

Convergence of Secant Slopes to the Derivative Function

Chapter 3 Technology Application Project

1 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 Maxima 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.

1.1 Technology Guidelines

NOTE: If you have just finished a document, restart Maxima before executing a new document. This can be done by choosing "Restart" from the Maxima menu.

TO OPEN OR CLOSE CELLS

Click on the arrow at the top of the cell bracket.

TO STOP AN EXECUTION

Click on STOP button from the toolbar.

ORDER OF EXECUTION

Execute commands in the order given. Do not skip any Maxima Input lines within a given document.

Alternatively, you can execute the entire worksheet by selecting the "Evaluate All Cells" command from the "Cell" drop down menu or simply press Ctrl-r.

SAVING WORKSHEETS

You can save anytime to any directory you choose, and it is wise to save often.

EXPERIENCING MAJOR PROBLEMS

Save if appropriate, and then shut down Maxima and start it up again.

2 PART I: Secant Slopes from a Discrete Set of Sample of Points

Section 2.7 and Section 3.1

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 worksheet and is not a built-in Maple command. You should not worry about understanding the commands in the second of the next two cells, that is, unless you are interested in doing so. These commands define the `DerivApprox()` function, which is used in the remainder of this worksheet. All that you need to do is execute the commands in the next two cells so that `DerivApprox()` will work.

```
(%i1) load(draw)$  
      numer:true$  
      ratprint:false$
```

```

(%i4) DerivApprox (f, var, interval, n) :=
block(
  gmargin: 1,
  plist: [],
  slist: [],
  g: [],
  g2: [],
  flen: length(f),
  ilen: length(interval),
  if (flen#ilen) then block(
    print("The number of functions and number of intervals do not match. Please try again."),
    return()),
  k: 1,
  x0: interval[k][1],
  xn: interval[k][2],
  a: interval[1][1],
  b: interval[ilen][2],
  smally: subst(a, var, f[1]),
  bigy: subst(a, var, f[1]),
  for i: 0 thru n-1 do block(
    xval: a+i/(n-1)*(b-a),
    if (xval>=xn and xval<b) then k: k+1,
    if (k<=flen and xval<=b) then block(
      x0: interval[k][1],
      xn: interval[k][2],
      yval: subst(xval, var, f[k]),
      if(yval<smally) then smally: yval,
      if(yval>bigy) then bigy: yval,
      plist: append(plist, [[xval, yval]])),
  for i:1 thru flen do g: append(g, [explicit(f[i],var,interval[i][1],interval[i][2])]),
  wxdraw2d(
    yrange=[smally-gmargin,bigy+gmargin],
    color=red,
    points_joined=true,
    points(plist),
    color=black,
    point_type=6,
    point_size=1,
    points_joined=false,
    points(plist),
    color=blue,
    g,
    title="y, sample points and secants"),
  print(" "),
  print("The derivative of ", f, " is ", diff(f,var,1)),
  print(" "),
  plen: length(plist),
  smally: (plist[2][2]-plist[1][2])/(plist[2][1]-plist[1][1]),
  bigy: (plist[2][2]-plist[1][2])/(plist[2][1]-plist[1][1]),
  for j: 0 thru n-2 do block(

```

The arguments of `DerivApprox(f, var, interval, n)` are:

`f`: a list of functions you wish to analyze

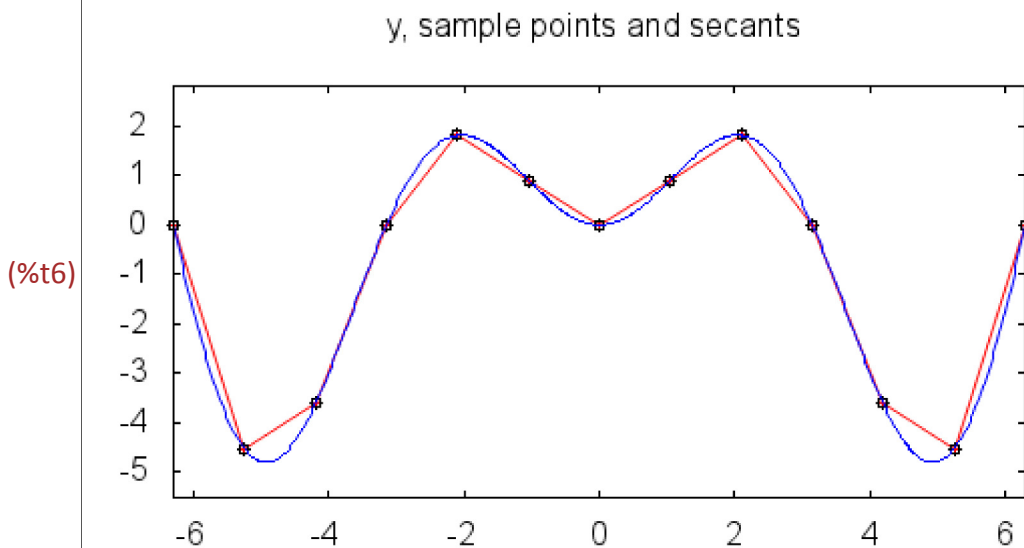
`var`: the independent variable

`interval`: a list of intervals of the form `[a,b]` for each function in `f` list

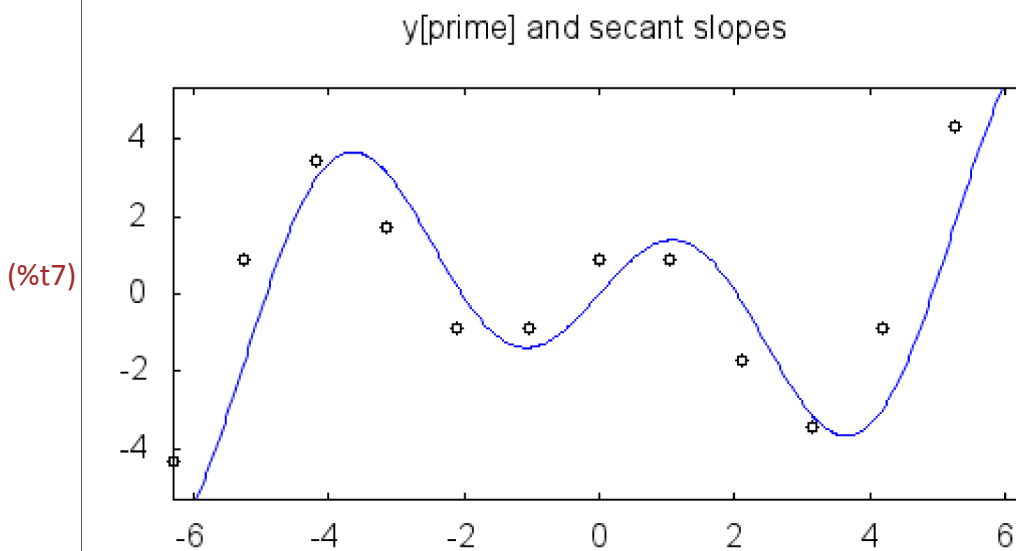
`n`: the number of evenly spaced sample points

The following example illustrates how `DerivApprox()` works. All that you need to do is change the arguments.

```
(%i5) interval1: [-2*%pi,2*%pi];
      DerivApprox([x*sin(x)], x, [interval1],13);
(%o5) [-6.283185307179586,6.283185307179586]
```



The derivative of $[x \sin(x)]$ is $[\sin(x) + x \cos(x)]$



(%o7)

2.1 You Try It: On Differentiable and Non-Differentiable Functions

Section 2.7 and Section 3.1

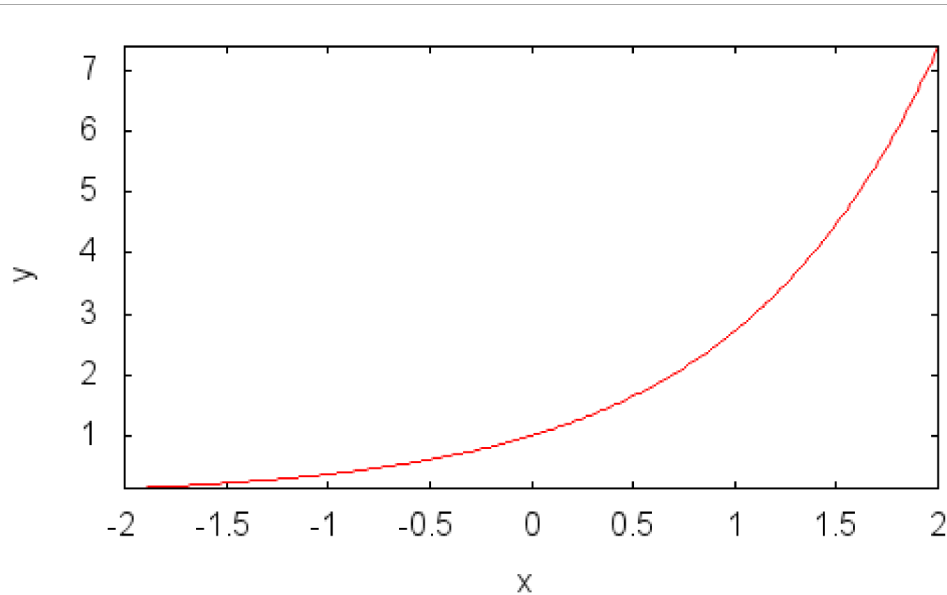
1. Change the value of n [the last argument of `DerivApprox()`] and describe what you see. To do this systematically, you might want to repeatedly double the number of sample points. That is, let $n = 6, 12, 24, 48, 96, \dots$.

2. Try `DerivApprox()` on some of 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. Do you see what makes this function special?

```
(%i8) f(x) := exp(x)$  
      f(x);  
      wxdraw2d(  
        color=red,  
        explicit(f(x),x,-2,2),  
        xlabel="x",  
        ylabel="y");
```

```
(%o9) %ex
```

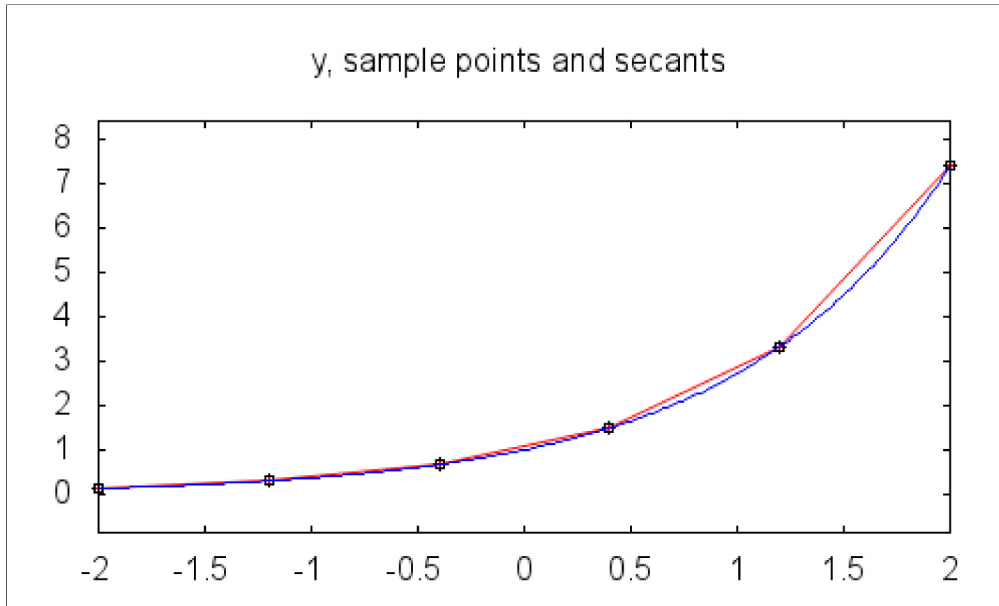
```
(%t10)
```



```
(%o10)
```

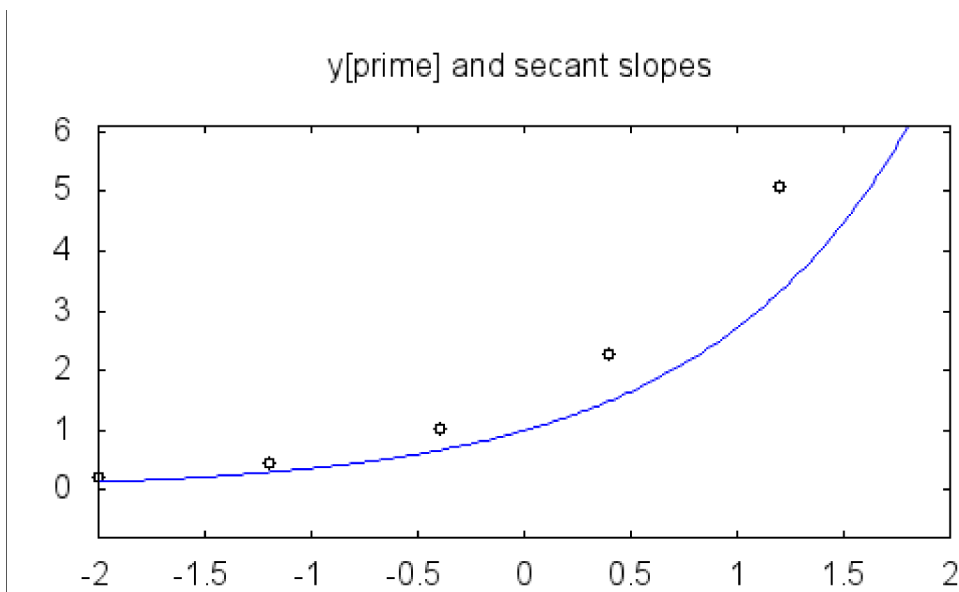
```
(%i11) DerivApprox([f(x)], x, [[-2, 2]], 6);
```

```
(%t11)
```



The derivative of [e^x] is [e^x]

```
(%t12)
```

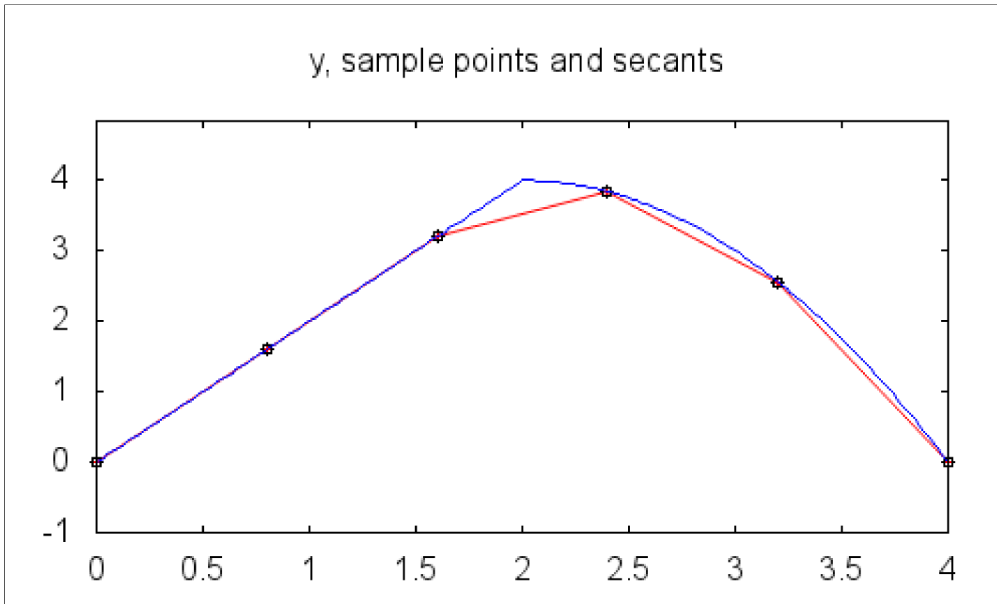


```
(%o12)
```

3. Try `DerivApprox()` on some functions that are not differentiable at one or more points. Here is an example.

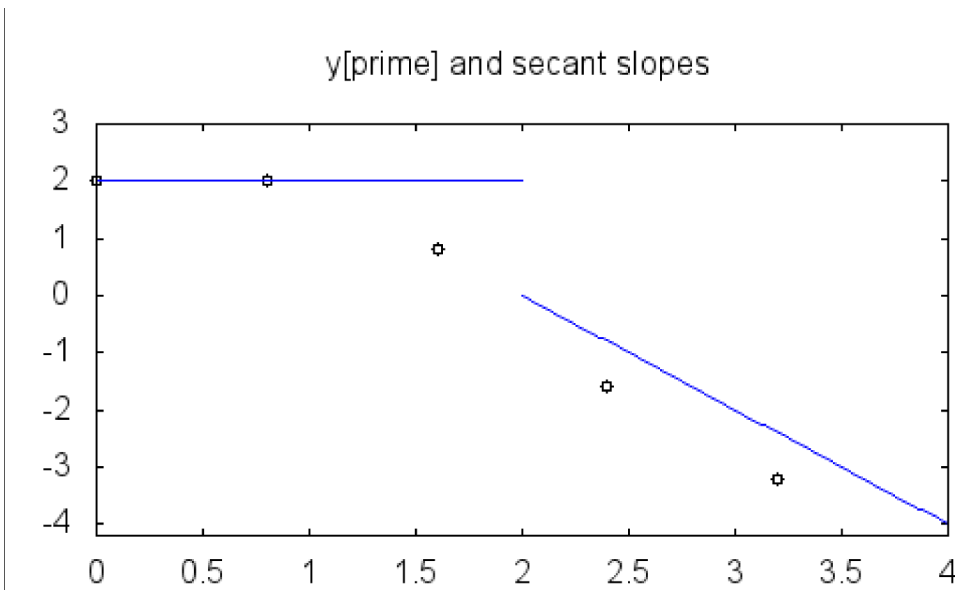
```
(%i13) DerivApprox([2*x,4*x-x^2],x,[0,2],[2,4],6);
```

```
(%t13)
```



The derivative of $[2x, 4x - x^2]$ is $[2, 4 - 2x]$

```
(%t14)
```



```
(%o14)
```

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 $x = 2$.

3 PART II: Secant Slopes at Every Point

Section 3.1

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 section.

```
(%i15) reset()$
      kill(all)$
      load(draw)$
      numer:true$
      ratprint:false$
```

```
(%i4) f(x) := x*sin(x);
      fprime: diff(f(x),x,1);
      msecant: (f(x+h)-f(x))/h;
```

```
(%o4) f(x) := x sin(x)
```

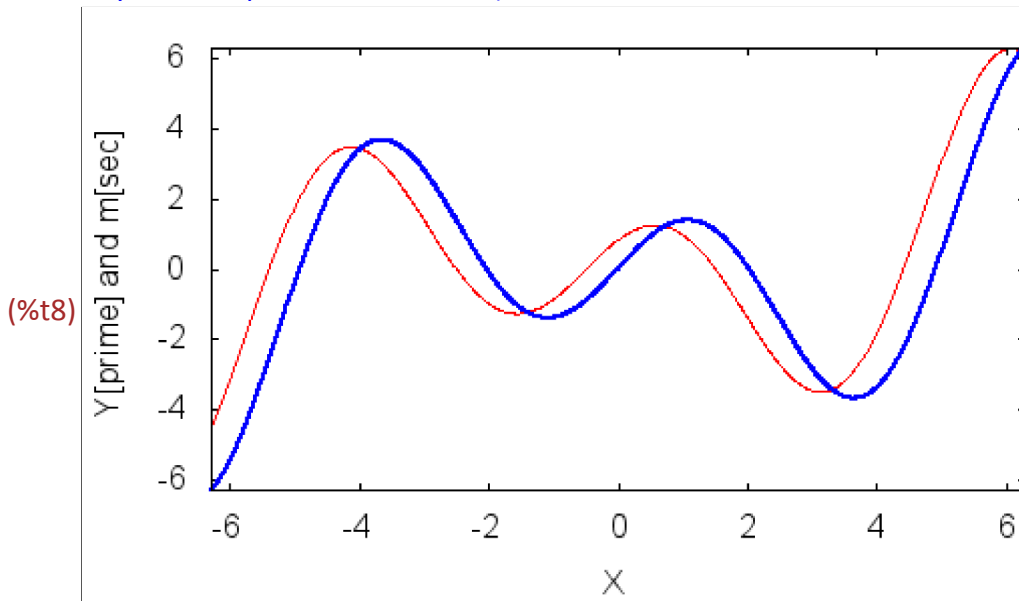
```
(%o5) sin(x) + x cos(x)
```

```
(%o6) 
$$\frac{(x+h) \sin(x+h) - x \sin(x)}{h}$$

```

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

```
(%i7) h: 1$
wxdraw2d(
  nticks=200,
  color=red,
  line_width=1,
  explicit(msecant,x,-2*%pi,2*%pi),
  color=blue,
  line_width=2,
  explicit(fprime,x,-2*%pi,2*%pi),
  xlabel="X",
  ylabel="Y[prime] and m[sec]");
```



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 .

3.1 You Try It: On Differentiable and Non-Differentiable Functions

Section 3.1

1. See what happens to the two graphs in the preceding section 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 $h = -1, -0.5, -0.25, -0.125, \dots$ for negative values of h . Describe what you observe.

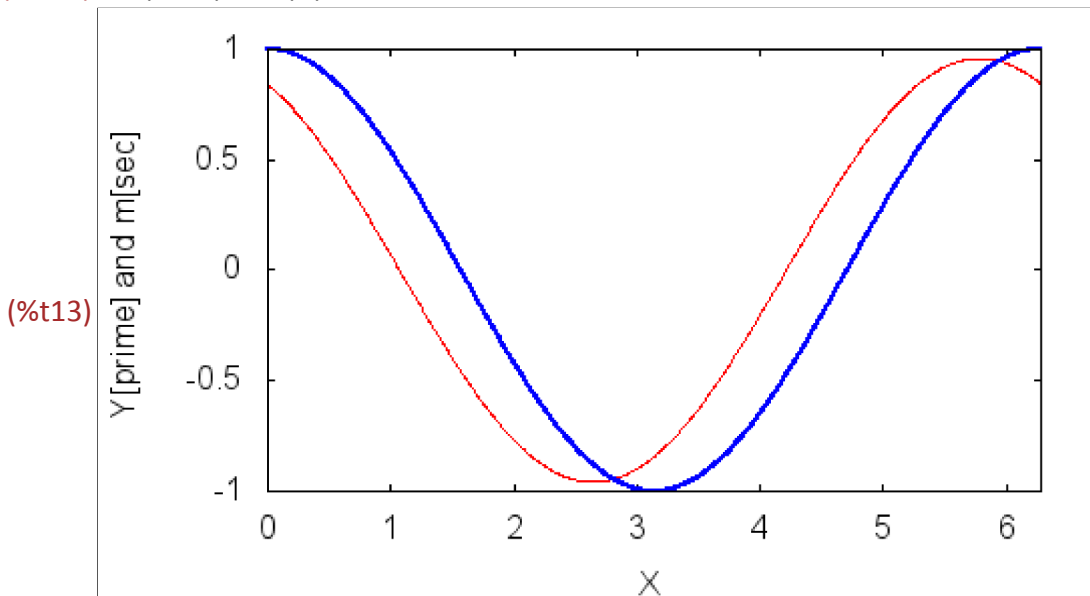
2. Repeat the steps in Part II on some of differentiable functions that you pick. Be sure to decrease values of h so that you can see how the secant-slopes function, `msecant`, converges on the derivative function as h approaches 0. Here is an example.

```
(%i9) f(x) := sin(x);
      fprime: diff(f(x),x,1);
      msecant: (f(x+h)-f(x))/h;
      h: 1$
      wxdraw2d(
        nticks=200,
        color=red,
        line_width=1,
        explicit(msecant,x,0,2*%pi),
        color=blue,
        line_width=2,
        explicit(fprime,x,0,2*%pi),
        xlabel="X",
        ylabel="Y[prime] and m[sec]");
```

```
(%o9) f(x) := sin(x)
```

```
(%o10) cos(x)
```

```
(%o11) sin(x + 1) - sin(x)
```



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

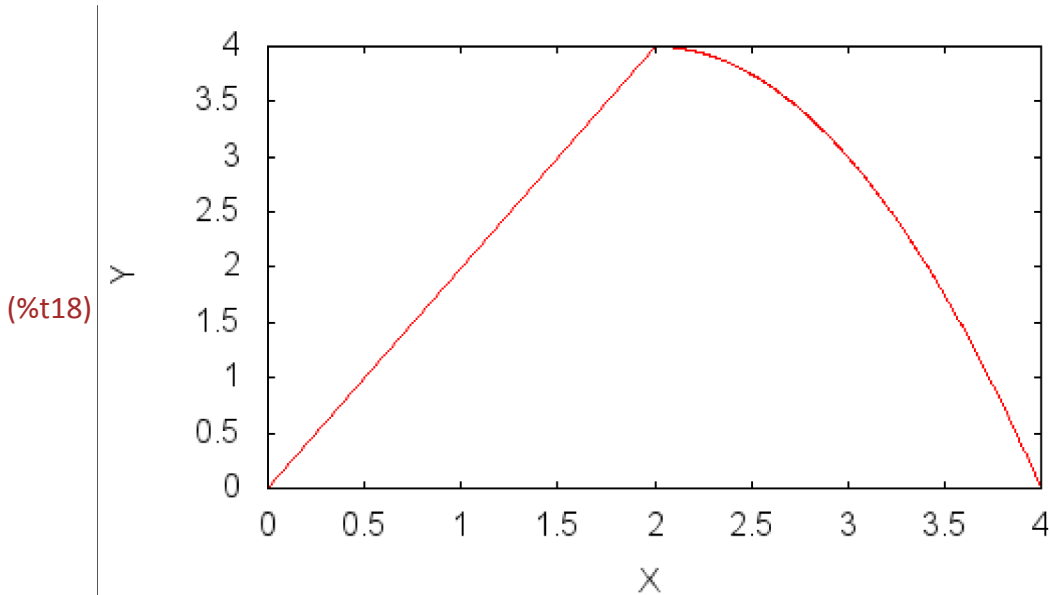
```
(%i14) f(x) := [2*x, 4*x-x^2];
interval: [[-2*%pi,2],[2,2*%pi]];
g1: [];
for i: 1 thru length(f(x)) do g1: append(g1, [explicit(f(x)[i],x,interval[i][1],interval[i][2])]);
wxdraw2d(
  nticks=200,
  color=red,
  g1,
  xlabel="X",
  ylabel="Y",
  xrange=[0,4],
  yrange=[0,4]);
fprime(x) := diff(f(x),x,1);
msecant(x) := (f(x+h)-f(x))/h;
```

```
(%o14) f(x):=[ 2 x, 4 x - x2]
```

```
(%o15) [[-6.283185307179586, 2],[2, 6.283185307179586]]
```

```
(%o16) []
```

```
(%o17) done
```



```
(%o18)
```

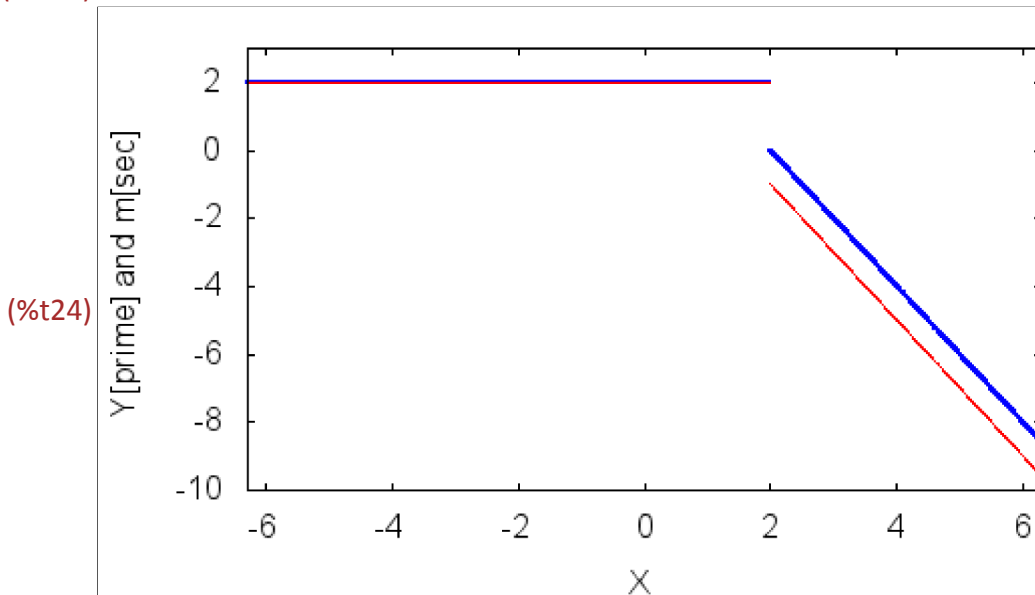
```
(%o19) fprime(x) := diff(f(x), x, 1)
```

```
(%o20) msecant(x) := 
$$\frac{f(x+h) - f(x)}{h}$$

```

```
(%i21) g2: []$
      g3: []$
      for i: 1 thru length(f(x)) do block(
        g2: append(g2, [explicit(fprime(x)[i],x,interval[i][1],interval[i][2])]),
        g3: append(g3, [explicit(msecant(x)[i],x,interval[i][1],interval[i][2])]);
      wxdraw2d(
        nticks=200,
        color=blue,
        line_width=2,
        g2,
        color=red,
        line_width=1,
        g3,
        xlabel="X",
        ylabel="Y[prime] and m[sec]",
        xrange=[interval[1][1],interval[2][2]],
        yrange=[-10,3]);
```

(%o23) done



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

Maxima's derivative command, `diff()`, 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 $x = 2$.

4 PART III: Taking It to the Limit

Section 3.1

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

First, we pick a function.

```
(%i25) reset()$
      kill(all)$
      load(draw)$
      numer:true$
      ratprint:false$
```

```
(%i4) f(x) := a*x^2+b*x+c;
```

```
(%o4) f(x) := a x^2 + b x + c
```

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

```
(%i5) ratsimp((f(x+h)-f(x))/h);
```

```
(%o5) 2 a x + a h + b
```

And we take the limit to form the derivative function.

```
(%i6) limit((f(x+h)-f(x))/h, h, 0);
```

```
(%o6) 2 a x + b
```

4.1 You Try It: Finding Derivatives Using the Definition

Section 3.1

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. $f(x) = 1/(x^2)$
3. $f(x) = \sin(x)$
4. $f(x) = A \sin(ax) + B \cos(bx)$, where A , a , B , and b are constants.
5. $f(x) = \tan(x)$
6. $f(x) = e^x$
7. $f(x) = C e^{ax}$