Using the TI-83 for Simpson's Rule

In order to use this information, you need to be familiar with creating and storing lists on the TI-83. This information is illustrated in the earlier documents *Approximating Areas on the TI-83* and *The Midpoint Rule on the TI-83* (see the Daily Assignments section of the web syllabus).

You also need to know an additional fact about handling lists: we have two lists of the same length, the TI-83 will "multiply" them "term-by-term" to create a new list. For example, if $L1 = \{2,4,6\}$ and $L2 = \{3,2,7\}$, then on the TI-83

 $L1*L2 = \{2*3, 4*2, 6*7\} = \{6, 8, 42\}$

Recall Simpson's Rule: to approximate $\int_a^b f(x) dx$, subdivide [a, b] into n equal subintervals, each of length h (or Δx) = $\frac{b-a}{n}$. For Simpson's Rule, n must be an <u>even</u> positive integer. The endpoints of the subintervals are $x_0 = a, x_1, x_2, ..., x_{n-1}, x_n = b$.

Simpson's approximation to $\int_a^b f(x) dx$ is then given by the formula

$$S_n = \frac{h}{3}(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + 4f(x_{n-1}) + f(x_n))$$

For a small n, it's easy to compute this. For a larger n, computing it one term at a time gets tedious, and it's harder to see how to do it efficiently on the TI-83 (unlike, say, for the Midpoint Rule) because of the pattern of "switching coefficients" $\{1, 4, 2, 4, 2, 4, ..., 2, 4, 1\}$. Here is a series of steps that do the job, followed by a detailed discussion of why they work.

Enter the function $Y1=1/(1+x^2)$ in the "Y=" window. Store the values of $a, b, h (= \Delta x)$ in the TI-83

a STO A
b STO B
n STO N

$$(B - A)/N$$
 STO H
seq(A+I*H, I, 1, N - 1) STO L1
seq(3+(-1)^(I - 1), I, 1, N - 1) STO L2
 $(H/3)^*(Y1(A) + sum(Y1(L1)*L2) + Y1(B))$ (Gives the value of S_n)

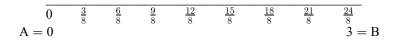
Then

Here's a step by step discussion of what's happening in a specific case. We compute S_8 for $\int_0^3 \frac{1}{1+x^2} dx$.

Enter $Y1(X)=1/(1+x^2)$ in the "Y=" window.

0 STO A	Stores 0 in location A
3 STO B	Stores 3 in location B
8 STO N	Stores n in location N
(B - A)/N STO H	Stores $h = \Delta x = \frac{3}{8}$ in location H

The subdivisions of [0, 3] and their endpoints look like



We want to find

$$S_8 = \frac{H}{3}(Y1(A) + 4^*Y1(\frac{3}{8}) + 2^*Y1(\frac{6}{8}) + 4^*Y1(\frac{9}{8}) + 2^*Y1(\frac{12}{8}) + 4^*Y1(\frac{15}{8}) + 2^*Y1(\frac{18}{8}) + 4^*Y1(\frac{21}{8}) + Y1(B))$$

We need to evaluate the function Y1 at the <u>9 points</u> in list $\{A, \frac{3}{8}, \frac{6}{8}, \frac{9}{8}, \frac{12}{8}, \frac{15}{8}, \frac{18}{8}, \frac{21}{8}, B\}$ No matter how large *n* is, A and B are "special": those are the only points in the sum for S_n where the coefficient in the formula is one. So we treat them as "special" and make a list of the other N – 1 (=7) points and store in the list variable L1:

seq(A+I*H, I, 1, N - 1) STO L1 Creates the list

$$L1 = \{\frac{3}{8}, \frac{6}{8}, \frac{9}{8}, \frac{12}{8}, \frac{15}{8}, \frac{18}{8}, \frac{21}{8}\}$$

(Be sure you think about why!)

In the formula for S_8 , the pattern of coefficients associated with these endpoints looks like

$\{\frac{3}{8},$	$\frac{6}{8}$,	$\frac{9}{8}, \frac{12}{8},$	$\frac{15}{8}$,	$\frac{18}{8}$,	$\frac{21}{8}$
		ă, 2,			

We create a list of these coefficients and store it in the list variable L2:

The commands:

Y1(L1)	Creates $\{Y1(\frac{3}{8}), Y1(\frac{6}{8}), Y1(\frac{9}{8}), \dots, Y1(\frac{21}{8})\}$
Y1(L1)*L2	Creates $\{4*Y1(\frac{3}{8}), 2*Y1(\frac{6}{8}), 4*Y1(\frac{9}{8}), \dots, 4*Y1(\frac{21}{8})\}$
sum(Y1(L1)*L2)	Creates $4*Y1(\frac{3}{8}) + 2*Y1(\frac{6}{8}) + 4*Y1(\frac{9}{8}) + \dots + 4*Y1(\frac{21}{8})$

Therefore we can get S_8 from the command

(H/3)*(Y1(A) + sum(Y1(L1)*L2) + Y1(B))

If you do these steps on the TI-83, it gives the result 1.24879899. In this example, we happen to know that the exact value of $\int_0^3 \frac{1}{1+x^2} dx = \arctan 3$. So, just out of curiosity, we can see how good or bad the approximation is:

 $\text{Error} = (\text{Exact value}) - (\text{Approximation}) = \arctan 3 - 1.248799 \approx 0.000246782.$

Of course, if you follow the steps above, it's no more work to compute, say, S_{100} .