Fourth International Derive TI-89/92 Conference
Liverpool John Moores University, July 12 – 15, 2000

Graphical Representation of Riemann Sums

Dr. Joel P. Lehmann
Valparaiso University, USA
Email: joel.lehmann@valpo.edu

In the Spring semester of 2000 I taught a second semester calculus class in a 3 plus 2 format (three hours of lecture per week plus a two hour computer laboratory) which used DERIVE as the computer algebra system. My guiding principles for planning laboratory assignments and activities were:

1) Activities should be centred around things that could not be done without a computer;

2) Assignments might start with rote following of instructions but should quickly challenge the students to think through the problem and ponder the meaning of what they were doing (it was a challenge to get students to believe assignments could not all be completed during the laboratory time);

3) Assignments should encourage (perhaps, force) students to learn how to explore and play with the concepts -- to observe and speculate; to form their own hypotheses and constructs. This was an additional reason for lab groups rather than individual effort;

4) Don't waste their time by giving make-work or time filling assignments.

Having to invent my own model for what a calculus laboratory ought to be pushed me to examine the "traditional" calculus class model I had learned in and taught for many years: to assess its strengths and weaknesses, to consider the capabilities of computers and how they could be used in the learning process, and then to look at how the lectures and laboratories could best complement each other. Then it would be time to experiment.

The Case of Riemann Sums

The "traditional" calculus class is the legacy of generations of mathematicians (including my generation) who lived in a time when computation was at best tedious, error prone and time consuming. Calculations were deferred as long as possible; algebraic manipulation was used as much as possible to simplify expressions and reduce the amount of actual calculating necessary. Algebra was the trusted tool of choice.

This preference for symbol manipulation before evaluation can be seen in the treatment of definite integrals. The definite integral may be introduced in terms of Riemann sums, but the major emphasis and time spent is traditionally on the Fundamental Theorem of Integral Calculus and on problems that can be solved exactly using algebraic methods. The message to the student is that a geometric understanding of what they are doing is an introductory digression but the "real" integration involves algebraic manipulation. Numerical methods for evaluating definite integrals may make a brief appearance later, presented as a last alternative when all else fails. This reinforces the belief in integration as algebraic manipulation, with numerical methods a means of treating the rare exception that does not yield to algebra.

There are serious pedagogical concerns about this approach. That the brief look-in at Riemann sums is just enough to be confusing is minor compared to the student failing to build a conceptual framework on which to build an understanding of the rest of the course. When (or if) numerical methods are encountered they are then also likely to be seen as algebraic formulae to be used without thinking or understanding. For many (most?) students applications of the integral remain a mystery, because setting up the integrals needed to find volumes of revolution, arc length, work, hydrostatic pressure, etc., requires understanding the concepts of Riemann sums. Sadly, in this traditional class a student may be “successful”, i.e., pass the course, without ever coming to grips with understanding Riemann sums and thus without ever understanding the definite integral.

Students have several problems when encountering Riemann sums. One problem is visualisation: how do the geometrical figures being used to approximate a definite integral (rectangles or trapezoids) relate to the behaviour of the function (monotonicity, concavity, etc.). A second problem is the brevity of the treatment: the amount of effort involved in doing a problem by hand with more than a trivial number of subintervals precludes doing more than one or two token examples before moving on to the Fundamental Theorem. Students do not get enough experience to be comfortable with the concept and see it as a digression to be forgotten once the "real" integrals appear.

The advent of graphing calculators and computer algebra systems is slowly changing this, however. Computers do bring solutions as well as problems. The effort involved in drawing rectangles and doing sums can be turned over to the computer, with graphs of very complicated functions and large numbers of rectangles easily generated. It becomes feasible to do many more examples using varying and larger numbers of intervals on functions beyond the linear or quadratic. Student time can be spent on understanding the concepts, not just on tedious calculations. There is a marvellous potential for guided discovery learning.

This ease of computation means numerical methods are becoming increasingly important as problems and topics become more realistic, recognising that “real world” problems in engineering, physics, statistics, etc. rarely have nice integrals that yield to exact methods. Problems need no be longer bound by constraints of producing integrals that conform to standard techniques of integration involving nice, easily calculable numbers. The vast resource of “nice” problems still makes up the bulk of current textbooks, but reality is beginning to creep in.

The Riemann Sum Laboratory Exercise

The first laboratory assignments in my calculus II class dealt with Riemann sums and an introduction to the definite integral. Prior to the actual laboratory the students were given a preliminary non-computer assignment. This introduced the idea of partitioning an interval and approximating area using rectangles, and then guided them through a traditional exercise:

Given the function and a hardcopy graph of the function

a) partition the interval into 4 subintervals;
sketch rectangles using left-hand endpoints and sum their areas;

b) repeat for right-hand endpoints.

c) observe which of these overestimates the actual area and which underestimates the area, and think about how these two values could be used to get a better approximation.

d) repeat a) - c) for n = 8 subintervals.

By the time the students came to the lab, they had hands-on experience of the partition-and-approximate process for a simple function and small value of *n* that could be built upon in the laboratory. They had also been led to consider increasing the number of partitions as a way to reduce error.

The first part of the actual laboratory assignment had the student groups (2-3 students per group) generalise the preliminary assignment to get three formulae for Riemann sums using left-hand endpoints, right-hand endpoints and midpoints. They were led through the process of converting their formulae into sigma notation and from there to DERIVE functions that could calculate the sums. The approximations done by hand in the preliminary assignment were used to test their functions. They now felt ready to proceed.

The next step was for them to use their three functions to approximate the area under on selected intervals [a, b], with n varying from 10 to 500. They were asked to observe when increasing n increased their sums and when it decreased them. The point was to have the student observe that on some intervals, increasing n had the left-hand estimate increasing and right-hand estimate decreasing, whereas on other intervals the effect was reversed, and on a symmetric interval they both behaved the same. This was to pique them into noticing that there was more to consider than just formulae and mindless number crunching.

At this point I provided them with the functions to draw the approximating rectangles. To illustrate the format of these functions, the function to draw rectangles using the left-hand endpoint, was called DRAW_LEFT_RECT with four parameters as follows:

```
DRAW_LEFT_RECT(f, x, a, b, n)
```

Parameters: `f` is the function being integrated
`x` is the argument of the function f
`a` and `b` are the endpoints of the interval of integration
n is the number of rectangles.

Approximating this function generates a vector of points which, when plotted, draws the approximating rectangles for the sum.

The third part of the assignment had students exploring conditions under which one would get an upper sum or a lower sum, or would be uncertain if it was either. They were given an assortment of carefully selected functions and intervals, and used their three DERIVE functions to compute sums and my functions to illustrate the rectangles being summed. From these they could make observations and draw conclusions.

The point of this part of the assignment was to reinforce the connection between the formulae being used to generate numbers and the graphical picture that let them interpret the numbers in a meaningful way. It was also a clear illustration of the concepts of upper and lower sums.

Student reaction was very positive. Different methods could be easily compared, large values of n did not present any problem of complication or tedium, and changing intervals or functions was also almost painless. Numbers generated by the students' DERIVE functions took on meaning when they could be compared to graphs with the approximating figures drawn in. What had been just piles of digits that they generated and then asked me if they looked right now could be seen as real areas, with a visual image of the accuracy of the approximation. As a side note, the students really liked watching the figures being drawn (especially in colours) and experimented with them far beyond what was assigned to do.

Later laboratory assignments built on this assignment:

> In using their Riemann sum routines on functions which were negative on part of the interval under consideration, the graphs of the approximating rectangles made very clear the reason for an otherwise mysterious behaviour: increasing the interval reduced the area. The concept of relative area and a generalisation of the interpretation of what a definite integral meant was much less confusing.
>
> In addition to the three initial drawing routines I added three more to graph rectangles using the endpoint with the maximum (or minimum) function value on each subinterval (almost upper or lower sums) and using trapezoids. Being able to look at the approximating figures helped them build a conceptual framework on which to make sense of what they were doing. When we considered truncation errors for these methods, the relation between the type of approximating function (constant for rectangles, linear for trapezoids) and the appropriate derivative of the function being integrated made much more sense. They can see that derivatives are much more than just an algebraic manipulation of a given formula.
>
> A more challenging assignment was to apply the same techniques when one had tabulated data rather than an algebraic expression. While in principle an easier form of what they had already done, with the step of going from a formula to a vector of points (the partition) removed, this

was perceived as a more complex and totally new situation, perhaps even threatening since there was no algebra to do or formula to plug into a calculator. However, it is clearly a situation that is likely to arise in a non-classroom setting where data acquisition or observation, not an a priori algebraic formula, provides the input. Because it is both unfamiliar and important, it is an area on which to concentrate next.

The Functions

I wanted to have functions that were as self-contained as possible, and did not depend on defining a function f(x) elsewhere, or values for a, b or n defined elsewhere. I prefer to avoid surprise results if possible (students take a while to learn that deleting a statement does not remove the effect that statement had). A sample of functions follows.

```
DRAW_LEFT_RECT(f, x, a, b, n) := APPEND(APPEND(VECTOR(APPEND([[a + (b - a)/n*i, 0]], [[a + (b - a)/n*i, LIM(f, x, a + (b - a)/n*i)]], [[a + (b - a)/n*(i + 1), LIM(f, x, a + (b - a)/n*i)]]), i, 0, n - 1)), [[b, 0]])
```

Notes:

i) This function generates successively vectors of three points of the rectangle,

$$[[x_i, 0],[x_i, f(x_i)],[x_{i+1}, f(x_i)]] \quad i = 1, 2, 3, \ldots \text{n-1}$$

then concatenates them with the final point at [b, 0] .

ii) The limits in the above merely substituted the appropriate expression for the xi into the function f.

```
DRAW_RIGHT_RECT(f, x, a, b, n) := APPEND(APPEND(VECTOR(APPEND([[a + (b - a)/n*i, 0]], [[a + (b - a)/n*i, LIM(f, x, a + (b - a)/n*(i + 1))]], [[a + (b - a)/n*(i + 1), LIM(f, x, a + (b - a)/n*(i + 1))]]), i, 0, n - 1)), [[b, 0]])
```

```
DRAW_MID_RECT(f, x, a, b, n) := APPEND(APPEND(VECTOR(APPEND([[a + (b - a)/n*i, 0]], [[a + (b - a)/n*i, LIM(f, x, a + (b - a)/n*(i + 1/2))]], [[a + (b - a)/n*(i + 1), LIM(f, x, a + (b - a)/n*(i + 1/2))]]), i, 0, n - 1)), [[b, 0]])
```

```
DRAW_TRAP(f, x, a, b, n) := APPEND(APPEND(VECTOR(APPEND([[a + (b - a)/n*i, 0]], [[a + (b - a)/n*i, LIM(f, x, a + (b - a)/n*i)]], [[a + (b - a)/n*(i + 1), LIM(f, x, a + (b - a)/n*(i + 1))]]), i, 0, n - 1)), [[b, 0]])
```

```
DRAW_MAX_RECT(f, x, a, b, n) := APPEND(APPEND(VECTOR(APPEND([[a + (b
- a)/n*i, 0]], [[a + (b - a)/n*i, MAX([LIM(f, x, a + (b - a)/n*i), LI
M(f, x, a + (b - a)/n*(i + 1))])]], [[a + (b - a)/n*(i + 1), MAX([LIM
(f, x, a + (b - a)/n*i), LIM(f, x, a + (b - a)/n*(i + 1))])]]), i, 0,
 n - 1)), [[b, 0]])
```