CS100B Fall 1999

Professor David I. Schwartz

Programming Assignment 6

Due: Thursday, December 2, 1999

1. Goals

In this assignment, you will practice using MATLAB.

2. Problems

Write an M-File that performs the Trapezoidal Rule to compute ***area*** above and below a function.

2.1 Overview

Your program must compute the area “underneath” and “above” a function of one variable x:

- When the function is positive, find the area below the function.
- When the function is negative, find the area above the function.

Sum all portions of area together to produce the total area.

2.2 Specifications

Your program should have the following features:

- The user supplies the range, **a** to **b**, and function, **f**, to integrate $f(x)$:

$$\int_a^b f(x)dx \text{, where } a \le x \le b .$$

- The user supplies **n**, the number of trapezoids to use within each zone of positive or negative behavior of the function.
- You may assume the function is smooth and continuous within $a \le x \le b$.
- Your program must detect if the sign of $f(x)$ is negative. Why? Area is a positive quantity. Hint: Think root finding. Either use a MATLAB function or use a numerical technique. If you use a numerical technique, like LHS/RHS or Newton, use MATLAB’s tolerance **eps**.
- Your program must report the total area.

2.3 Things To Do

For the problems listed in Section 2.4:

- Plot the function. You must clearly label the axes and title the plot according to test case. Why? You should “get a feel for” the function behavior before analyzing!

- Run your Trapezoidal Rule program for values of **`n`** of **`1`**, **`10`**, **`100`**, and **`1000`** within each zone of positive and negative function behavior.
- Use MATLAB's function **`trapz`**.
- Use MATLAB's function **`int`** for numerical solutions. Hints: Investigate **`int`**, **`syms`**, and **`eval`**. For a "prettier" solution, investigate Maple. If you wish, run Maple and enter the command **`?int`**. Try **`?student[trapezoid]`**, too.

2.4 Problems

1. $\int_0^1 dx$ (always test a program with the "easiest" thing you can think of, something you can do "by hand" that will provide an easy check)
2. $\int_0^{2\pi} \sin x dx$ (always test a program with something that exhibits a range of behaviors)
3. $\int_0^{\pi/4} \sqrt{\tan x} dx$ (test a program by battering it with something bizarre)

3. What To Hand In

Submit the following. Assemble your printouts in a clear fashion:

- Title sheet with name(s), ID(s), Section(s), Instructor, Section Day/Time.
- All MATLAB code. Be sure to comment all crucial portions and use a clear, consistent style.
- All results of your MATLAB code for the values of **`n`** of **`1`**, **`10`**, **`100`**, and **`1000`** for all problems.
- The results of **`trapz`** and **`int`** for each problem. You should perform these runs along with the runs of your MATLAB programs for a more organized report.
- A comparison of results from your plots, your MATLAB program, **`trapz`**, and **`int`**. How closely do the results of your program and **`trapz`** compare? Do the increasing values of **`n`** improve or lessen the accuracy of the trapezoidal rule? Type your answers!

4. Submitting Your Work

4.1 Due Date

This assignment is due in lecture on Thursday, December 2, 1999. You may submit your assignment to a consultant *before*, but not on, that date in the consulting room in Carpenter. Late programs will not be accepted.

4.2 Labeling Your Work

Always write your name, Cornell ID#, and the day/time/instructor for your section in the first comment of each program you hand in for credit. You must type all solutions and commentary! If you wrote the program with a partner, turn in only one printout with your partner's name and ID# in the comment, as well as your own. The comment must also include the section day/time/

instructor for the partner. The program will be returned to the first person listed. Sign your name(s) in the comment. Please staple the pages of your assignment together.

4.3 Grading

This assignment will be given two grades: the first based on correctness, the second on program organization and style. Each grade will be a 0, 1, or 2. Not only should your program work, but it should contain adequate comments to guide a reader interested in understanding it. The declaration of every significant variable should include a comment describing that variable. Include appropriate comments in the code so the reader can see the structure of the program, but not so many that the reader has trouble reading program text.

4.4 Academic Integrity

All work submitted should be your own or your partner's. The output submitted should exactly match the code submitted. Issues of academic integrity will be taken very seriously. See the Academic Integrity link from the CS100B web page for more information.

5. Acknowledgments

The CS100B staff would like to thank all those who contributed to and reviewed this assignment, especially Raju Rohde