay)

The response to the first question is summarized in Figure 3. 65% of students agreed or strongly agreed that the in-class lecture activities were indeed helpful, 24% were neutral, and 11% disagreed or strongly disagreed that in-class activities were of any use to them.
Some sample student comments about the in-class lecture activities:

- It was helpful for me to work on examples in MATLAB, as that prepared me for recitation and homework assignments. The lectures were very interactive.
- The lecture activities were always very in depth and straight forward. The lecture activities always cleared up any confusion on the topic and strongly prepared me for recitation.
- I liked the lecture activities. I thought they were very helpful for learning how to apply the material. I also thought that some of the topics were helpful too. A couple of the recitation activities talked about the same topics as some of my other classes. For example, when we worked on the while loops my Calculus class was working on infinites series. I don't know if that was intentional but it was definitely helpful.
- Showing or introducing the way to solve real problems was helpful, because it provides a general approach in coding through Matlab, and helps us as students, in preparing on how to write code to solve engineering problems, I think it could be more helpful if some examples in lecture could be focused in personal interest, to give an example: coding a small script that by input counts balls, strikes, outs and advances through innings on a baseball game… , mixing the new knowledge with something that the class may consider fun to do.
- The lecture activities were not very helpful for recitation. The activities often did not tell us what was needed to be done on the labs.
- The lecture was just a repetition of the videos that we watched the night before in order to take the quiz. The lecture felt like a waste of time.

The students that did not feel that the lecture activities were helpful seemed to be divided into two groups. The first group felt that the lecture activities should directly pertain to the recitation assignments and homework assignments. In other words, recitation and homework assignments should consist of taking the lecture examples and plugging in different numbers. The second group consisted of students whose professors apparently chose not to do any in-class lecture activities but instead simply read the PowerPoints slides from the videos that students presumably watched prior to attending recitation. This is a problem that we will clearly need to address.

Discussion

Overall, the inverted classroom appears to have worked well for Engineering Models I. Most students found the videos to be helpful and enjoyed the “hands on” active learning style in lecture. The D-F-W rate dropped significantly from 2012 to 2013 while the final exam scores improved slightly. Many of the students that did not watch the videos indicated that they reviewed the PowerPoint slides instead so they were still coming to lecture prepared.

Some of the videos were a half hour long and clearly need to be shortened by cutting some of the material and breaking them up into shorter videos. It will be interesting to see if our students really are willing to watch three short videos in place of one longer video.
In order to address the needs of students that are not auditory learners, we are planning on developing an interactive learning environment similar perhaps to Code Academy\textsuperscript{10} where students will be able to read a short description of a concept then immediately write and test code in MATLAB\textsuperscript{®}.

**Acknowledgment**

This work was supported, in part, through a MathWorks grant from the Curriculum Development Program in spring 2013. A portion of the grant supported the purchase of the Adobe Captivate software used to create the videos.

**Bibliography**