Convergence of Secant Slopes to the Derivative Function

Introduction

OBJECTIVE: To visualize the secant line between successive points on a curve, observe what happens as the distance between successive points becomes small, compare the graph of successive secant line slopes with the graph of the derivative function, and apply the definition of the derivative in computing it.

One of the central problems of calculus is measuring and quantifying slopes on the graphs of nonlinear functions. In this module, you will explore two methods for approximating these slopes and see how the approximations approach the derivative function in the limit.

The first method is to extract a sample of evenly spaced points from the graph of a function y=f(x), calculate the slopes of the secant lines between the sample points, and graph the secant slope values together with the derivative function $f'(x)$. You will see that as the number of sample points increases and the horizontal spacing between consecutive points decreases, the set of discrete secant-slope values gets closer to the derivative function.

The second method for approximating the slopes on the graph of a function is to take every value of *x* in the domain of the function y=f(x) and calculate the slope of the secant line between the two points (x,f(x)) and (x+h, f(x+h)) for a fixed nonzero value of *h*. This gives a new function of *x* and *h*, namely, the function of secant slopes for any *x* in the domain of *f*. This secant-slope function is plotted together with the derivative $f'(x)$. You will see that as the value of *h* gets closer to 0, the secant slope function gets closer to $f'(x)$. Finally, you will see how to use *Mathematica* to take the limit of the secant-slope function as *h* approaches 0.

Before you begin this module, we recommend that you refer to "Tangents and Secants," a JAVA applet included in this supplement. This applet allows you to explore how the slopes of secant lines converge to the slope of the line tangent to a curve at a point.

▪ Technology Guidelines

NOTE: If you have just finished a module, restart *Mathematica* or close the *Kernel* before executing a new module.
TO OPEN CELLS, put your cursor on the right cell bracket and double click.
INITIALIZATION CELLS
When asked if you want to ". . . automatically evaluate all the initialization cells in the notebook . . . ," respond by pressing the "Yes" button.
TO STOP AN EXECUTION
Select the *Kernel* pull-down menu and click on *Abort Evaluation.*
ORDER OF EXECUTION
Execute cells in the order given. Do not skip any Input cells within a given notebook.
SAVING NOTEBOOKS
You can save anytime to any directory you choose, and it is wise to save often.
However, before you do your final save, delete all your output by selecting the
Delete All Output selection under the *Kernel* pull-down menu.
EXPERIENCING MAJOR PROBLEMS
Save if appropriate, then shut down *Mathematica* and start it up again.

Part I: Secant Slopes from a Discrete Set of Sample of Points

Chapter 2, Section 7 and Chapter 3, Section 1

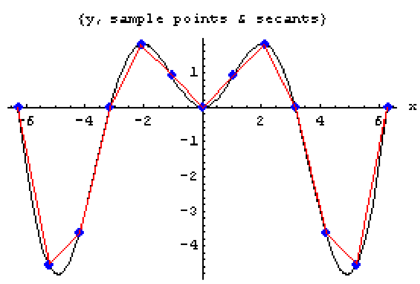
The function **derivApprox[ ]** extracts **n** evenly spaced sample points from a differentiable function and calculates the slopes of the secant lines between consecutive pairs of sample points. It then plots the function, the sample points, and the secant lines on one graph. A second graph is plotted to show the slopes of the secant lines in comparison with the derivative of the function. As the number of sample points increases, the secant slopes get closer to the derivative function.

(Note: **derivApprox[ ]** was written specially for this module and is not a built-in *Mathematica* command.)

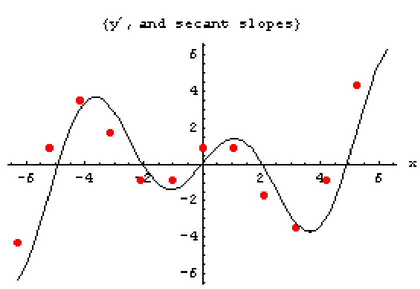
The arguments of **derivApprox[f_, x_, a_, b_, n_]** are: **f**, the function you wish to analyze; **x**, the independent variable; **a**, the beginning of the domain interval; **b**, the end of the domain interval; and **n**, the number of evenly spaced sample points. The following example illustrates how **derivApprox[ ]** works. All that you need to do is change the arguments.

In[171]:=

```
derivApprox[x*Sin[x], x, -2*Pi, 2*Pi,
 12];
```




```
The derivative of xSin[x] is  xCos[x] + Sin[x]
```



You Try It: On Differentiable and Non-Differentiable Functions

Chapter 2, Section 7 and Chapter 3, Section 1

1. Change the value of **n** (the last argument of **derivApprox[ ]**) and describe what you see. To do this systematically, you

might want to repeatedly double the number of sample points. That is, let **n**=12, 24, 48, 96,. . . .

2. Try **derivApprox[ ]** on some differentiable functions that you pick. Be sure to use several sample sizes so that you can see how the secant slopes function converges on the derivative. Here is an example of a function that has a very interesting derivative. See if you can see what makes it special.

In[172]:=

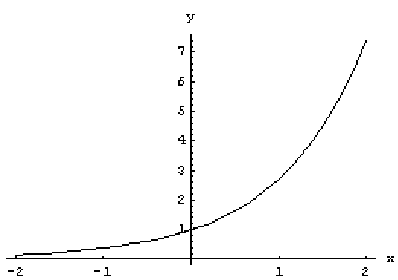
```
f[x_] = E^x
```

Out[172]=

e^x

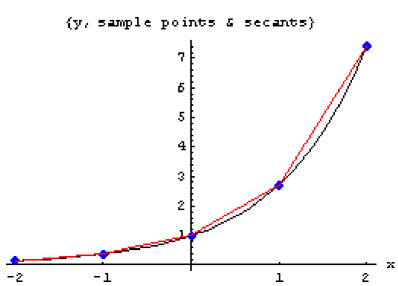
In[173]:=

```
Plot[f[x], {x, -2, 2},
 AxesLabel → {"x", "y"}];
```

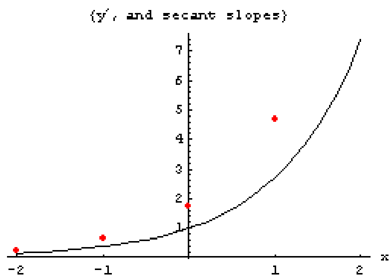


In[174]:=

```
derivApprox[f[x], x, -2, 2, 4]
```



The derivative of e^x is e^x



3. Try **derivApprox[ ]** on some functions that are not differentiable at one or more points. Here is an example.

In[175]:=

```
f[x_] = Which[x ≤ 2, 2 x, x > 2, 4 x - x^2]
```

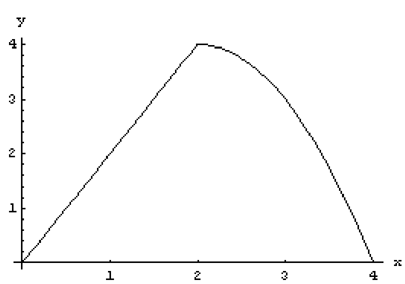
Out[175]=

```
Which[x ≤ 2, 2 x, x > 2, 4 x - x^2]
```

"> About *Mathematica*

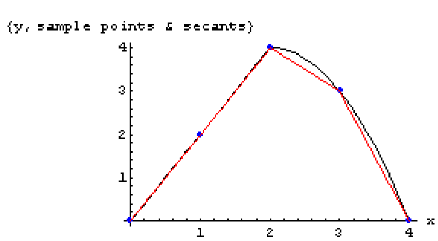
In[176]:=

```
Plot[f[x], {x, 0, 4},
 AxesLabel → {"x", "y"}];
```

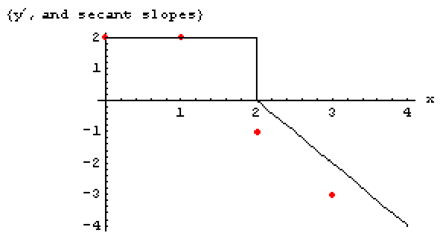


In[177]:=

```
derivApprox[f[x], x, 0, 4, 4]
```




```
The derivative of Which[x ≤ 2, 2 x, x > 2, 4 x - x^2] is  Which[x ≤ 2, 2, x > 2, 4 - 2 x]
```



The **derivApprox[ ]** command gives us a formula for the derivative function that is printed above. Is it correct in this case? To answer this question you should consider what happens at x=2.

Part II: Secant Slopes at Every Point

Chapter 3, Section 1

Another approach to approximating the slopes on the graph of a function is to take every value of *x* in the domain of the function y=f(x) and calculate the slope of the secant line between the two points (x,f(x)) and (x+h, f(x+h)) for a fixed nonzero value of *h*. This gives a new function of *x* and *h*, namely, the function of secant slopes for any *x* in the domain of *f*. We form this secant-slopes function in the next cell.

In[178]:=

```
msecant[x_, h_] := (f[x + h] - f[x])/h;
```

"> About *Mathematica*

Now, we graph the secant slopes for a specified *h* value together with the derivative function.

In[179]:=

```
f[x_] = x * Sin[x]
```

Out[179]=

```
x Sin[x]
```

In[180]:=

```
fprime[x_] = D[f[x], x]
```

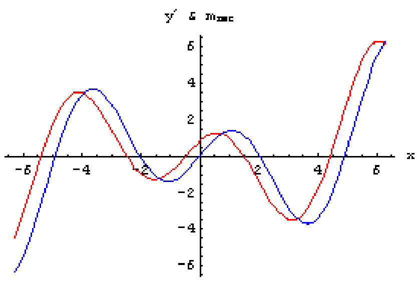
Out[180]=

```
x Cos[x] + Sin[x]
```

In[181]:=

```
h = 1;
```

```
Plot[{msecant[x, h], fprime[x]},
 {x, -2*Pi, 2*Pi},
 AxesLabel → {"x", "y′ & msec"},
 PlotStyle → {{RGBColor[1, 0, 0]},
   {RGBColor[0, 0, 1]}}];
```



The red curve gives the secant slope for each value of *x*, with h=1, and the blue curve gives the value of the derivative, that is, the slope of the tangent to the graph of f(x) for each value of *x*.

You Try It: On Differentiable and Non-Differentiable Functions

Chapter 3, Section 1

1. See what happens to the two graphs in the preceding cell when you decrease the value of *h*. Also, try some negative values of *h*. To be systematic, you should repeatedly cut *h* in half. For example, you could use h=-1,-0.5,-0.25, -0.125 for negative values of *h*. Describe what you observe.

2. Repeat the steps in Part II on some of differentiable functions that you pick. Be sure to decrease values of *h* so that you can see how the secant-slopes function, **msecant[ ]**, converges on the derivative function as *h* approaches zero. Here is an example.

In[183]:=

```
f[x_] = Sin[x]
```

Out[183]=

```
Sin[x]
```

In[184]:=

```
fprime[x_] = D[f[x], x]
```

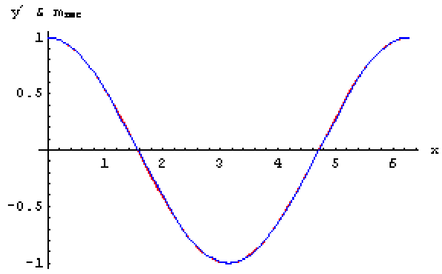
Out[184]=

```
Cos[x]
```

In[185]:=

```
h = 0.01;

Plot[{msecant[x, h], fprime[x]},
 {x, 0, 2*Pi},
 AxesLabel → {"x", "y′ & msec"},
 PlotStyle → {{RGBColor[1, 0, 0]},
   {RGBColor[0, 0, 1]}}];
```



3. Repeat the above steps on some functions that are not differentiable at one or more points. Here is an example.

In[187]:=

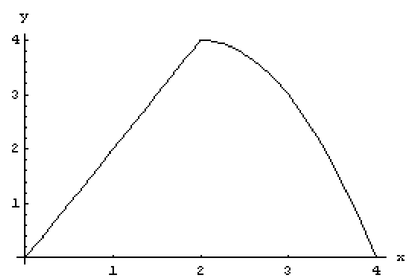
```
f[x_] = Which[x ≤ 2, 2 x, x > 2, 4 x - x^2]
```

Out[187]=

```
Which[x ≤ 2, 2 x, x > 2, 4 x - x^2]
```

In[188]:=

```
Plot[f[x], {x, 0, 4},
 AxesLabel → {"x", "y"}];
```



In[189]:=

```
fprime[x_] = D[f[x], x]
```

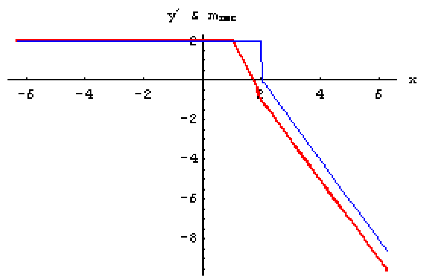
Out[189]=

```
Which[x ≤ 2, 2, x > 2, 4 - 2 x]
```

In[190]:=

```
h = 1;

Plot[{msecant[x, h], fprime[x]},
 {x, -2*Pi, 2*Pi},
 AxesLabel → {"x", "y′ & m_sec"},
 PlotStyle →
  {{RGBColor[1, 0, 0], Thickness[0.010]},
   {RGBColor[0, 0, 1]}}];
```



Note that we make the **msecant[ ]** graph a little thicker so that it is not hidden behind the graph of the derivative function.

Mathematica's derivative command, **D[ ]**, gives us a formula for the derivative function above. Is it correct in this case? To answer this question, you should consider what happens at x=2.

Part III: Taking It to the Limit

Chapter 3, Section 1

The slope of the tangent to the graph of y=f(x) for each value of *x* is given by $\lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$ provided, of course, that the limit exists. This gives the derivative function f′(x). In the next group of cells, we use this definition to find the derivative function for a specific function, y=f(x).

In[192]:=

```
Clear[f, h, x];
```

First, we pick a function.

In[193]:=

```
f[x_] = a x^2 + b x + c
  (* a, b, and c are constants *)
```

Out[193]=

```
c + b x + (6 - 6 t) x^2
```

Next, we calculate the secant slopes, that is, the difference-quotient.

In[194]:=

```
(f[x + h] - f[x])/h // Simplify
```

Out[194]=

```
b - 6 (-1 + t) (h + 2 x)
```

And we take the limit to form the derivative function.

In[195]:=

```
Limit[(f[x + h] - f[x])/h, h → 0]
```

Out[195]=

$b - 12(-1 + t)x$

You Try It: Finding Derivatives Using the Definition

Chapter 3, Section 1

Repeat the steps in Part III to find the derivatives of several functions that you pick. Here are some suggestions.

1. f(x)= $\sqrt{x}$

2. f(x)= $\frac{1}{x^2}$

3. f(x)=sin x

4. f(x)=A sin (a x) +B cos (b x)   where *A*, *a*, *B*, and *b* are constants

5. f(x)=tan x

6. f(x)= e^x

7. f(x)= $Ce^{a x}$ where *C* and *a* are constants

About *Mathematica*

The **Which[ ]** command can be used to form piecewise defined functions, as illustrated in the preceding cell. To learn more about the **Which[ ]** command, pull down the Help menu, select the Help Browser, and type **Which**. Go Back.

In the definition of **msecant[ ]**, we use the delayed assignment with the symbol **:=** (colon-equals). This is because we do not want the definition to store a secant slope definition for a specific **f[x_]**; rather, we want it to use the definition of **f[x_]** that is current when **msecant[ ]** is called. To learn more about the delayed assignments, see the "Overview of *Mathematica*" module that is included with this set of modules or pull down the Help menu, select the Help Browser, and type **:=**.
Go Back.