5.4 Project
Calculator/Computer Riemann Sums

Suppose you want to approximate the integral $\int_a^b f(x)\,dx$ numerically using *midpoint sums*. If $\Delta x = (b-a)/n$ and

$$m_i \;=\; x_i - \tfrac{1}{2}\Delta x \;=(a + i\,\Delta x) - \tfrac{1}{2}\Delta x \;=\; a + \left(i - \tfrac{1}{2}\right)\Delta x \qquad (1)$$

is the midpoint of the ith subinterval $[x_{i-1}, x_i]$, then the selection $x_i^* = m_i$ in Eq. (14) of the text gives

$$\int_a^b f(x)\,dx \;=\; \lim_{n\to\infty} \sum_{i=1}^{n} f(m_i)\,\Delta x. \qquad (2)$$

Many calculators and computer algebra systems include "sum" commands that can be used to calculate easily and rapidly the midpoint sum with larger and larger values of n. A common practice is to start with perhaps $n = 50$ subintervals, and then calculate midpoint sums with successively doubled numbers of subintervals — that is, with $n = 50, 100, 200, \cdots$ — until successive sums appear to agree to the desired number of decimal places of accuracy.

In the following paragraphs we illustrate this procedure using graphing calculators and typical computer algebra systems. As an example we consider the integral $\int_0^2 \sqrt{x}\,dx$, whose exact value is known (see Section 5.5) to be given by

$$\int_0^2 \sqrt{x}\,dx \;=\; \tfrac{2}{3}\cdot 2^{3/2} \;=\; \tfrac{2}{3}\sqrt{8} \;\approx\; 1.8856. \qquad (3)$$

With $a = 0$ and $\Delta x = 2/n$, Eq. (2) above gives

$$\int_0^2 \sqrt{x}\,dx \;=\; \lim_{n\to\infty} \Delta x \sum_{i=1}^{n} \sqrt{(i - \tfrac{1}{2})\Delta x}. \qquad (4)$$

You can use the illustrated techniques to carry out the following investigations.

1. Approximate the integral $\int_0^{\pi} \sin x\,dx \;=\; 2 \;\approx\; 2.0000$ of Problem 56 accurate to four decimal places.

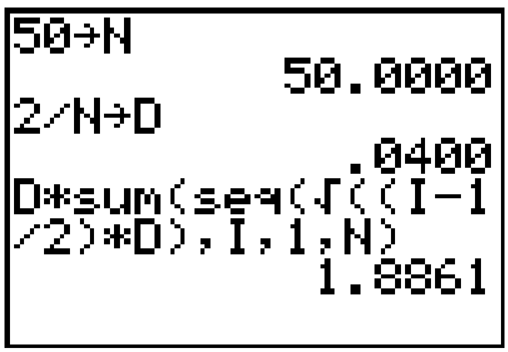
2. First explain why Fig. 5.4.13 in the text and the circle area formula $A = \pi r^2$ imply that

$$\int_0^1 4\sqrt{1-x^2}\,dx \;=\; \pi.$$

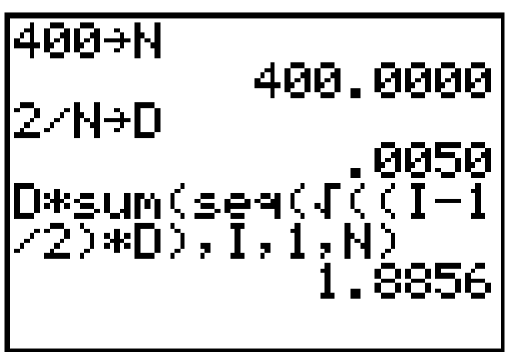
Then use midpoint sums to approximate this integral and thereby the numerical value of π. Begin with $n = 50$ subintervals, and then successively double n. How large must n be for you to obtain the familiar four-place approximation $\pi \approx 3.1416$?

Using a Graphing Calculator

With a TI-83 calculator the **seq** command is used to form a list of terms to be summed, and then the **sum** command adds up their sum. Thus the commands shown in the screen



calculate the approximate midpoint sum 1.8861 in (4) using $n = 50$ subintervals; we write **D** for Δx. To proceed to the next step, we need only double the stored value of n and re-execute the three commands that are shown. With $n = 400$ we get the four-place accuracy shown in (3).



Using Maple

We first define the integrand function $f(x)$, enter the desired number n of subintervals, and calculate the resulting subinterval length $dx = \Delta x$:

```
f  := x->sqrt(x);
n  := 50;
dx := 2/n;
```

We can then calculate the corresponding midpoint sum in (4) with the command

```
evalf( sum(f((i-1/2)*dx),i=1..n)*dx );
```

1.886081621

where **evalf** converts the result to decimal form. For a further approximation we need only increase the number of subintervals and re-evaluate the last three commands.

```
n  := 400;
dx := 2/n;
evalf( sum(f((i-1/2)*dx),i=1..n)*dx );
```

1.885639245

Thus with $n = 400$ we get the four-place value 1.8856 shown in (3) above.

Using Mathematica

We first define the integrand function $f(x)$, enter the desired number n of subintervals, and calculate the resulting subinterval length $dx = \Delta x$:

```
f[x_] = Sqrt[x];
n = 50;
dx = 2/n;
```

We can then calculate the corresponding midpoint sum in (4) with the command

```
Sum[ f[(i-1/2)dx], {i,1,n} ] dx // N
```

1.88608

where the **// N** converts the result to decimal form. For a further approximation we need only increase the number of subintervals and re-evaluate the last three commands.

```
n = 400;
dx = 2/n;
```

```
Sum[ f[(i-1/2)dx], {i,1,n} ] dx // N
1.88564
```

Thus with $n = 400$ we get the four-place value 1.8856 shown in (3) above.

Using MATLAB

We first define the integrand function $f(x)$, enter the desired number n of subintervals, and calculate the resulting subinterval length $dx = \Delta x$:

```
f = inline('sqrt(x)');
n = 50;
dx = 2/n;
```

The command

```
x = dx/2 : dx : 2-dx/2;
```

then defines a list **x** of the midpoints from $m_1 = \frac{1}{2}\Delta x$ by increments of $\Delta x = dx$ to $m_n = 2 - \frac{1}{2}\Delta x$. We can now calculate the corresponding midpoint sum in (4) with the command

```
sum(f(x))*dx

ans =
    1.8861
```

that first calculates and then sums the list **f(x)** of function values $f(m_1)$ through $f(m_n)$, and finally multiplies the sum by Δx. For a further approximation we need only increase the number of subintervals and re-evaluate the last four commands.

```
n = 400;
dx = 2/n;
x = dx/2 : dx : 2-dx/2;
sum(f(x))*dx
ans =
    1.8856
```

Thus with $n = 400$ we get the four-place value 1.8856 shown in (3) above.