HANDS-ON LEARNING IN CALCULUS VIA MAPLETS-BASED PROJECTS

Keith Howard
Kenyon College
Gambier, OH 43022
HOWARDK@KENYON.EDU

Lewis D. Ludwig
Denison University
Granville, OH 43023
LUDWIGL@DENISON.EDU

1. Introduction

In a response to meet the ever demanding challenges of vastly different backgrounds, abilities, and interests of their students, many colleges and universities have adjusted their calculus curriculum. A common approach emphasizes concepts over computation using graphical, numerical, and symbolic points of view. Technologies, such as computer algebra systems, have proved valuable with such an approach. For a number of years, faculty members in the departments of mathematics at Denison University and Kenyon College have used the computer algebra system Maple to create in-class labs and demonstrations. The in-class labs have provided students the opportunity for structured, self-pace exploration which help to develop the analytical skills necessary to solve open-ended problems.

Although we were satisfied with the overall results of these labs, it was clear that our existing labs had some inherent problems. After exploring how a number of other institutions were using computer algebra systems, it was clear that the problems with our labs were not unique. The following is a detailed summary of a collaborative effort between Denison University and Kenyon College to address these issues. This works was generously supported by The Mellon Foundation's Collaboration with Technology Grant. The grant was in two parts: Phase I and Phase II. In the summer of 2002, Judy Holdener, Keith Howard, Zaven Karian, and Lew Ludwig were funded to create a number of in-class labs for the calculus sequence. [1] In the spring of 2003, Keith Howard and Lew Ludwig were funded to extend the original labs to a Maplet based platform.

2. Issues with traditional in-class labs

As noted, computer algebra systems are a great benefit to the ever-changing needs of the calculus curriculum. But as with any new technology, there are challenges of how to incorporate it into the classroom. Before this work, the vast majority of the existing in-class labs developed at Dension and Kenyon were totally contained in the Maple worksheet format [2]. This approach required students to open a Maple file, read the directions provided on the screen, and then follow explorations and/or answer questions on the screen. We will refer to this type of lab as an all-in-one-lab, everything was contained and presented in one Maple file. Our existing all-in-one-labs had several issues which we now outline.

[1] These labs are available at www.denison.edu/mathsci/projects/maplelabs.

[2] We found this to be true at our institutions as well as the other institutions we surveyed.

1) Difficult to read/navigate	2) Enter-key syndrome
3) Difficult to grade	4) Lost information

In the all-in-one-lab model, all of the narrative and information regarding the lab was printed on the screen. Often, the text would fill more than one screen and students had to scroll up and down to read and reread the material. Maple is not a typesetting program, so involved mathematical expressions were cumbersome and difficult to read. The narrative was often sprinkled with Maple commands that the students were asked to use. A command such as PLOT would frequently fill the entire screen with a graph and further scramble the lengthy narrative. If students performed this command several times, they had several screens full of output to navigate through before continuing the narrative. If the command was entered incorrectly, the screen was filled with error messages which further scrambled the flow of the lab. Frequently, students would get lost in such a sea of text, error messages, and output and throw their hands up in frustration.

To address the issue of incorrectly entering commands, some labs provided the narrative as well as the Maple commands needed to complete the exercise. Students quickly developed what we term the "Enter-key syndrome." They needed only to locate the Maple command and execute it by pressing the Enter-key without any regard to the narrative. In this manner, they neither learned the Maple commands or the underlining mathematical content presented in the lab.

Grading was a major challenge with the all-in-one-lab model. Through the labs, students often generated sophisticated graphs or lengthy lists of data. It was impractical for these to be copied by hand and ran contrary to the power of technology. Students would print out their results. Lab reports that should have taken one or two pages of results were turning into ten to twenty pages of narrative, error messages, multiple copies of the same command, and large amounts of white space generated improper management of the Maple commands. It was left to the instructor to sort through the jumble to find the pertinent information. Having students submit their work electronically posed similar problems and was not a sufficient solution.

Unlike a traditional tangible assignment, the all-in-one-labs were electronic files that needed to be treated with some care. It was not uncommon for students to lose their assignments through carelessness. This included forgetting to save their work, losing it to a power outage, saving their work to a machine in the lab whose memory was routinely erased, and/or saving a different file over their existing file.

3. Phase I: Addressing the issues of the all-in-one-lab

3.1. **Implementation.** To address the issues of the all-in-one-lab described above, we developed a new style of in-class lab: the 3-component lab. The 3-component lab was based on the discovery learning model and consisted of an exploratory handout, a Maple file, and a follow-up assignment handout (see Appendix 2: Arc Length Lab Handout). The exploratory handout contained the narrative and inquiry-based questions aimed at developing the new content. The Maple file was quite small in comparison to the all-in-one-lab file and contained only the commands necessary to answer the questions on the handout. This was often accomplished via a programmed Maple procedure (function) that the students had to understand how to manipulate effectively, thus avoiding the Enter-key syndrome. Students used the exploratory handout as notes as well as a place to record their findings from the inquiry-based questions. The exploratory handouts were often structured to include real world applications. In this way, students used the discovery method to develop the abstract theory and at the same time extend the theory to relative applications. The follow-up assignment handouts were designed to extend the theory developed in the in-class lab. Often students had to modify the existing Maple procedures or invoke them in a way different from the in-class portion. Thus, extending their comprehension of the theory and the use of the technology. The assignments were designed to be handwritten like a traditional homework assignment.

3.2. **Evaluation.** After the completion of Phase I in the summer of 2002, the labs were used in the calculus sequence the following semester at Denison University and Kenyon College. Overall, the feedback from faculty members and students was positive. The use of the exploratory handout avoided the issues of reading and navigation issues found in the all-in-one-lab format. In an end of the semester evaluation, one student commented, "I like having the handouts instead of all the info being on a Maple file. This way I have something to study and review when it's time for the test." Another remarked, "I used to get so lost in the Maple worksheets, now the handouts keep everything neat and tidy and I only have to use Maple when I need to."

Instructors found that by using questions based on the discovery learning model and keeping the Maple file to a minimum, the Enter-key syndrome was avoided. As one instructor commented on the Arc Length Lab, "I like how this lab develops the arc length formula through the well-known distance formula. Students extend their prior knowledge to develop this new theory. Through the handout, the students developed the theory behind the Maple procedure, then saw how a closed form is not always possible and that approximation via technology is sometimes required."

From the instructor viewpoint, one of the greatest benefits of the 3-component lab was the ease of grading. The assignment handout format avoided the large Maple printouts previously submitted by students. Moreover, since the developmental questions were addressed in the exploratory handout, the assignment handout could address more involved and interesting problems. One instructor went so far as to require the students to submit their answers through a MS Word template. In this way, students could cut and paste relative graphs and present their material in a more legible, readable format.

3.3. **Issues.** Although many of the previous problems from the all-in-one-lab format were addressed in the 3-component lab, the new labs still had some issues that needed to be addressed. Due to file location issues, the 3-component labs contained all the code for the Maple procedures which students had to activate in order to use. At times the code was quite long and would fill more than a screen, so some savvy navigation was still required at times. There was also some confusion on the part of the student on how to implement the Maple procedures. The students still needed a basic knowledge of the Maple commands and their structure to effectively implement the new procedures.

4. Phase II: Development of Maplet labs

After the initial evaluation of the 3-component lab approach, it was clear that each institution still had to spend valuable class time teaching students how to use Maple before the students could effectively interact with the procedures developed in the new labs. Around the time the first set of labs were nearing completion in the summer of 2002, the makers of Maple released version 8. The main advance in this version was the ability to create applet-like procedures called Maplets. Applets are a Java based graphic user interface that have been widely used on the web. They are generally self contained applications which perform tasks for the user without the user having to know any specific programming language. Maplets were designed to provide such "black boxes" for users while using the power of Maple commands and procedures. In short, a student could use a Maplet without knowing how to use Maple. This approach seemed ideal for some of our labs. Specifically, this approach seemed well suited for the labs that were heavily dependent on visualizations that the students would use over and over in the discovery method.

4.1. **Implementation.** Our initial plan was to convert all of the labs from Phase I into the Maplet format. However, we quickly found that the Maplets required a good deal of programming sophistication. Not all of our existing procedures converted well to the Maplet structure. Therefore, we focused on 4 labs: The Mean Value Theorem, The Fundamental Theorem of Calculus, Arc Length, and a new lab on Parameterizations.

To describe the conversion process to the Maplet format, we consider the Arc Length Lab described earlier. Since this lab required a minimal amount of Maple commands and relied heavily on visualizations, it was a natural candidate for the Maplet format. The previous Arc Length lab had two procedures. One, `Circum(f,a,b,n)`, was a graphical visualization of a polygonal line of length n approximating a function f on the interval $[a, b]$. The second procedure, `Polylen(f,a,b,n)`, computed the length of said polygonal line via an extension of the distance formula. Since the Maplet structure allows multiple outputs and graphing to occur in the same window, these two procedures were condensed into one Maplet window (see Figure 1). Students provided the function f and the interval $[a, b]$. Then they could interactively adjust the number of sides of the polygonal line. The length of the polygonal line was recalculated after every adjustment in the number of sides. This Maplet freed the student from the burden of the procedure syntax. Moreover, the visualization of the graph and the calculation of the length of the polygonal line provided a more meaningful demonstration of the approximation process. The Maplet structure can be exported to its own stand alone file (provided the machine contains Maple). In this way, a

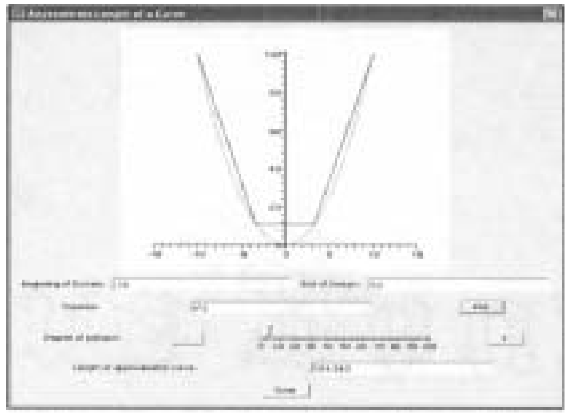

FIGURE 1. Maplet

student could run the Maplet without opening and executing a Maple worksheet.

4.2. **Evaluation.** At the time of this publication, a formal evaluation of the Maplet version labs has not been conducted. The labs were completed in the early summer of 2003 and the topics covered by the labs have not been addressed yet in the fall semester. Initial comments from instructors testing the labs favor the "clean" presentation of the procedure in a Maplet, but there are concerns that students will even further avoid Maple if not required to use the worksheet format. Commenting as the developers of these labs, we can definitely see the worth of the Maplet format, but it takes quite a time investment to acquire the skills necessary to develop the Maplets. So, one should weigh the benefits of the Maplet format against the time required to develop a functioning Maplet.

5. Conclusion

Through our own observations and those of other instructors and students at Denison University and Kenyon College, it is clear that this was an informative and beneficial endeavor. We found that the 3-component lab format was a great improvement on the all-in-one-lab format. It clearly resolved three of the main issues of the all-in-one-lab format: difficult to read and/or navigate, challenge to grade, and the Enter-key syndrome. At this point, it is unclear how we will proceed with the development of Maplet based labs. On one-hand, the Maplet format offers a tidy way to visualize a mathematical concept, with minimal knowledge of Maple. On the other hand, we expect our majors to have a working knowledge of Maple and these skills were previously developed in the calculus sequence. Converting all of our labs to Maplets may not be best for our majors. On the whole, the Maplet labs took a great deal more time to develop than the procedure based labs. Clearly, our inexperience in the programming of Maplets contributed to this, but on the whole the Maplet labs required more code than the procedure based labs. Hopefully, as the makers of Maple refine the Maplet protocol, the programming of Maplets will become more accessible.