Using Riemann Sums to Estimate Areas, Volumes, and Lengths of Arc

Note: You may notice differences between this Maple worksheet and the equivalent Mathematica notebook. These differences were introduced to preserve the content of these modules and were necessary because of major functional differences between Maple and Mathematica.

Introduction

OBJECTIVE: Use Riemann sums to approximate areas, volumes, and lengths of arc. Construct accumulation functions and see how the accumulated quantities converge on the antiderivative in the limit.

In this module, we explore the use of Riemann sums to estimate areas, volumes of revolution, and the length of a curve. First, we write an expression for ΔQ_i , where *Q* is the quantity of interest (area, volume, or arc length), we look at $\frac{\Delta Q_i}{\Delta x}$ as the rate at which *Q* accumulates with respect to increments in *x*, we form a list of accumulated quantities, Q_i , and finally, we use the list of Q_i 's to reconstruct $\frac{\Delta Q_i}{\Delta x}$.

We create three new procedures in this module **areas( ), volumes( ), lengths( )** to help you explore the use of Reimann Sums with areas, volumes and arc lengths. Make sure you execute the special code before running the rest of the worksheet.

Technology Guidelines

NOTE: If you have just finished a worksheet, **restart** Maple before executing a new worksheet.
TO OPEN SECTIONS,
Click on the **PLUS** sign at the left hand side of the screen *or* select **Expand All Sections** from the **View** drop down menu.

TO STOP AN EXECUTION
Click on **STOP** button from the toolbar.

ORDER OF EXECUTION
Execute commands in the order given. Do not skip any Maple Input lines within a given worksheet

Alternatively, you can execute the entire worksheet by selecting the **Execute Worksheet** command from the **Edit** drop down menu.

SAVING WORKSHEETS.

You can save anytime to any directory you choose, and it is wise to save often.
EXPERIENCING MAJOR PROBLEMS
Save if appropriate, then shut down Maple and start it up

Special Code

The code that follows defines the three special functions that are used in this worksheet. You do not have to understand the code to do this worksheet, however, you will need to execute it in order to define the special functions that are used later. You can execute the code by placing the cursor anywhere in the next input cell and then pressing Enter. After you execute the following code, proceed to Part I of the worksheet.

```
> restart;
  with(plots):
  areas:=proc(a,b,n,fnc)

  local A, arealist, ratelist, Alist,  m, deltaA, msecant,
       p0, p1, p2, p4, h, k,ff,aa, eqn, varlist, c1,c2,f,i,ii;



  f:=unapply(fnc,x);

  h:=(b-a)/n:
  deltaA:=unapply(evalf(f(a+i*h))*h,i):
  print(`The` ,'Delta*A[i]', `function is `, deltaA(i));
  ratelist:= seq([a+ii*h, deltaA(ii)/h], ii=0..n-1):
  print(`The`, 'Delta(A[i])/(Delta(x))', `function is`, deltaA(i)/h);
  print('Delta*A[i]/Delta(x)');
  p0:=pointplot([ratelist], color=blue, symbol=cross, labels=[`x[i]`,""]):
  print(p0);

  A[total]:=unapply(sum(evalf(deltaA(j)), j=0..k-1),k):
  print(`The`, 'A[total](k)',`function is`,A[total](k));
  arealist:=seq([a+k*h, A[total](k)],k=0..n):
  print('A[total](k)');
  p1:=pointplot([arealist],color=red, labels=[`X[k]`,``]):
  print(p1);

  Alist:=[[seq(arealist[c1,1],c1=1..nops([arealist]))],[seq(arealist[c2,2],c2=1..nops([arealist]))]


  A[approx]:=x->expand(A[total]((x-a)/h));
```

```
print(`The function`, 'A[approx](x)',` is`, A[approx](x));
print('A[approx](x)',`&`,'A[total](k)');
p2:=plot(A[approx](x), x=a..a+n*h, color=blue, labels=[`x`,``]):
print(plots[display]([p1,p2],labels=["x",""]));

m[secant]:=i->(A[total](i+1)-A[total](i))/h:
print(`The function`, 'm[secant](i)', ` is `, expand(m[secant](i)));
msecant:=seq([a+k*h, m[secant](k)],k=0..n-1):
print('m[secant](i)');
p4:=pointplot([msecant], color=red, labels=[`x[i]`,``]);
print(p4);
print('Delta*A[i]/Delta(x)',`&`,'m[secant](i)');
display([p4,p0],labels=[`x[i]`,``]);

end:

volumes:=proc(a,b,n,fnc)

local volumesum, volumelist, ratelist, Vlist, V, msecant, deltaV,
    p0, p1, p2, p4, h, k,ff,aa, eqn, varlist, c1,c2, f, i,ii,m;


f:=unapply(fnc,x);

h:=(b-a)/n:
deltaV:=unapply(Pi*f(a+i*h)^2*h,i):
print(`The`, 'Delta*V[i]', `function is `, deltaV(i));
ratelist:= seq([a+ii*h, deltaV(ii)/h],ii=0..n-1);
unassign('i'):
print(`The`, 'Delta*V[i]/(Delta(x))', `function is`, deltaV(i)/h);
print('Delta*V[i]/Delta(x)');
p0:=pointplot([ratelist], color=blue, symbol=cross, labels=["x[i]",""]):
print(p0);

V[total]:=unapply(sum(deltaV(j), j=0..k-1),k):
print(`The`, 'V[total](k)',`function is`,expand(V[total](k)));
volumelist:=seq([a+k*h, V[total](k)],k=0..n):
print('V[total](k)');
p1:=pointplot([volumelist],color=red, labels=["x[k]",""]):
print(p1);

Vlist:=[[seq(volumelist[c1,1],c1=1..nops([volumelist]))],[seq(volumelist[c2,2],c2=1..nops
([volumelist]))]]:

V[approx]:=x->V[total]((x-a)/h);

print(`The function`, 'V[approx](x)',` is`, expand(V[approx](x)));
print('V[approx](x)',`&`,'V[total](k)');
p2:=plot(V[approx](x), x=a..a+n*h, color=blue, labels=["x",""]):
print(plots[display](p1,p2));
```

```
m[secant]:=i->expand((V[total](i+1)-V[total](i))/h):
print(`The function`, 'm[secant](i)', ` is `, expand(m[secant](i)));
msecant:=seq([a+k*h, m[secant](k)],k=0..n-1):
print('m[secant](i)');
p4:=pointplot([msecant], color=red, labels=["x[i]",""]):
print(p4);
print('Delta*V[i]/Delta(x)',`&`,'m[secant](i)');
display([p4,p0],labels=["x[i]",""]);

end:

lengths:=proc(a,b,n,fnc)

local lengthsum, lengthlist, ratelist, Llist, Length, msecant, deltaL,
    p0, p1, p2, p4, h, k,ff,aa, eqn, varlist, c1,c2,f,i,m;


f:=unapply(fnc,x);

h:=(b-a)/n:
deltaL:=unapply(sqrt(1+((f(a+i*h+h)-f(a+i*h))/h)^2)*h,i):
print(`The`, 'Delta*L[i]', `function is `, deltaL(i));
ratelist:= seq([a+i*h, deltaL(i)/h],i=0..n-1);
print(`The`, 'Delta*L[i]/(Delta(x))', `function is`, deltaL(i)/h);
print('Delta*L[i]/Delta(x)');
p0:=pointplot([ratelist], color=blue, symbol=cross, labels=["x[i]",""]):
print(p0);

unassign('i');
Length[total]:=unapply(sum(evalf(deltaL(i)), i=0..k-1),k):
print(`The`, 'Length[total](k)',`function is`,Length[total](k));
lengthlist:=seq([a+k*h, Length[total](k)],k=0..n):
print('Length[total](k)');
p1:=pointplot([lengthlist],color=red, labels=["x[k]",""]):
print(p1);

Llist:=[[seq(lengthlist[c1,1],c1=1..nops([lengthlist]))],[seq(lengthlist[c2,2],c2=1..nops
([lengthlist]))]]:

Length[approx]:=x->Length[total]((x-a)/h);

print(`The function`, 'Length[exact](x)',` is`, Length[approx](x));

print('Length[total](k)',`&`,'Length[approx](x)');
p2:=plot(Length[approx](x), x=a..a+n*h, color=blue, labels=["x",""]):
print(plots[display](p1,p2));

m[secant]:=i->(Length[total](i+1)-Length[total](i))/h:
print(`The function`, 'm[secant](i)', ` is `, m[secant](i));
msecant:=seq([a+k*h, m[secant](k)],k=0..n-1):print('m[secant](i)');
p4:=pointplot([msecant], color=red, labels=["x[i]",""]):
```

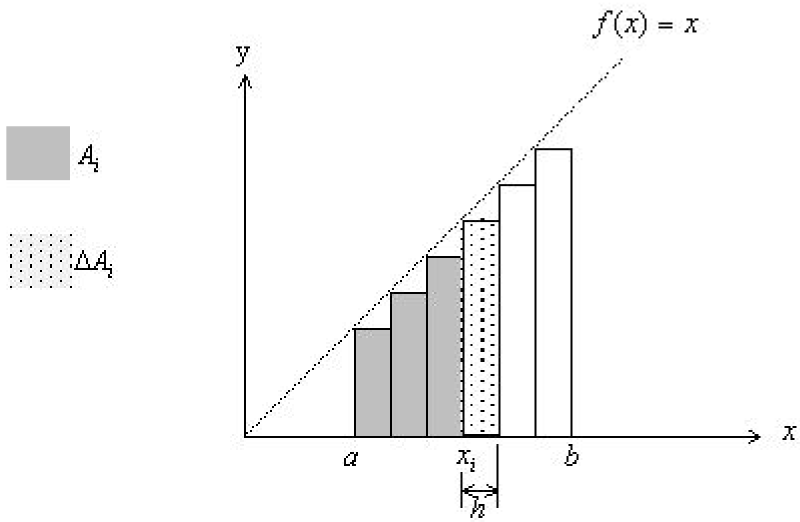
```
print(p4);
print('Delta*L[i]/Delta(x)',`&`,'m[secant](i)');
display([p4,p0],labels=["x[i]",""]);

end:
```

Warning, the name changecoords has been redefined

Part I: Areas

Chapter 5, Sections 3 and 4



Note: In this exercise, ΔA_i represents the area of the i^{th} rectangle, as shown in the figure above.

In this Part, we will do the following:

1. Form a function, ΔA_i , that gives the area of the i^{th} rectangle in terms of $f(x_i)$ and h.

2. Form a rate of change function, $\frac{\Delta A_i}{\Delta x}$, that gives the rate of accumulation of area at $x_i = a + ih$, for $i = 0, 2, 3, \ldots, n-1$, where n is the number of rectangles between $x = a$ and $x = b$.

3. Form a function, $A_{total}(k)$, that gives the sum of the areas of the rectangles, that is, the

ΔA_i 's, as i ranges from 0 to *k*. This approximates the area under the graph of *y* = *f*(*x*), between *x* = *a* and *x* = *a* + *kh*, for *k* = 0, 2, 3, . . ., $n-1$, where *n* is the number of rectangles between *x* = *a* and *x* = *b*.

4. Form a function that gives the slopes of the secant lines on the graph of the $A_{total}(k)$ function found in step 3, and show that this is the same as the rate of change function found in step 2.

Our first objective is to write ΔA_i , the area of the i^{th} rectangle, in terms of *f*(x_i) and *h* where x_i = *a* + *ih*. At first, we let *a* = -5, *b* = 10, and *n* = 10.

```
> f:=x->x:
  a:=-5:
  b:=10:
  n:=10:
  h:=(b-a)/n:
  Delta_A:=i->evalf(f(a+i*h)*h):
  print(`The`, 'Delta*A[i]', `function is`, Delta_A(i));
```

$$The, \Delta A_i, function\ is, -7.500000000 + 2.250000000\, i$$

You should note that if *f*(x_i) is negative and $\Delta x = h$ is positive, then ΔA_i is negative. The areas can be positive, negative, or zero, and so we refer to them as "signed areas". Areas above the *x*-axis are positive and areas below the *x*-axis are negative, provided $\Delta x = h$ is positive.

Next we form the rate-of-change function, $\frac{\Delta A_i}{\Delta x}$. The function $\frac{\Delta A_i}{\Delta x}$ is the rate at which the total signed area of the rectangles accumulates with each increment of $\Delta x = h$ that is added to *x*. We generate a list of the rate-of-change values for each x_i = *a* + *ih*, and plot the rate-of-change versus x_i , assigning the graph to the symbol name **p0** for later reference.

```
> print(`The rate of accumulation of area function`, 'Delta*A[i]/Delta(x)',` is `, Delta_A(i)/h);
  print('i',`        `,'x[i]=a+i*h',`       `,'Delta*A[i]/Delta(x)');
  ratelist_print:=evalf([seq([i,a+i*h,Delta_A(i)/h],i=0..n-1)]):
  matrix(ratelist_print);
```

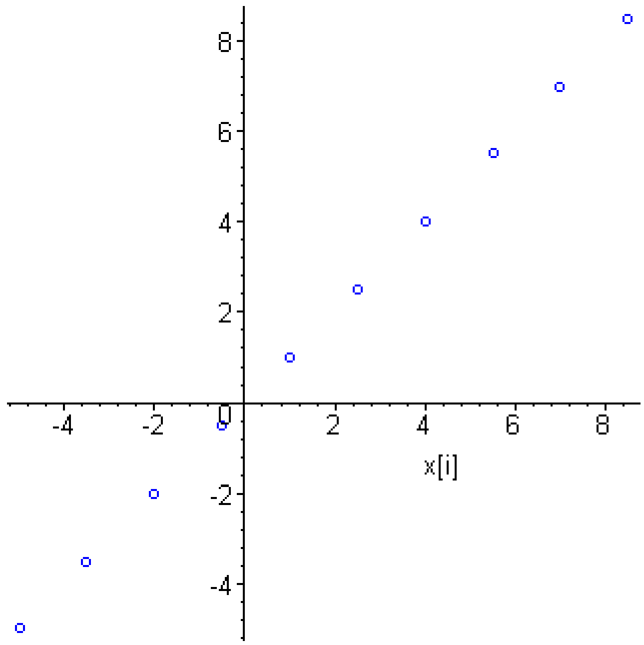
The rate of accumulation of area function, $\frac{\Delta A_i}{\Delta(x)}$, *is* , $-5.000000000+1.500000000\,i$

$i,\qquad ,\ x_i = a + i\,h,\qquad ,\ \frac{\Delta A_i}{\Delta(x)}$

$$\begin{bmatrix} 0. & -5. & -5.000000000 \\ 1. & -3.500000000 & -3.500000000 \\ 2. & -2. & -2.000000000 \\ 3. & -0.5000000000 & -0.5000000000 \\ 4. & 1. & 1.000000000 \\ 5. & 2.500000000 & 2.500000000 \\ 6. & 4. & 4.000000000 \\ 7. & 5.500000000 & 5.500000000 \\ 8. & 7. & 7.000000000 \\ 9. & 8.500000000 & 8.500000000 \end{bmatrix}$$

```
> ratelist_plot:=evalf([seq([a+i*h,Delta_A(i)/h],i=0..n-1)]):
  print('Delta*A[i]/Delta(x)');
  p0:=plots[pointplot](ratelist_plot,color=blue,symbol=circle,
  labels=["x[i]",""]):
  plots[display](p0);
```

$$\frac{\Delta A_i}{\Delta(x)}$$



You have probably noticed that, for the area problem, the rate at which the total area of the rectangles accumulates with each added increment *h* in *x* is simply $f(x_i)$, the height of the i^{th} rectangle. That is, $\frac{\Delta A(i)}{\Delta x} = f(x_i)$.

Now we form a function that calculates the left Riemann sum of the ΔA_i 's for the first *k* rectangles, where *k* can range between 0 and *n*.

```
> A[total]:=k->sum(Delta_A(i),i=0..k-1);
  print(`The`, 'A[total](k)', `function is`, A[total](k));
```

$$A_{total} := k \to \sum_{i=0}^{k-1} \text{Delta\_A}(i)$$

$$\mathit{The}, A_{total}(k), \mathit{function\ is}, -8.625000000\, k + 1.125000000\, k^2$$

For example, we can use $A_{total}(4)$ to calculate the total signed area of the first 4 rectangles.

```
> A[total](4);
```

$$-16.50000000$$

Or, we can calculate the area of all 10 rectangles. (Recall that we set n = 10.)

```
> A[total](10);
```

26.25000000

Next we form a list of the ordered pairs, where the first element of each ordered pair is the x-coordinate of the left side of the last rectangle in the sum and the second element in each ordered pair is the area of the first k rectangles.

```
> arealist_print:=evalf([seq([kk,a+kk*h,A[total](kk)],kk=0..n)]):
  print('k',`      `,'x[i]=a+k*h',`   `,'A[total](k)');
  matrix(arealist_print);
```

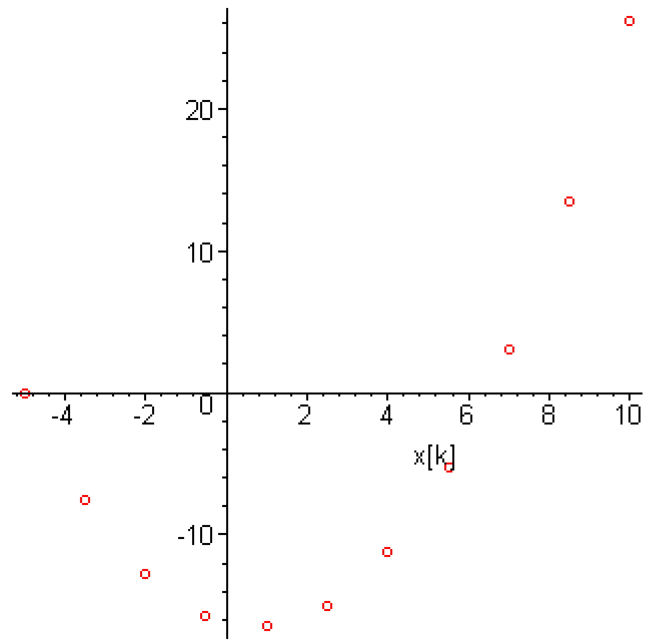
$k, \quad , x_i = a + k\,h, \quad , A_{total}(k)$

0.	-5.	0.
1.	-3.500000000	-7.500000000
2.	-2.	-12.75000000
3.	-0.5000000000	-15.75000000
4.	1.	-16.50000000
5.	2.500000000	-15.
6.	4.	-11.25000000
7.	5.500000000	-5.250000000
8.	7.	3.
9.	8.500000000	13.50000000
10.	10.	26.25000000

We plot the list of points and assign it to the symbol name, **p1,** for later use.

```
> arealist_plot:=evalf([seq([a+kk*h,A[total](kk)],kk=0..n)]):
  print('A[total](k)');
  p1:=plots[pointplot](arealist_plot,color=red,
  labels=["x[k]",""],symbol=circle):
  plots[display](p1);
```

$A_{total}(k)$



The function $A_{total}(k)$ is a quadratic polynomial in k, so now we express $A_{total}(k)$ as a function of x, recalling that $k = \frac{x-a}{h}$. We call the new area function $A_{approx}(x)$ since it approximates the area under the graph of $f(x)$.

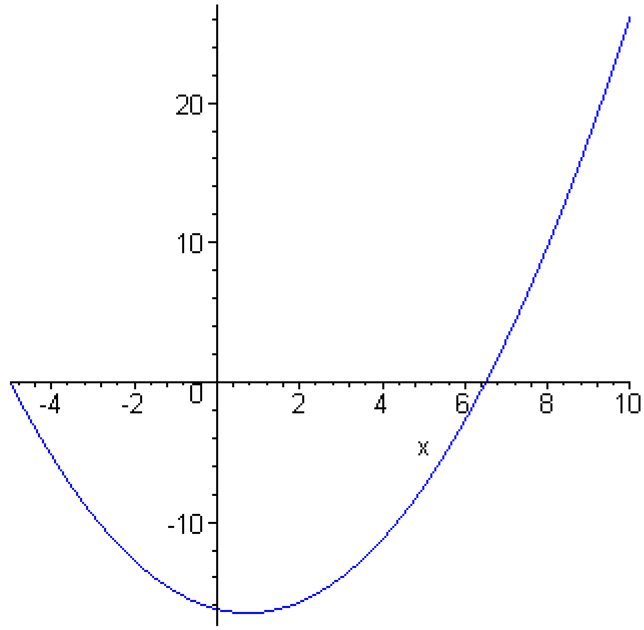
```
> A[approx]:=x->A[total]((x-a)/h):
  print(`The function`, 'A[approx](x)',` is `, expand(A[approx](x)));
```

$$\text{The function}, A_{approx}(x), \text{ is }, -0.750000000\,x - 16.25000000 + 0.5000000001\,x^2$$

Next we plot $A_{approx}(x)$ and the points in **arealist** to show that the points do in fact lie on the graph of the quadratic polynomial. First, we plot $A_{approx}(x)$ and then show the two graphs on the same plot.

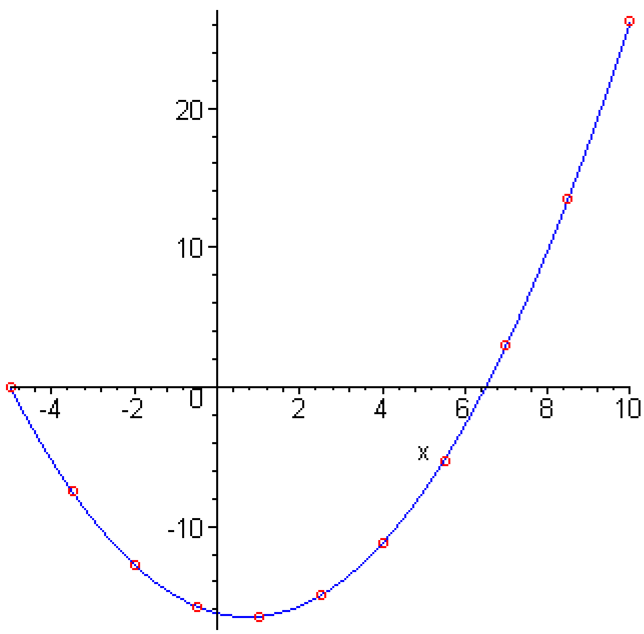
```
> print('A[approx](x)');
  p2:=plot(A[approx](x),x=a..a+n*h,labels=["x",""], color=blue):
  plots[display](p2);
```

$$A_{approx}(x)$$




```
> print('A[approx](x)',`and`,'A[total](k)');
  plots[display](p1,p2,labels=["x",""]);
```

$A_{approx}(x)$, **proc**() **option** *builtin*; 166 **end proc**, $A_{total}(k)$



The slopes of the secant lines taken from the graph of $A_{total}(k)$ above give the rate at which the total area of the rectangles accumulates with each *h* increment of *x*. Now let's calculate the slopes of the secant lines between consecutive pairs of points on the graph above and plot them.

```
> m[secant]:=i->(A[total](i+1)-A[total](i))/h;
  print(`The secant slopes function`, 'm[secant](i)', ` is `, expand(m[secant](i)));
```

```
msecantlist_print:=evalf([seq([k,a+k*h,m[secant](k)],k=0..n-1)]):
print('i',`     `,'x[i]=a+ih',`     `,'m[secant](i)');
matrix(msecantlist_print);
```

$$m_{secant} := i \to \frac{A_{total}(i+1) - A_{total}(i)}{h}$$

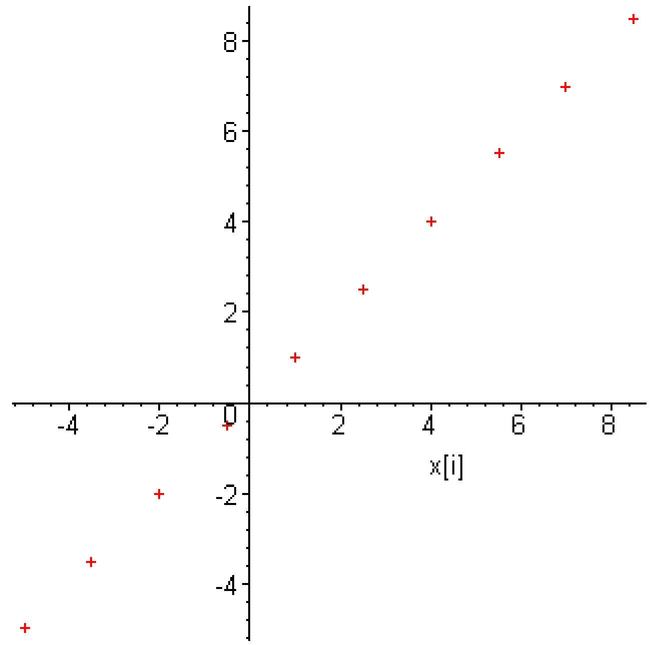
The secant slopes function, $m_{secant}(i)$, *is* , $-5.000000000 + 1.500000000\, i$

$$i, \quad , x_i = a + ih, \quad , m_{secant}(i)$$

0.	-5.	-5.000000000
1.	-3.500000000	-3.500000000
2.	-2.	-2.000000000
3.	-0.5000000000	-0.5000000000
4.	1.	1.000000000
5.	2.500000000	2.500000000
6.	4.	4.000000000
7.	5.500000000	5.500000000
8.	7.	7.000000000
9.	8.500000000	8.500000000

```
> msecantlist_plot:=evalf([seq([a+k*h,m[secant](k)],k=0..n-1)]):
print('m[secant](i)');
p4:=plots[pointplot](msecantlist_plot,color=red,labels=["x[i]",""],
symbol=cross):
plots[display](p4);
```

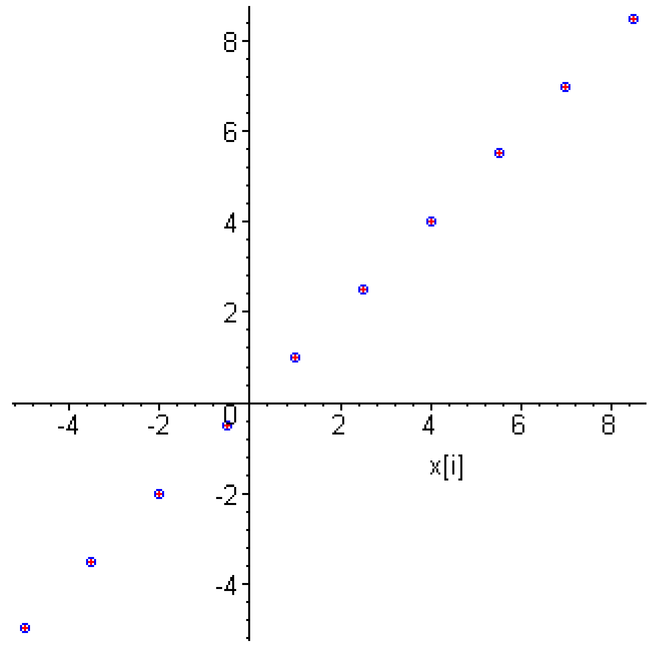
$$m_{secant}(i)$$



But this is the same as the $\frac{\Delta A_i}{\Delta x}$ that we calculated at the beginning of this exercise. To confirm this we show **msecantslist** and $\frac{\Delta A_i}{\Delta x}$ together on the same graph.

```
> print('Delta*A[i]/Delta(x)',`&`,'m[secant](i)');
  plots[display]({p0,p4},labels=["x[i]",""]);
```

$$\frac{\Delta A_i}{\Delta(x)}, \&, m_{secant}(i)$$



It looks as though we have come full circle. In the "You Try It" exercise that follows, you are asked to explore what happens as *n*, the number of rectangles, goes to infinity and as $h \to 0$.

You Try It: Part I - Taking It to the Limit and Other Functions

Chapter 5, Sections 3 and 4

To help you with the exercises that follow, we have copied all of the commands from Part I into a single procedure called **areas( )**. In each exercise, you are asked to change some of the entries in the commands at the bottom of this section and then answer the questions. (Note that we have hidden all but the input commands and put them in the **areas( )** procedure. In addition, we suppress displaying the lists of values that are generated because they become too large when *n* is a large number.)

1. For the function *f*(*x*) considered in Part I, increase the number of rectangles, *n*, so that the total signed area of the rectangles approaches the total signed area under the graph of *f*(*x*). Try *n* = 25, 50, 100, 250, 500. For larger values of *n*, the evaluation of the commands may take a while because of the large number of ΔA_i 's that must be added to form each value of $A_{total}(k)$.

(There are more efficient ways to calculate these sums, and these are investigated in the module, "**Riemann, Trapezoids, and Simpson**".) As *n* gets larger, what function is $A_{approx}(x)$ approaching? Also, as *n* gets larger, $\Delta x = h$ gets closer to 0. What is the

$\lim_{\Delta x \to 0} \frac{\Delta A_i}{\Delta x}$? In the limit as $n \to \infty$ and $h \to 0$, what is the relationship between

$\frac{\Delta A_i}{\Delta x}$ and $A_{approx}(x)$?

2. Repeat Exercise 1 for the following: $f(x) = x^2$, $a = -5$, and $b = 5$.

3. Repeat Exercise 1 for the following: $f(x) = \cos(x)$, $a = 0$, and $b = 2\pi$.

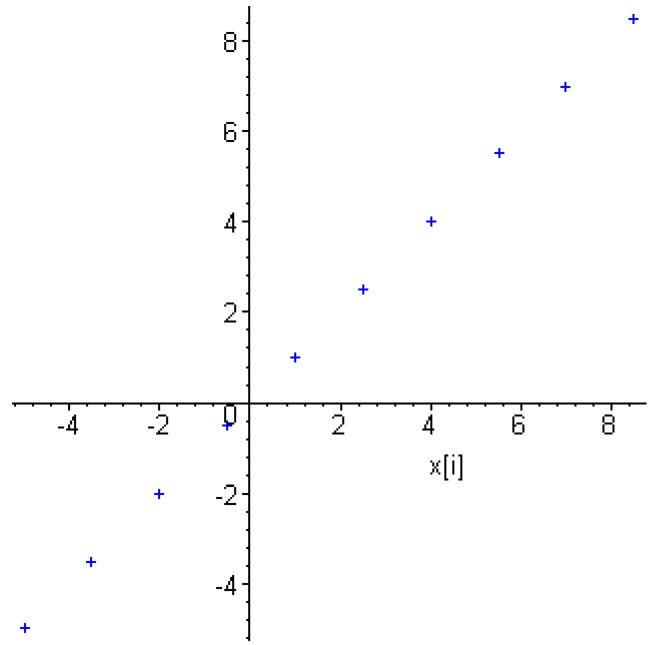
4. Repeat Exercise 1 for the following: $f(x) = e^x$, $a = -1$, and $b = 1$.

```
> f:=x:
  a:=-5:
  b:=10:
  n:=10:
  areas(a,b,n,f);
```

$$\mathit{The}, \Delta A_i, \mathit{function\ is}, -7.500000000 + 2.250000000\, i$$

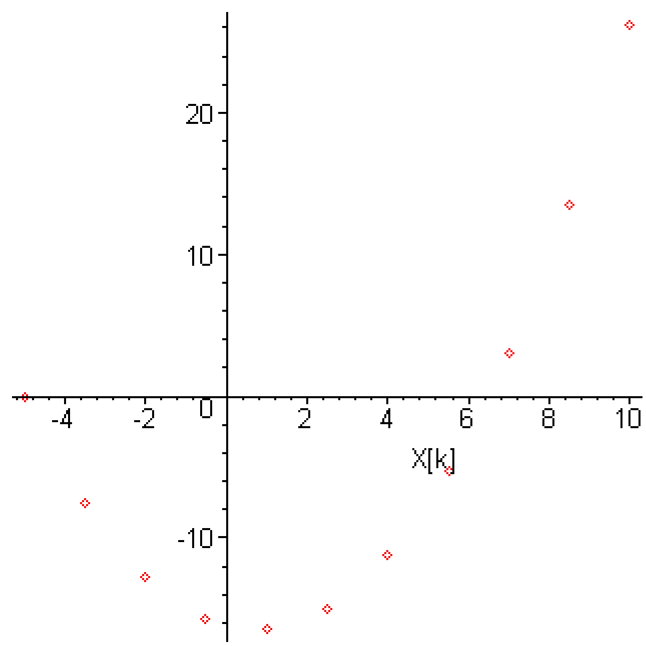
$$\mathit{The}, \frac{\Delta(A_i)}{\Delta(x)}, \mathit{function\ is}, -5.000000000 + 1.500000000\, i$$

$$\frac{\Delta A_i}{\Delta(x)}$$



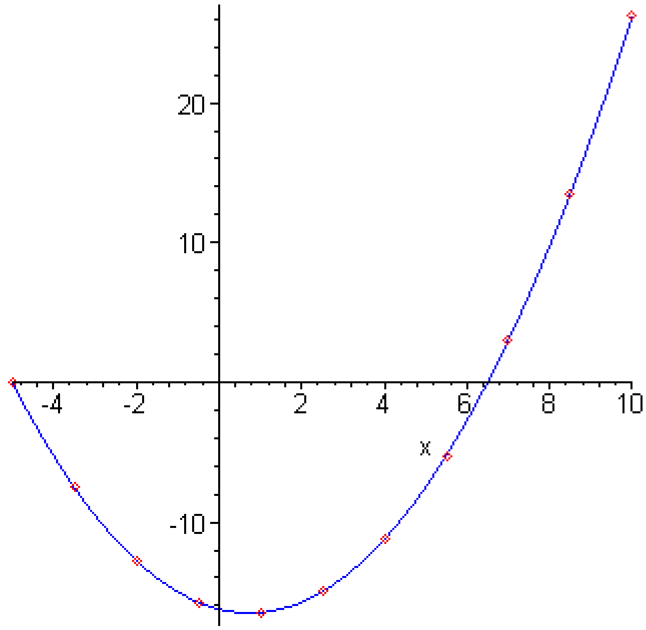
The, $A_{total}(k)$, *function is*, $-8.625000000\,k + 1.125000000\,k^2$

$A_{total}(k)$



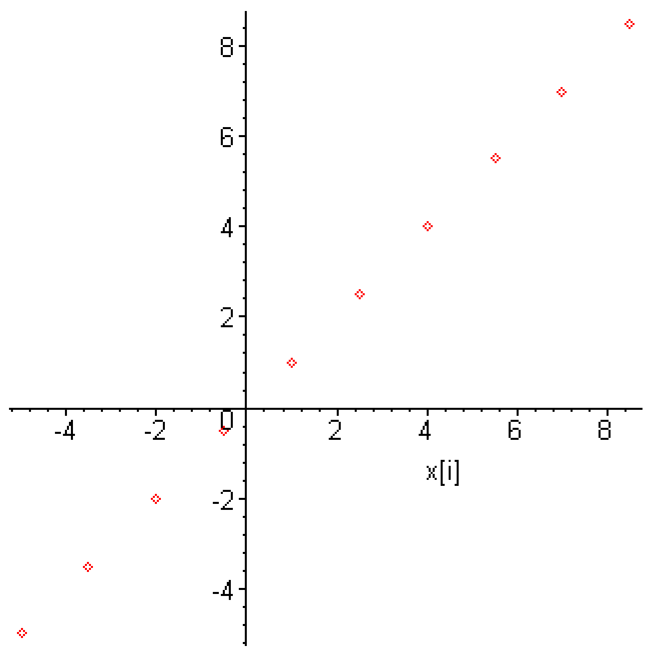
The function, $A_{approx}(x)$, *is*, $-0.750000000\,x - 16.25000000 + 0.5000000000\,x^2$

$A_{approx}(x)$, &, $A_{total}(k)$

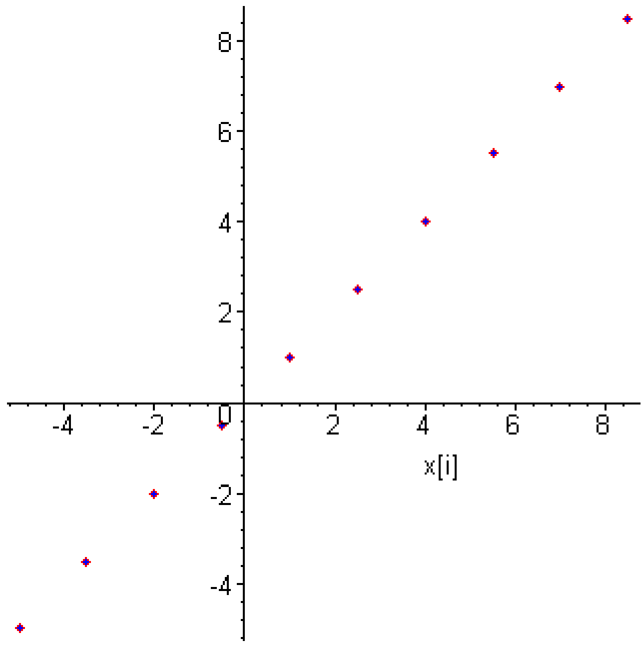


The function, $m_{secant}(i)$*, is* , $-5.000000000 + 1.500000000\, i$

$m_{secant}(i)$

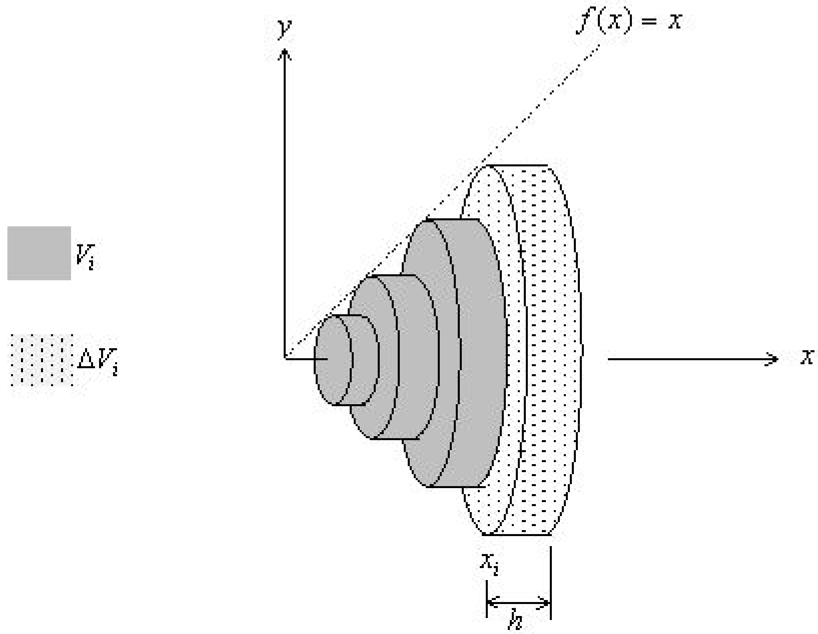


$\frac{\Delta A_i}{\Delta(x)}$, &, $m_{secant}(i)$



Part II: Volumes

Chapter 6, Sections 1 and 2



Note: In this exercise, ΔV_i represents the volume of the i^{th} disk, as shown in the figure above.

Our first objective is to form a function $V_{total}(k)$, that estimates the volume of the solid that is generated when the region bounded by the graph of a function *f*(*x*), the *x*-axis, and the vertical lines $x = a$ and $x = x_k = a + k\,h$ is rotated about the *x*-axis. We do this by adding the ΔV_i 's, the volumes of the first *k* disks, where *k* can range from 0 up to *n*, the number of disks of thickness *h* between *x* = *a* and *x* = *b*.

For the function *f*(*x*) = *x*, we write ΔV_i , the volume of the i^{th} disk, in terms of $f(x_i)$ and *h* where $x_i = a + ih$. At first, we let $a = -5$, *b* = 5, and *n* =10.

```
> f:=x->x:
  a:=-5:
  b:=10:
  n:=10:
  h:=(b-a)/n:
  Delta_V:=i->Pi*(f(a+i*h))^2*h:
  print(`The`, 'Delta*V[i]', `function is`, Delta_V(i));
```

$$The,\ \Delta V_i,\ function\ is,\ \frac{3\,\pi\left(-5+\frac{3\,i}{2}\right)^2}{2}$$

The function $\frac{\Delta V_i}{\Delta x}$ is the rate at which the total volume of the solid accumulates with each increment of *h* added to *x*. We generate a list of the rate-of-change values for each $x_i = a + ih$, plot the rate-of-change versus x_i , and then assign the graph to the symbol name **p0** for later reference.

```
> print(`The rate of accumulation of volume function`, 'Delta*V[i]/Delta(x)',` is `, Delta_V(i)/
  print('i',`      `,'x[i]=a+i*h',`      `,'Delta*V[i]/Delta(x)');
  ratelist_print:=evalf([seq([i,a+i*h,Delta_V(i)/h],i=0..n-1)]):
  matrix(ratelist_print);
```

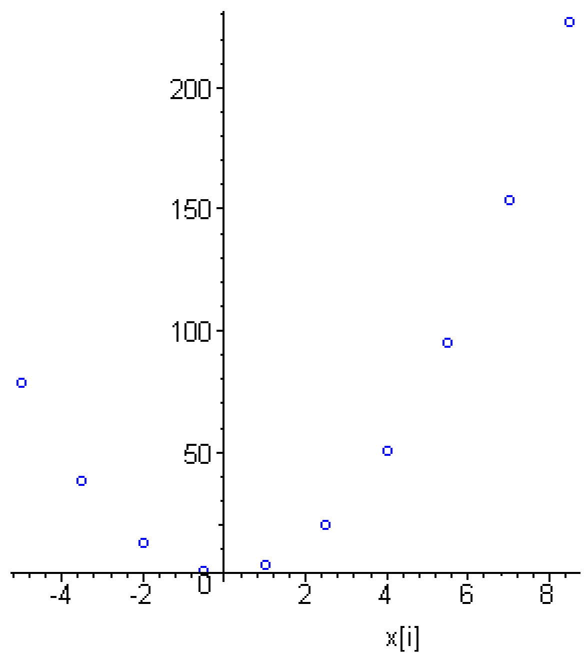
$$The\ rate\ of\ accumulation\ of\ volume\ function,\ \frac{\Delta V_i}{\Delta(x)},\ is\ ,\ \pi\left(-5+\frac{3\,i}{2}\right)^2$$

$$i,\qquad ,\ x_i = a + i\,h,\qquad ,\ \frac{\Delta V_i}{\Delta(x)}$$

0.	-5.	78.53981635
1.	-3.500000000	38.48451001
2.	-2.	12.56637062
3.	-0.5000000000	0.7853981635
4.	1.	3.141592654
5.	2.500000000	19.63495409
6.	4.	50.26548246
7.	5.500000000	95.03317778
8.	7.	153.9380400
9.	8.500000000	226.9800693

```
> ratelist_plot:=evalf([seq([a+i*h,Delta_V(i)/h],i=0..n-1)]):
  print('Delta*V[i]/Delta(x)');
  p0:=plots[pointplot](ratelist_plot,color=blue,
  labels=["x[i]",""], symbol=circle):
  plots[display](p0);
```

$$\frac{\Delta V_i}{\Delta(x)}$$



You have probably noticed that for the volume problem the rate at which the total volume of the disks accumulates with each added increment h is $\pi f(x_i)^2$, the area of the circular end of the

i^{th} disk. That is, $\frac{\Delta V_i}{\Delta x} = \pi f(x_i)^2$.

Now we form a function that calculates the left Riemann sum of the ΔV_i 's for the first *k* disks where *k* can range between 0 and *n*.

```
> V[total]:=k->sum(Delta_V(i),i=0..k-1):
  print(`The`, 'V[total](k)', `function is`, V[total](k));
```

$$The, V_{total}(k), function\ is, \frac{789}{16}\pi k - \frac{207}{16}\pi k^2 + \frac{9}{8}\pi k^3$$

We form a list of the ordered pairs where the first element of each ordered pair is the *x*-coordinate of the left side of the last disk in the sum and the second element in each ordered pair is the volume of the first *k* disks.

```
> volumelist_print:=evalf([seq([kk,a+kk*h,V[total](kk)],kk=0..n)]):
  print('k',`    `,'x[k]=a+k*h',`   `,'V[total](k)');
  matrix(volumelist_print);
```

$$k, \quad , x_k = a + k\,h, \quad , V_{total}(k)$$

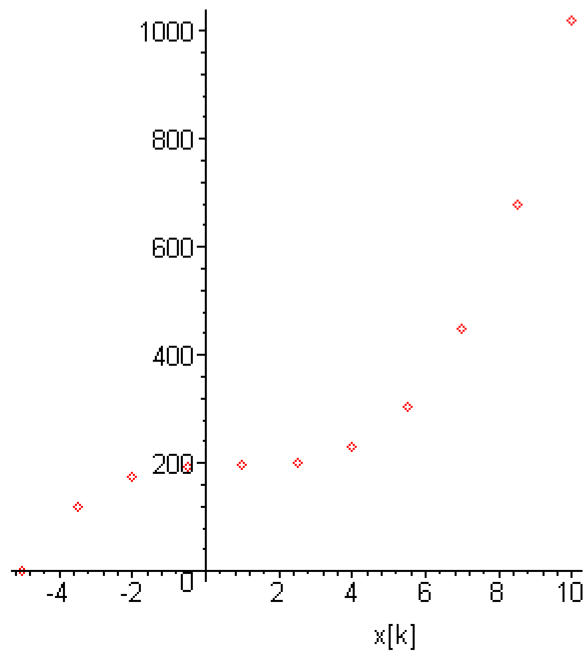
0.	-5.	0.
1.	-3.500000000	117.8097245
2.	-2.	175.5364895
3.	-0.5000000000	194.3860455
4.	1.	195.5641427
5.	2.500000000	200.2765317
6.	4.	229.7289628
7.	5.500000000	305.1271865
8.	7.	447.6769532
9.	8.500000000	678.5840133
10.	10.	1019.054117

Let's plot the list of points and assign it to the symbol name, **p1**, for later use.

```
> volumelist_plot:=evalf([seq([a+kk*h,V[total](kk)],kk=0..n)]):
  print('V[total](k)');
```

```
p1:=plots[pointplot](volumelist_plot,color=red,labels=["x[k]",""]):
plots[display](p1);
```

$V_{total}(k)$



The function $V_{total}(k)$ is a cubic polynomial in *k*, so now we express $V_{total}(k)$ as a function of *x* recalling that $k = \frac{x - a}{h}$. We call the new volume function $V_{approx}(x)$ since it approximates the volume obtained by revolving the area under the graph of *f*(*x*).

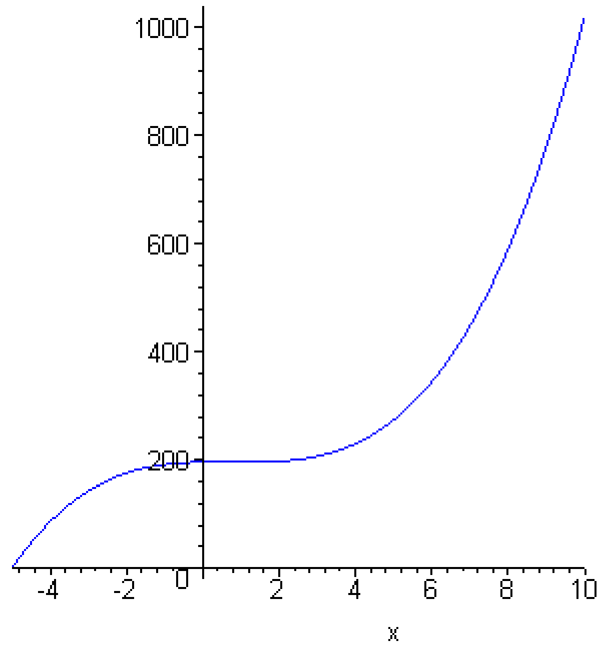
```
> V[approx]:=x->V[total]((x-a)/h):
  print(`The function`, 'V[approx](x)', ` is `, expand(V[approx](x)));
```

$$\text{The function, } V_{approx}(x), \text{ is }, \frac{3}{8}\pi x + \frac{1495}{24}\pi - \frac{3}{4}\pi x^2 + \frac{1}{3}\pi x^3$$

Now we plot $V_{approx}(x)$ and the points in **volumelist** to show that the points do in fact lie on the graph of the cubic polynomial. First, we plot $V_{approx}(x)$ and then show the two graphs on the same plot.

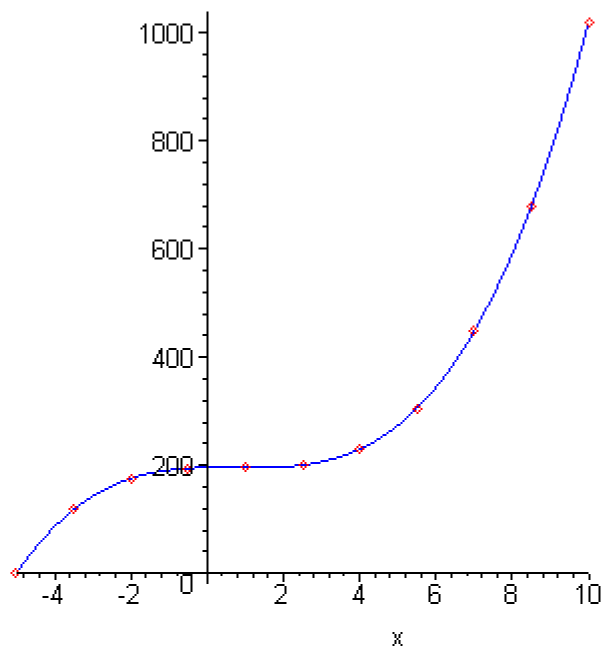
```
> print('V[approx](x)');
  p2:=plot(V[approx](x),x=a..a+n*h,labels=["x",""], color=blue):
  plots[display](p2);
```

$V_{approx}(x)$




```
> print('V[approx](x)',`&`,'V[total](k)');
  plots[display](p1,p2,labels=["x",""]);
```

$$V_{approx}(x), \&, V_{total}(k)$$



The slopes of the secant lines taken from the $V_{total}(k)$ graph give the rate at which the total area of the rectangles accumulates with each *h* increment of *x*. Now let's calculate the slopes of the secant lines between consecutive pairs of points on the graph above and plot them.

```
> m[secant]:=i->(V[total](i+1)-V[total](i))/h;
```

```
print(`The secant slopes function`, 'm[secant](i)', ` is `, expand(m[secant](i)));
msecantlist_print:=evalf([seq([k,a+k*h,m[secant](k)],k=0..n-1)]):
print('i',`     `,'x[i]=a+ih',`     `,'m[secant](i)');
matrix(msecantlist_print);
```

$$m_{secant} := i \to \frac{V_{total}(i+1) - V_{total}(i)}{h}$$

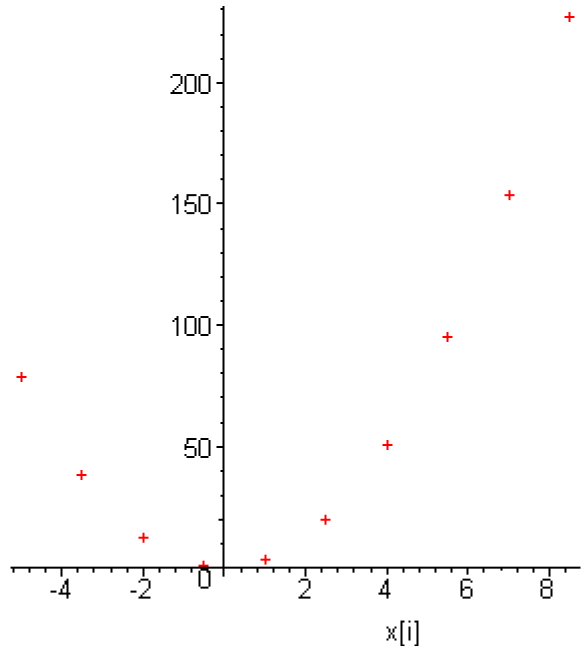
$$The\ secant\ slopes\ function,\ m_{secant}(i),\ is\ ,\ -15\,\pi\, i + 25\,\pi + \frac{9}{4}\pi\, i^2$$

$$i,\quad ,\ x_i = a + ih,\quad ,\ m_{secant}(i)$$

0.	-5.	78.53981635
1.	-3.500000000	38.48451001
2.	-2.	12.56637062
3.	-0.5000000000	0.7853981635
4.	1.	3.141592654
5.	2.500000000	19.63495409
6.	4.	50.26548246
7.	5.500000000	95.03317778
8.	7.	153.9380400
9.	8.500000000	226.9800693

```
> msecantlist_plot:=evalf([seq([a+k*h,m[secant](k)],k=0..n-1)]):
print('m[secant](i)');
p4:=plots[pointplot](msecantlist_plot,color=red,labels=["x[i]",""],symbol=cross):
plots[display](p4);
```

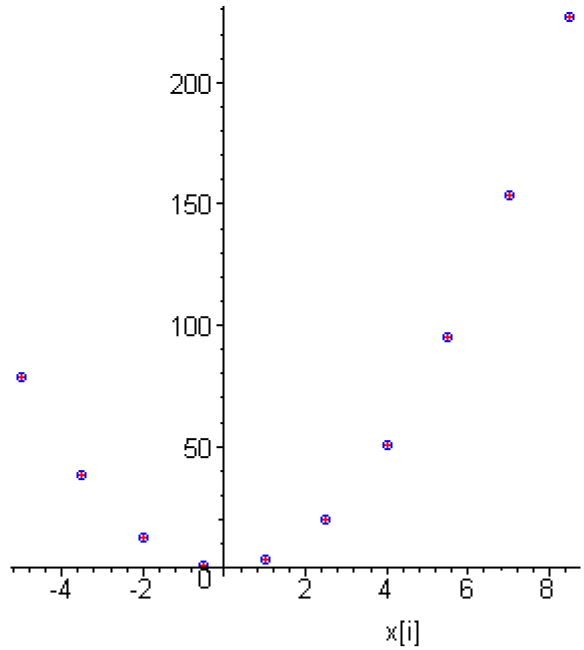
$$m_{secant}(i)$$



But this is the same as the $\frac{\Delta V_i}{\Delta x}$ that we calculated at the beginning of this exercise. To confirm this we show **msecants** and $\frac{\Delta V_i}{\Delta x}$ together on the same graph.

```
> print('Delta*V[i]/Delta(x)',`&`,'m[secant](i)');
  plots[display](p0,p4,labels=["x[i]",""]);
```

$$\frac{\Delta V_i}{\Delta(x)}, \&, m_{secant}(i)$$



It looks as though we have come full circle again.

You Try It: Part II - Taking It to the Limit and Another Function

Chapter 6, Sections 1 and 2

To help you with the exercises that follow, we have copied all of the commands from Part II into a procedure called **volumes( )**. In each exercise, you are asked to change some of the entries in the commands at the bottom of this section and then answer the questions. (Note that we have hidden all but the input commands and put them in the **volumes( )** procedure. In addition, we suppress displaying the lists of values that are generated because they become too large when *n* is a large number.)

1. For the function $f(x) = x$ considered in Part II, increase *n*, the number of disks between $x = a$ and $x = b$, so that the total volume of the disks approaches the volume of revolution. Try $n =$ 50, 100, 250, 500. For larger values of *n*, the evaluation of the commands may take a while because of the large number of ΔV_i

's that must be added to form each value of $V_{total}(k)$. As *n* gets larger, what function does $V_{approx}(x)$ appear to be approaching? Also, as *n* gets larger, $\Delta x = h$ gets closer to 0. What is the $\lim_{\Delta x \to 0} \frac{\Delta V_i}{\Delta x}$? In the limit as $n \to \infty$ and $h \to 0$, what is the relationship between

$\frac{\Delta V_i}{\Delta x}$ and $V_{approx}(x)$?

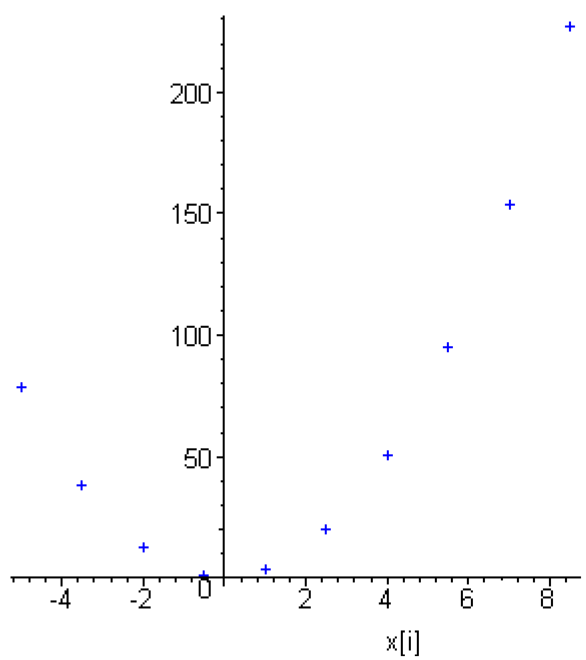
2. Repeat Exercise 1 for the following: $f(x) = \sin x$, $a = 0$, and $b = 2\pi$. Explain the stair step pattern in the graph of $V_{approx}(x)$.

```
> f:=x:
  a:=-5:
  b:=10:
  n:=10:
  volumes(a,b,n,f);
```

$$The,\ \Delta V_i,\ function\ is\ ,\ \frac{3\pi\left(-5+\frac{3i}{2}\right)^2}{2}$$

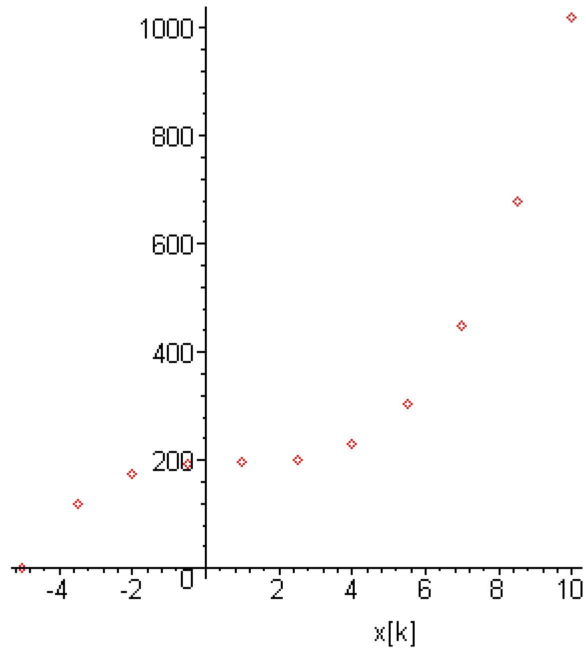
$$The,\ \frac{\Delta V_i}{\Delta(x)},\ function\ is,\ \pi\left(-5+\frac{3i}{2}\right)^2$$

$$\frac{\Delta V_i}{\Delta(x)}$$



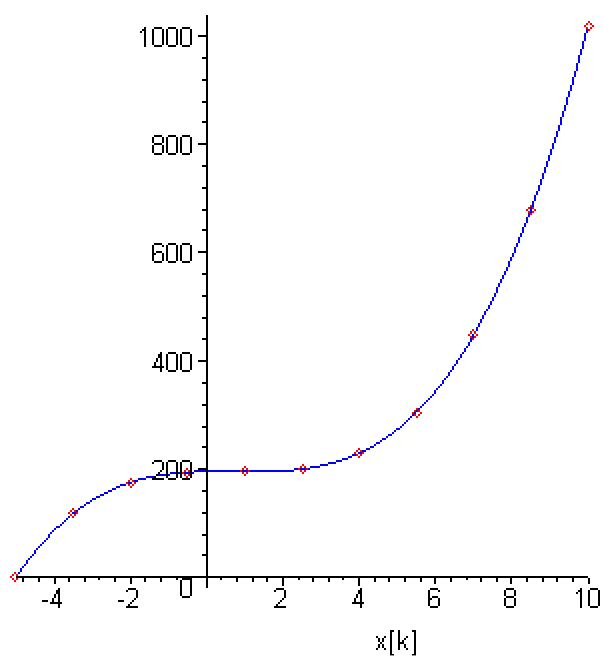
$$\textit{The, } V_{total}(k)\textit{, function is, } \frac{789}{16}\pi k - \frac{207}{16}\pi k^2 + \frac{9}{8}\pi k^3$$

$V_{total}(k)$



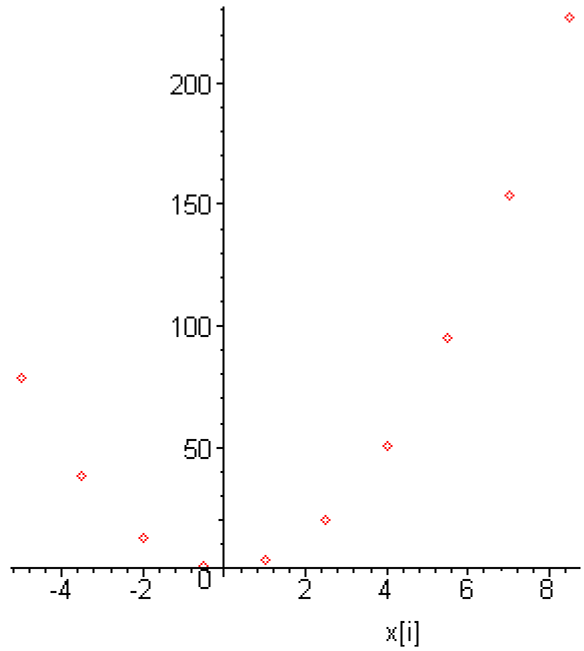
$$\textit{The function, } V_{approx}(x)\textit{, is, } \frac{3}{8}\pi x + \frac{1495}{24}\pi - \frac{3}{4}\pi x^2 + \frac{1}{3}\pi x^3$$

$V_{approx}(x)$, &, $V_{total}(k)$

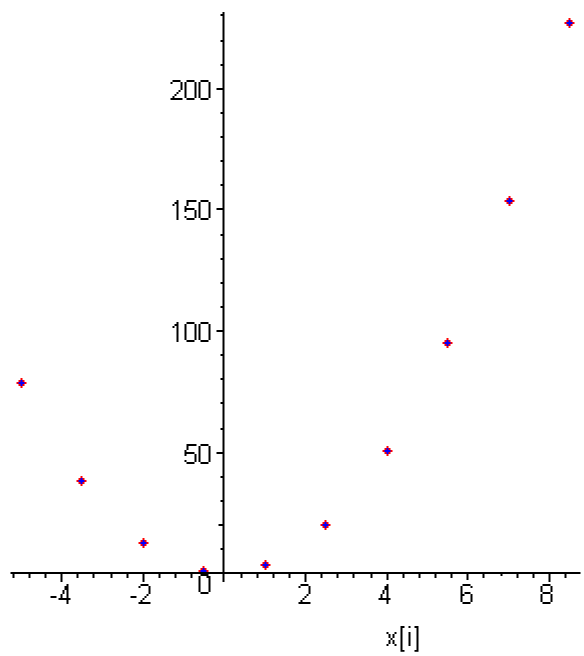


The function, $m_{secant}(i)$, *is* , $-15\,\pi\,i + 25\,\pi + \frac{9}{4}\pi\,i^2$

$m_{secant}(i)$

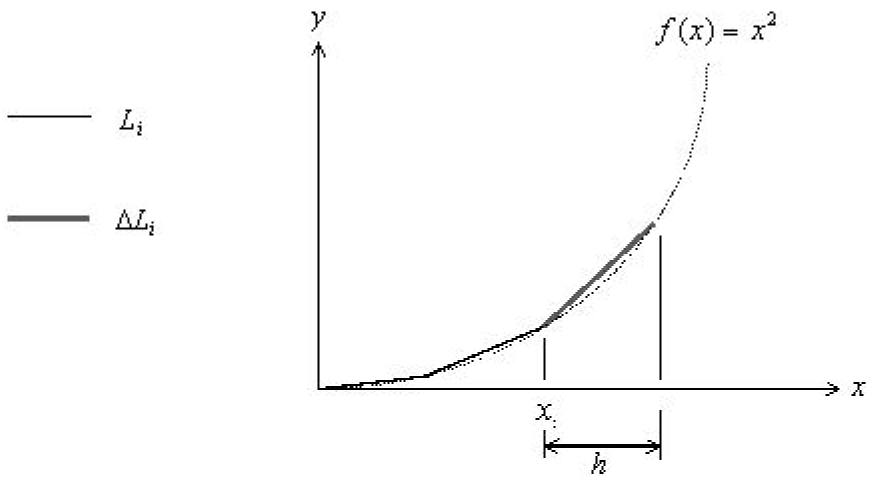


$\frac{\Delta V_i}{\Delta(x)}$, &, $m_{secant}(i)$



Part III: Lengths of Arc

Chapter 6, Section 3



Note: In this exercise, ΔL_i represents the length of the i^{th} line segment, as shown in the figure above.

Our first objective is to form a function $Length_{total}(k)$, that estimates the length of the graph of a function, $f(x)$, as x varies from $x = a$ and $x = x_k = a + kh$. We do this by adding the ΔL_i 's, the lengths of the first k line segments, where k can range from 0 up to n, the number of line segments between $x = a$ and $x = b$.

For the function $f(x) = x^2$, we need to write ΔL_i in terms of $f(x_i)$, $f(x_i + h)$, and h where $x_i = a + ih$. To do this, we use the Pythagorean theorem to calculate the length of the line segment that begins at the point (x_i , $f(x_i)$) and ends at the point ($x_i + h$, $f(x_i + h)$):

ΔL_i

$$= \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2} = \sqrt{h^2 + \left(\frac{\Delta y_i}{h}\right)^2 h^2} = h\sqrt{1 + \left(\frac{\Delta y_i}{h}\right)^2} =$$

$$h\sqrt{1+\left(\frac{f(x_i+h)-f(x_i)}{h}\right)^2}$$

Now let's put it into Maple. We let $a = -5$, $b = 5$ and $n = 10$.

```
> f:=x->x^2:
  a:=-5:
  b:=10:
  n:=10:
  h:=(b-a)/n:
  Delta_L:=i->h*sqrt(1+((f(a+i*h+h)-f(a+i*h))/h)^2):
  print(`The`, 'Delta*L[i]', `function is`, Delta_L(i));
```

$$The, \Delta L_i, function\ is, \frac{3\sqrt{293-204\,i+36\,i^2}}{4}$$

The function $\frac{\Delta L_i}{\Delta x}$ is the rate at which the total length of the line segments accumulates with each increment of *h* added to *x*. We generate a list of the rate-of-change values for each x_i = $a+ih$ and plot the rate-of-change versus x_i , assigning the graph to the symbol name **p0** for later reference.

```
> print(`The rate of accumulation of the length function`, 'Delta*L[i]/Delta(x)',` is `, Delta_L
  print('i',`       `,'x[i]=a+i*h',`       `,'Delta*L[i]/Delta(x)');
  ratelist_print:=evalf([seq([i,a+i*h,Delta_L(i)/h],i=0..n-1)]):
  matrix(ratelist_print);
```

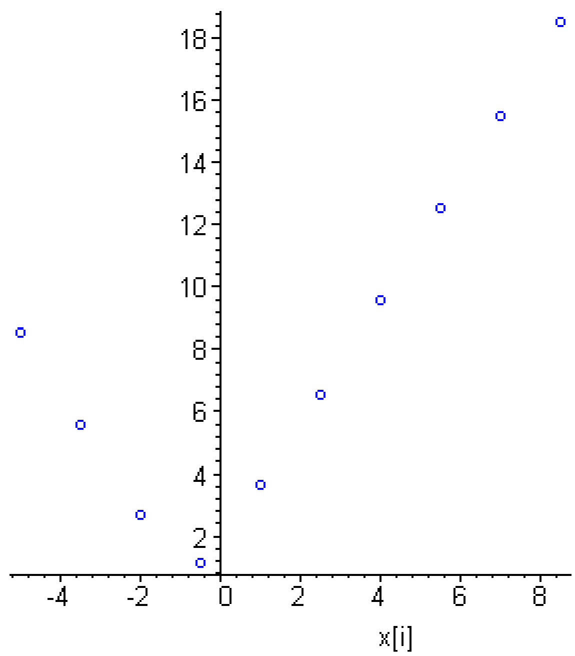
$$The\ rate\ of\ accumulation\ of\ the\ length\ function, \frac{\Delta L_i}{\Delta(x)}, is , \frac{\sqrt{293-204\,i+36\,i^2}}{2}$$

$$i, \quad , x_i = a + i\,h, \quad , \frac{\Delta L_i}{\Delta(x)}$$

0.	-5.	8.558621385
1.	-3.500000000	5.590169942
2.	-2.	2.692582404
3.	-0.5000000000	1.118033988
4.	1.	3.640054944
5.	2.500000000	6.576473220
6.	4.	9.552486585
7.	5.500000000	12.53993620
8.	7.	15.53222456
9.	8.500000000	18.52700732

```
> ratelist_plot:=evalf([seq([a+i*h,Delta_L(i)/h],i=0..n-1)]):
print('Delta*L[i]/Delta(x)');
p0:=plots[pointplot](ratelist_plot,color=blue,symbol=circle,
labels=["x[i]",""]):
plots[display](p0);
```

$$\frac{\Delta L_i}{\Delta(x)}$$



The rate at which the total length of the line segments accumulates with each added increment *h* is

$$\frac{\Delta L_i}{\Delta x} = \sqrt{1 + \left(\frac{f(x_i + h) - f(x_i)}{h}\right)^2}\quad .$$

Now we form a function that calculates the left Riemann sum of the ΔL_i 's for the first *k* line segments where *k* can range between 0 and *n*.

```
> Length[total]:=k->sum(Delta_L(i),i=0..k-1):
  print(`The`, 'L[total](k)', `function is`, Length[total](k));
```

$$The, L_{total}(k), function\ is, \sum_{i=0}^{k-1}\left(\frac{3\sqrt{293 - 204\,i + 36\,i^2}}{4}\right)$$

You should notice that for $L_{total}(k)$ Maple leaves the sum unevaluated. This is because Maple is unable to find a simple formula for the sum in terms of *k*, which is not surprising since the terms in the sum are square roots. In contrast, most of the sums for the area and volume problems were of simpler algebraic, trigonometric, or exponential terms. Nonetheless, we can still evaluate $L_{total}(k)$ for specific values of *k*. Maple will simply add the terms to obtain a numeric value whenever a value for *k* is specified in $L_{total}(k)$.

Next we form a list of the ordered pairs, where the first element of each ordered pair is the *x*-coordinate of the left end of the last line segment in the sum and the second element in each ordered pair is the total length of the first *k* line segments.

```
> lengthlist_print:=evalf([seq([kk,a+kk*h,Length[total](kk)],kk=1..n)]):
  print('k',` `,'x[k]=a+k*h',` `,'Length[total](k)');
  matrix(lengthlist_print);
```

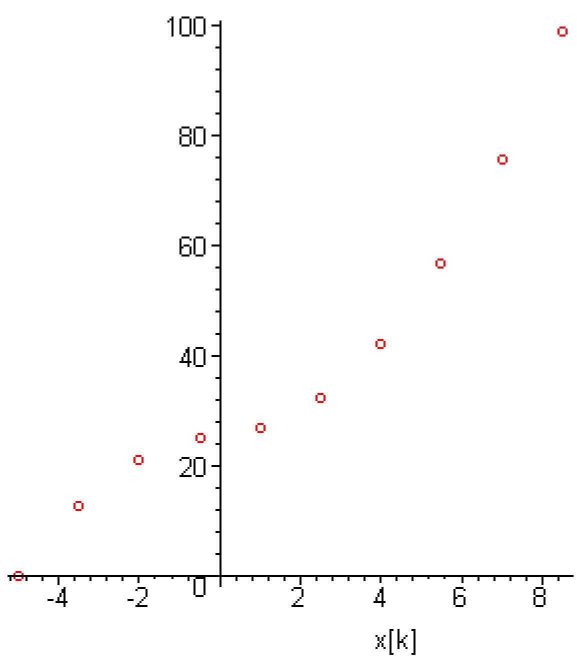
$$k, \ , x_k = a + k\,h, \ , Length_{total}(k)$$

1.	-3.500000000	12.83793208
2.	-2.	21.22318700
3.	-0.5000000000	25.26206060
4.	1.	26.93911158
5.	2.500000000	32.39919400
6.	4.	42.26390383
7.	5.500000000	56.59263371
8.	7.	75.40253802
9.	8.500000000	98.70087487
10.	10.	126.4913858

Let's plot the list of points and assign it to the symbol name **p1** for later use.

```
> lengthlist_plot:=evalf([seq([a+kk*h,Length[total](kk)],kk=0..n-1)]):
  print('Length[total](k)');
  p1:=plots[pointplot](lengthlist_plot,color=red,labels=["x[k]",""], symbol=circle):
  plots[display](p1);
```

$Length_{total}(k)$



Now we express $Length_{total}(k)$ as a function of x recalling that $k = \frac{x-a}{h}$. We call the new length function $Length_{approx}(x)$ since it approximates the length of the graph of $f(u)$ from $u = a$ to $u = x$.

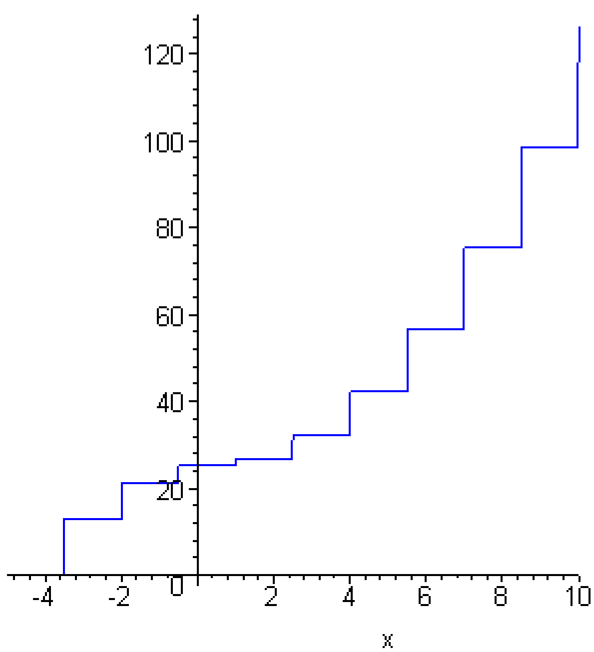
```
> Length[approx]:=x->Length[total]((x-a)/h):
  print(`The function`, 'Length[approx](x)',` is `, Length[approx](x));
```

$$\text{The function, } Length_{approx}(x), \text{ is , } \sum_{i=0}^{\frac{2x}{3}+\frac{7}{3}} \left(\frac{3\sqrt{293 - 204\,i + 36\,i^2}}{4} \right)$$

Now lets plot the $Length_{approx}(x)$ function, and then show the $Length_{total}(k)$ points and the $Length_{approx}(x)$ function on the same graph.

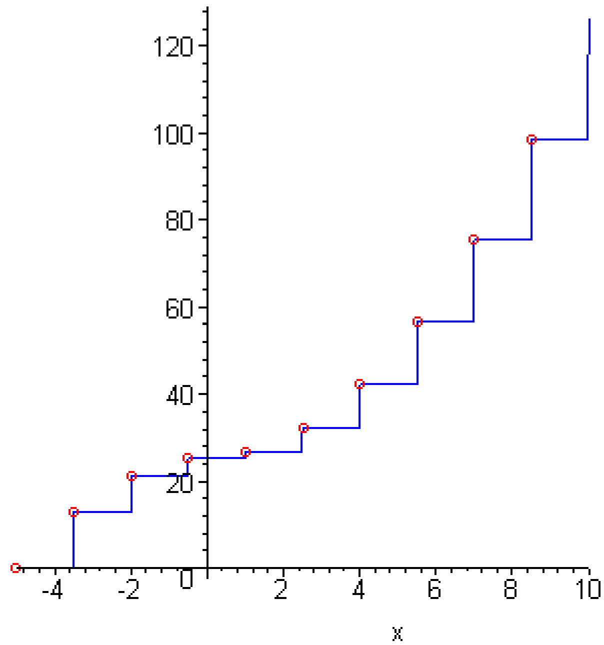
```
> print('Length[approx](x)');
  p2:=plot(Length[approx](x),x=a..a+n*h,color=blue,labels=["x",""]):
  plots[display](p2);
```

$Length_{approx}(x)$




```
> print('Length[approx](x)',`&`,'Length[total](k)');
  plots[display](p1,p2,labels=["x",""]);
```

$Length_{approx}(x), \&, Length_{total}(k)$



Can you explain the stair step pattern in the graph of $Length_{approx}(x)$?

The slopes of the secant lines taken from the $Length_{total}(k)$ graph give the rate at which the total length of the straight-line segments accumulates with each *h* increment of *x*. Now let's calculate the slopes of the secant lines between consecutive pairs of points on the graph above and plot them.

```
> m[secant]:=i->(Length[total](i+1)-Length[total](i))/h;
  print(`The secant slopes function`, 'm[secant](i)', ` is `, m[secant](i));
  msecantlist_print:=evalf([seq([k,a+k*h,m[secant](k)],k=0..n-1)]):
  print('i',`      `,'x[i]=a+ih',`      `,'m[secant](i)');
  matrix(msecantlist_print);
```

$$m_{secant} := i \to \frac{Length_{total}(i+1) - Length_{total}(i)}{h}$$

The secant slopes function, $m_{secant}(i)$, *is* ,

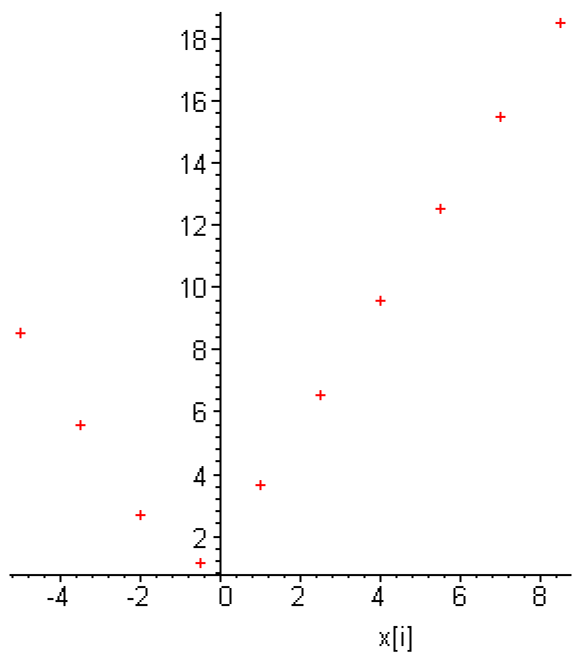
$$\frac{2}{3}\left(\sum_{i=0}^{i}\left(\frac{3\sqrt{293-204\,i+36\,i^2}}{4}\right)\right) - \frac{2}{3}\left(\sum_{i=0}^{i-1}\left(\frac{3\sqrt{293-204\,i+36\,i^2}}{4}\right)\right)$$

$$i, \quad , x_i = a + ih, \quad , m_{secant}(i)$$

0.	-5.	8.558621385
1.	-3.500000000	5.590169945
2.	-2.	2.692582404
3.	-0.5000000000	1.118033988
4.	1.	3.640054944
5.	2.500000000	6.576473220
6.	4.	9.552486585
7.	5.500000000	12.53993620
8.	7.	15.53222456
9.	8.500000000	18.52700732

```
> msecantlist_plot:=evalf([seq([a+k*h,m[secant](k)],k=0..n-1)]):
  print('m[secant](i)');
  p4:=plots[pointplot](msecantlist_plot,color=red,labels=["x[i]",""],symbol=cross):
  plots[display](p4);
```

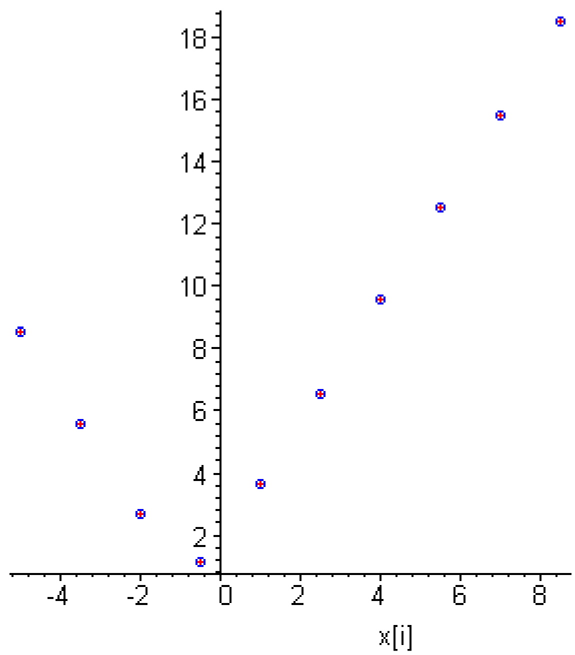
$m_{secant}(i)$



But this is the same thing as $\frac{\Delta L_i}{\Delta x}$ that we calculated at the beginning of this exercise. To confirm this we show **msecantslist** and $\frac{\Delta L_i}{\Delta x}$ together on the same graph.

```
> print('Delta*L[i]/Delta(x)',`&`,'m[secant](i)');
  plots[display](p0,p4,labels=["x[i]",""], symbol=circle);
```

$$\frac{\Delta L_i}{\Delta(x)}, \&, m_{secant}(i)$$



And yet again, it looks as though we have come full circle.

You Try It: Part III - Taking It to the Limit and Another Function

Chapter 6, Section 3

To help you with the exercises that follow, we have copied all of the commands from Part III into a procedure called **lengths( )**. In each exercise, you are asked to change some of the entries in the commands at the bottom of this section and then answer the questions. (Note that we have hidden all but the input commands and put them in the **lengths( )** procedure. In addition, we suppress displaying the lists of values that are generated because they become too large when *n* is a large number.)

1. For the function f(x) = x^2 considered in Part III, increase n, the number of straight-line segments between $x = a$ and $x = b$, so that the total length of the line segments approaches the length of the graph of $f(x)$ for $a <= x <= b$. Try n = 25, 50, 100, 250, 500. (For larger values of n, the evaluation of the commands will take a while because of the large number of a that must be added to form each value of $Length_{total}(k)$.) What is the $\lim_{\Delta x \to 0} \frac{\Delta L_i}{\Delta x}$? In the limit as $n \to \infty$ and $h \to 0$, what is the relationship between $\frac{\Delta L_i}{\Delta x}$ and $Length_{exact}(x)$?

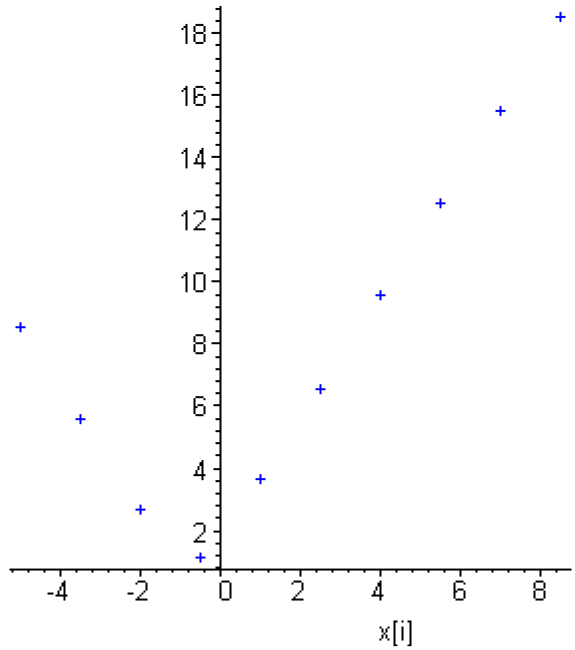
2. Repeat Exercise 1 for the following: $f(x) = 5\sin(2x)$, $a = 0$, $b = \pi$. In addition, explain the stair step pattern in the graph of $Length_{exact}(x)$. Under what conditions does the accumulated length of the straight-line segments increase the slowest with each increment h added to x, and under what conditions does it increase most rapidly? (Note that for the larger values of n, this will take a very long time to evaluate because of all the sums, square roots, and trig functions that need to be evaluated. Your computer will have to do some real "number crunching.")

```
> unassign('x','f');
  f:=x^2:
  a:=-5:
  b:=10:
  n:=10:
  lengths(a,b,n,f);
```

$$The, \Delta L_i, function\ is\ , \frac{3\sqrt{293 - 204\,i + 36\,i^2}}{4}$$

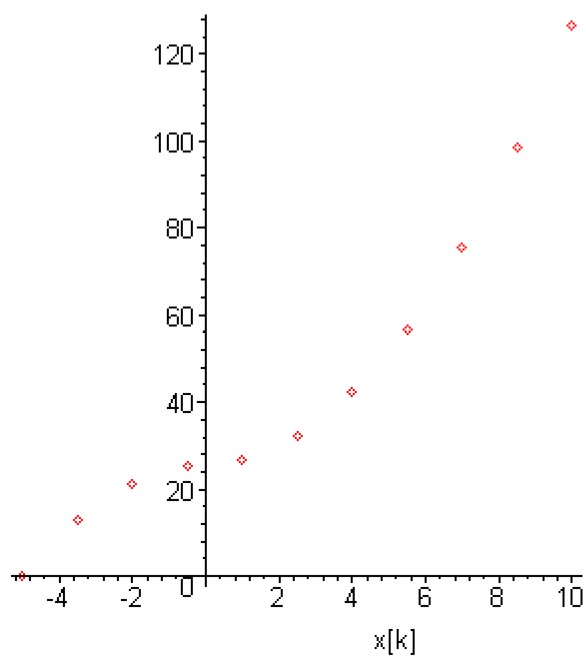
$$The, \frac{\Delta L_i}{\Delta(x)}, function\ is, \frac{\sqrt{293 - 204\,i + 36\,i^2}}{2}$$

$$\frac{\Delta L_i}{\Delta(x)}$$



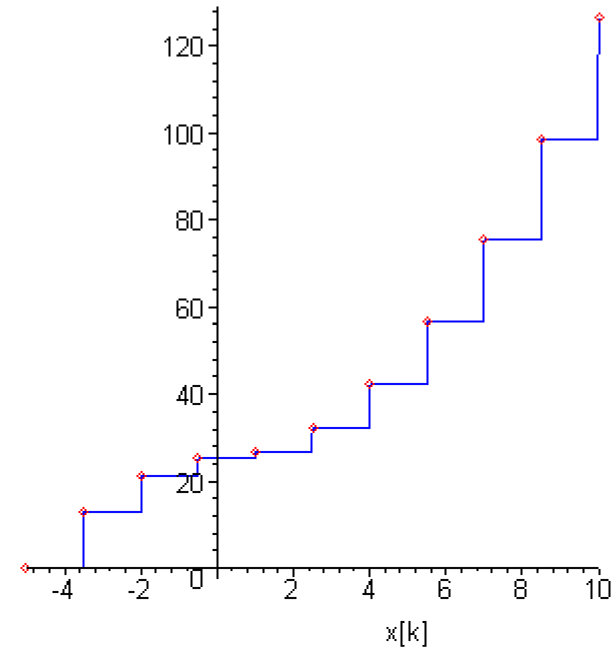
The, $Length_{total}(k)$, function is, $\sum_{i=0}^{k-1} \left(0.7500000000 \sqrt{293. - 204.\, i + 36.\, i^2}\right)$

$Length_{total}(k)$



The function, $Length_{exact}(x)$, is, $\sum_{i=0}^{\frac{2x}{3}+\frac{7}{3}} \left(0.7500000000 \sqrt{293. - 204.\, i + 36.\, i^2}\right)$

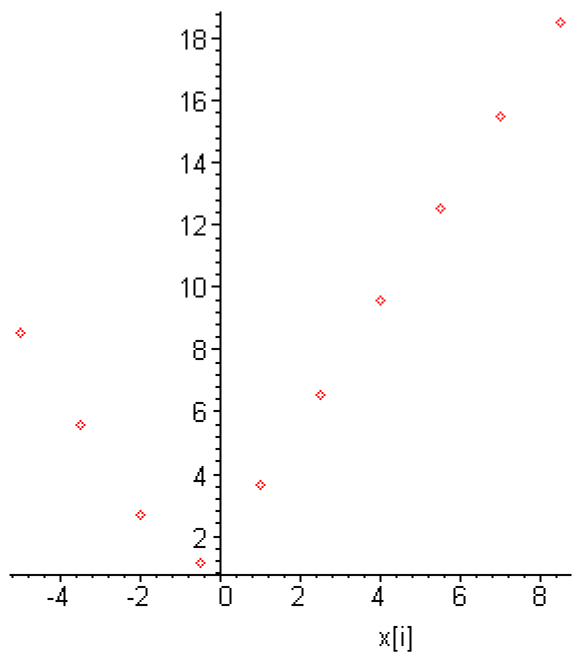
$Length_{total}(k), \&, Length_{approx}(x)$



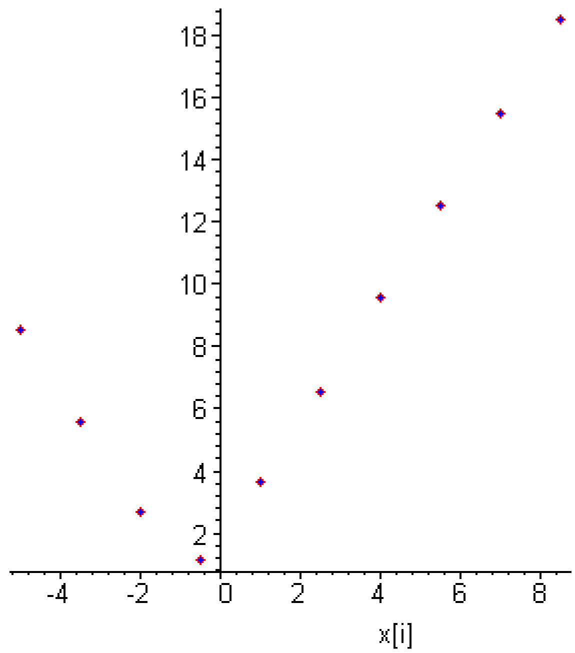
The function, $m_{secant}(i)$, *is* ,

$$\frac{2}{3}\left(\sum_{i=0}^{i} (0.7500000000\sqrt{293. - 204.\, i + 36.\, i^2})\right) - \frac{2}{3}\left(\sum_{i=0}^{i-1} (0.7500000000\sqrt{293. - 204.\, i + 36.$$

$m_{secant}(i)$



$$\frac{\Delta L_i}{\Delta(x)}, \&, m_{secant}(i)$$



>