INCORPORATING MATLAB WITHIN MATHEMATICS COURSES

David Szurley
Francis Marion University
Department of Mathematics
Florence, SC 29501
dszurley@fmarion.edu

1 Introduction

Computers play a major role in today's business and academic settings. Exposure to commercial packages is a major advantage for graduates seeking employment or continuing their studies at graduate school. Matlab is a high-level language that provides an interactive graphical user interface. Matlab is featured in hundreds of textbooks in engineering and the sciences, and is integrated in the curriculum at many universities [1]. This paper will describe three instances in which Matlab has been used in different capacities by the author. The experiences from these instances will be mentioned, and some concluding thoughts will be presented.

2 Calculus

The third semester calculus course (Calculus III) at Francis Marion University contains the following topics (among others).

- L'Hôpital's Rule
- Improper integrals
- Sequences and series
- Power series
- Taylor and Maclaurin series

Students are generally sophomores when they take Calculus III. Majors include physics, mathematics, and chemistry. Many of these topics deal with the question "What happens when a quantity tends to infinity?". Matlab was used within projects to obtain numerical solutions to various problems. In order to facilitate this, the author created Matlab m-files which would solve a similar problem. Students were asked to change parameters within the Matlab m-files and rerun to obtain a result for their particular problems. Some projects were geared towards understanding programming concepts (such as loops, case structures, input/output, etc.) so that the students could gain a better appreciation of the written code.

The students would receive a project dealing with a certain topic within Calculus III (say improper integrals). Generally the project involved students executing the supplied m-file to obtain a solution to a problem the author had chosen. The project would then ask the student to edit the m-file to change certain aspects of the problem. For improper integrals, one example would be the integrand. Then, students would be asked to rerun the m-file to obtain a solution to the new problem. These projects were generally collected after two weeks.

Suppose we are considering L'Hôpital's Rule. A project was assigned for which the supplied m-file plotted the function $f(x) = (\cos x)^{\frac{1}{x^2}}$ on $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Students were asked to change the endpoints in order to plot $f(x)$ over the intervals $\left[-\frac{1}{2}, \frac{1}{2}\right]$, $\left[-\frac{1}{4}, \frac{1}{4}\right]$, and $\left[-\frac{1}{8}, \frac{1}{8}\right]$. Based upon the plots that Matlab produces, students were asked to guess the value of $\lim_{x\to 0^+} (\cos x)^{\frac{1}{x^2}}$. Students are also asked to change the function $f(x)$ the m-file is plotting, and to plot the new function over a set of intervals to guess a value of an appropriate limit.

For another example, we may consider improper integrals. Students were given m-files that would numerically integrate a function $f(x)$ over a finite interval $[a, b]$. In the beginning, the m-file would numerically calculate the value of $\int_1^{10} \frac{1}{x} dx$. In order to understand the behavior of $\int_1^{\infty} \frac{1}{x} dx$, students were asked to change the upper limit of integration and create a table relating the value of the upper limit to the value of the integral. As with L'Hôpital's Rule, this project would eventually require the student to change the form of the integrand $f(x)$.

Projects were also created that would plot the values of a sequence $\{a_n\}$ over some interval $[1, n]$ which would require the students to determine the behavior of $\lim_{n\to\infty} a_n$. Similar to the project involving L'Hôpital's Rule, the students would be asked to increase the upper limit on the interval $[1, n]$ or the form of the terms of $\{a_n\}$. The students were required to have changed both of these by the end of the project. With some minor changes, this project was changed for a similar project involving series.

There were many obstacles involved with assigning these projects. For each project, the author spent about the equivalent of one class period outlining what was to be done in order to accomplish the objectives for the projects during class. This amounted to about four complete class periods through the semester devoted to answering questions about the projects.

There were many questions involving Matlab syntax, as expected. In answering these questions, they appeared to follow two general themes.

- The difference between symbolic and non-symbolic mathematics.
- Array arithmetic syntax within Matlab.

Unlike Maple, Matlab does not work with symbols such as x or $f(x)$. Matlab only works with arrays of numbers. Students generally know that when working with an independent variable x it represents a set of real numbers for which a function $f(x)$ is defined. While Maple also interprets x in this manner, Matlab requires x to be defined as an array of real numbers. Then $f(x)$ may be defined as an associated array of real numbers, and then the points may be plotted (and the points connected) to obtain a graph of the function $f(x)$. There were two main complications with this concept. First, students who were familiar with Maple did not understand why x could not just represent x (as opposed to an array of numbers). Second, students did not understnad why they could not just use their calculator to produce the same plot. In order to address the former, the differences between Matlab and Maple had to be explained, and for the latter, the author has the opinion that Matlab is more powerful and diverse than many calculators.

When working with a array x in Matlab, it is often necessary to perform the same operations on each idividual component of x. To this end, Matlab uses the "dot-star" notation for array arithmetic. For example, in order to multiply like components of the arrays x and y, we would type `x .* y`. Students would tend to leave out the dot, which in many cases would produce an error.

Students were less than enthusiatic about the projects. The other sections of Calculus III do not require Matlab projects. Hence it is the author's opinion that the students believe that the Matlab projects are above and beyond what is to be expected of them. However, the author believes that the experience gained from this limited exposure to Matlab will benefit the students eventually. Future efforts include creating a graphical user interface in which changes to the m-file would be made via entry in text boxes.

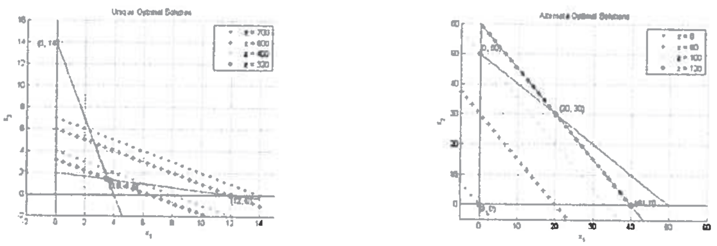
3 Linear Programming

Francis Marion University offers a course in linear programming. Among many others, a major topic is the graphical solution of linear programming problems. Students who take this course are typically juniors or seniors in standing and include mathematics and computer science majors.

Although graphical solutions to linear programming problems may be explored by many means, the author chose to use Matlab. This entails

- drawing the feasible region,
- determining the extreme points, and
- drawing level curves of the objective function.

Examples of solutions may be seen in Figure 1. Students were not asked to edit the m-files used to produce graphs such as in Figure 1. Solution cases were simply pre-



(a) Unique optimal solution (b) Alternate optimal solutions

Figure 1: Graphical solutions to linear programming problems via Matlab.

sented in class. As a whole, the class seemed generally receptive to illustrating these cases on Matlab as opposed to illustrating them on a chalkboard. The author made these graphs available to students in a file that was distributed during that particular class period. It is the opinion of the author that the students appreciated being able to devote more time to making notes on the graphs instead of copying the graphs themselves.

While beginners to Matlab may not be able to produce graphs such as those in Figure 1 by themselves, some future work would be to create a graphical user interface. The student could enter the linear programming problem into various text boxes and the rest could be automated by Matlab.

4 Linear Algebra

In the spring semester of 2009 the author had the occasion to teach linear algebra. There are many applications that could be covered during a first semester linear algebra course. The author chose four topics not normally covered in the typical curriculum of a mathematics undergraduate major and created semester-long projects. These semester projects had two main components.

- A theoretical portion in which students were asked to comprehend the topic.
- A computational portion in which students were asked to simulate certain aspects of the topics with Matlab.

The topics, along with some objectives, are the following.

- Computer Graphics
  Students were asked to use the standard matrices for linear transformations in order to investigate their effect on a square in the *xy*-plane.
- Markov Chains
  Students were given a two-state Markov chain and were asked to setup and solve the linear system to determine the long-term probabilities.

- Dynamical Systems
  Students were given a two-dimensional continuous time dynamical system and were asked to use eigenvalues and eigenvectors to determine its general solution.

- Data Interpolation
  Students were given a set of real-world data and were asked to setup and solve the linear system that determines the coefficients of the interpolating cubic spline.

Students were given an m-file to accomplish similar tasks for each project. Much of what was required of the students computationally was to edit the m-file to reflect their particular problem. Unfortunately, as of this writing, the projects were not completed. However, students seemed to be generally surprised that mathematics is used within real-world applications. They implied that they thought mathematics only existed in the classroom without any uses outside of teaching.

5 Conclusions

Overall, regardless of the caliber of the student, these projects have been less than well received. Again, it is believed that since other instructors do not require Matlab, the students feel this is more than what should be asked of them. In the future, graphical user interfaces will be developed that will accomplish many of the same tasks but remove the nuances of Matlab from student control. However, the author still believes that the long-term benefits of the exposure to any commercial package far outweighs the drawbacks.

References

[1] Business/Technology Editors. (2001). *The MathWorks Announces Worldwise Availability of Matlab Students Release 12.* Retrieved February 7, 2007, from BusinessWire. Web site:
http://www.businesswire.com/webbox/bw.012401/210242193.htm.

[2] Lay, D. (2003). *Linear Algebra and Its Applications.* New York: Addison-Wesley.

[3] Heath, M. (1997). *Scientific Computing.* St. Louis: McGraw-Hill.

[4] Ross, S. (1997). *Introduction to Probability Models.* New York: Academic Press.

[5] Strogatz, S. (1994). *Nonlinear Dynamics and Chaos.* New York: Addison-Wesley.

[6] Zill, D. (1994). *Differential Equations with Computer Lab Experiments.* New York: PWS Publishing Company.