

## Explore

---

The [Explore](#) function provides an easy-to-use tool for creating interactive applications within the current document or worksheet. Maple 2016 introduces significant updates and additions:


### ▼ Simplified calling sequence

Parameters which take on values from a numeric range or a list can now be specified more simply, without using the *parameters* option of the *Explore* command. This makes *Explore* easier to use, in the simplest cases.

For example, in previous versions the following example would require a more verbose argument, such as *parameters* = [*n* = 1 .. 6].

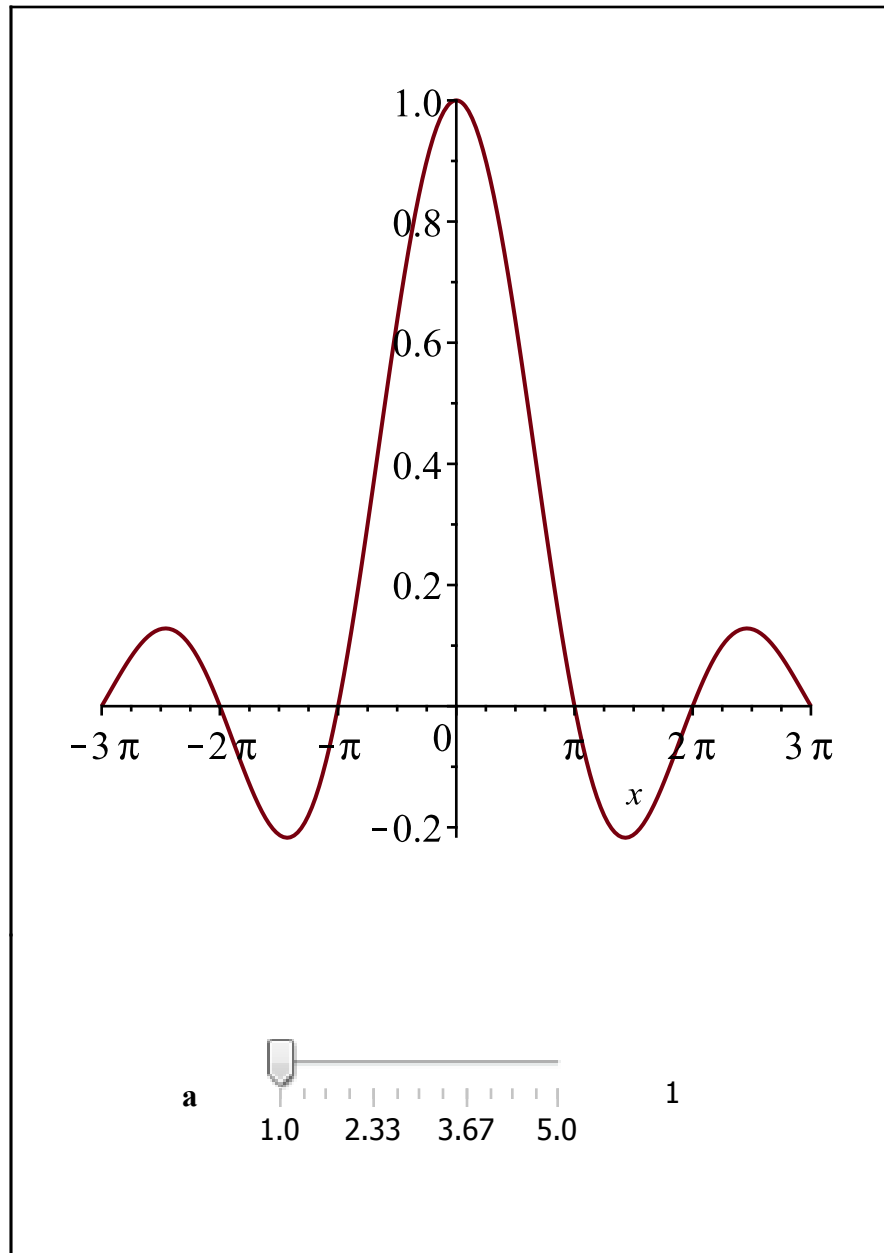
`Explore( factor( sum( i^n, i = 1 .. N ) ), n = 1 .. 6 )`

$$\text{factor}\left(\sum_{i=1}^N i^2\right) = \frac{1}{6} N (N + 1) (2 N + 1)$$



The slider shows the parameter **n** ranging from 1 to 6. The current value is 2.

*Explore*(*plot*( $\frac{\sin(a \cdot x)}{x}$ ,  $x = -3 \cdot \pi .. 3 \cdot \pi$ ),  $a = 1.0 .. 5.0$ )

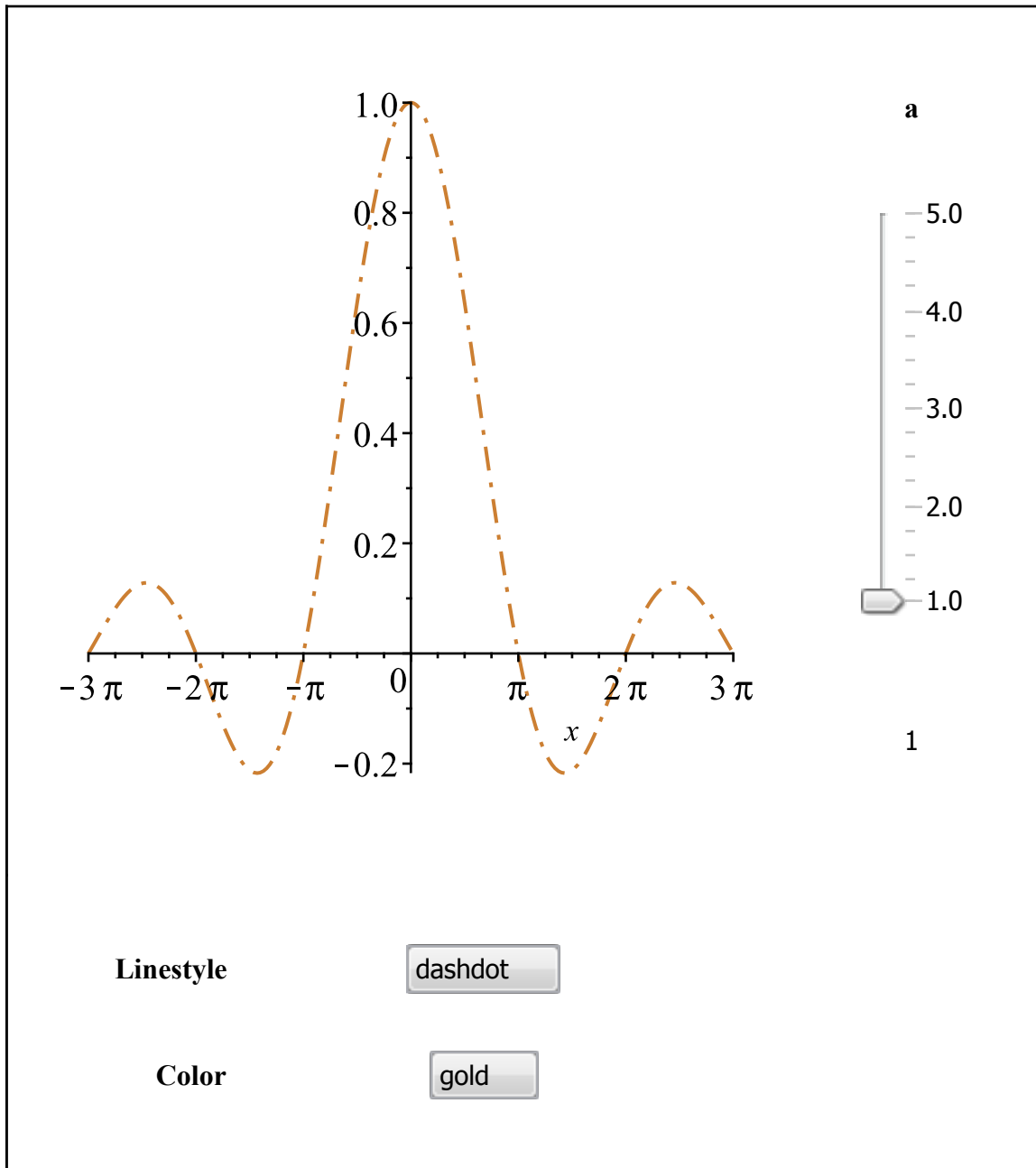


Parameters with a list of values can also be used with this shorter calling sequence. This calling sequence also works in the mixed case of some additional parameters being specified in more detail using the *parameters* option.

*Explore*(*plot*( $\frac{\sin(a \cdot x)}{x}$ ,  $x = -3 \pi .. 3 \pi$ , *color* = *Color*, *linestyle* = *Linestyle*),

*Linestyle* = [*dashdot*, *dot*, *dash*, *solid*], *Color* = [*gold*, *blue*, *red*],

```
parameters = [[a = 1.0 ..5.0, placement = right, orientation = vertical],  
placement = bottom)
```



## ▼ Name resolution

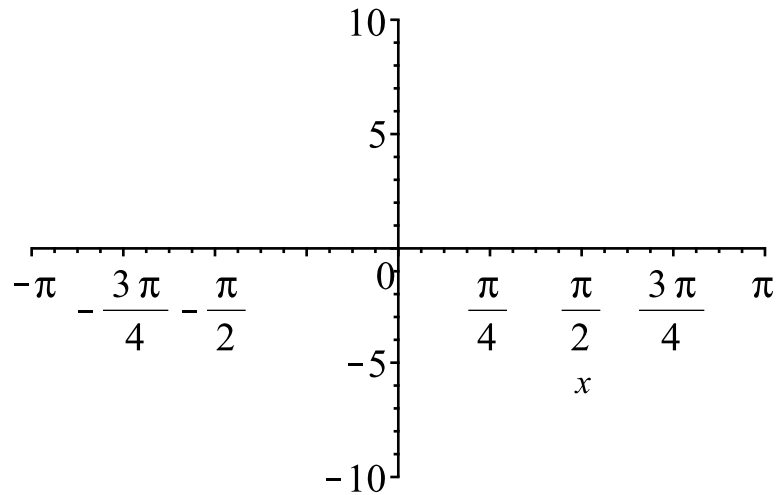
The *Explore* command has special evaluation rules, and its first argument *expr* is not evaluated before the *Explore* command has the opportunity to examine it. This is an

important mechanism, and is central to how *Explore* can accept, for example, a plotting command containing unassigned parameters.

In the following example, the standalone call to *plot* fails to produce a meaningful plot since the parameter *a* has no assigned numeric value. As expected, an empty plot is produced, along with a warning about the presence of the unassigned name *a*.

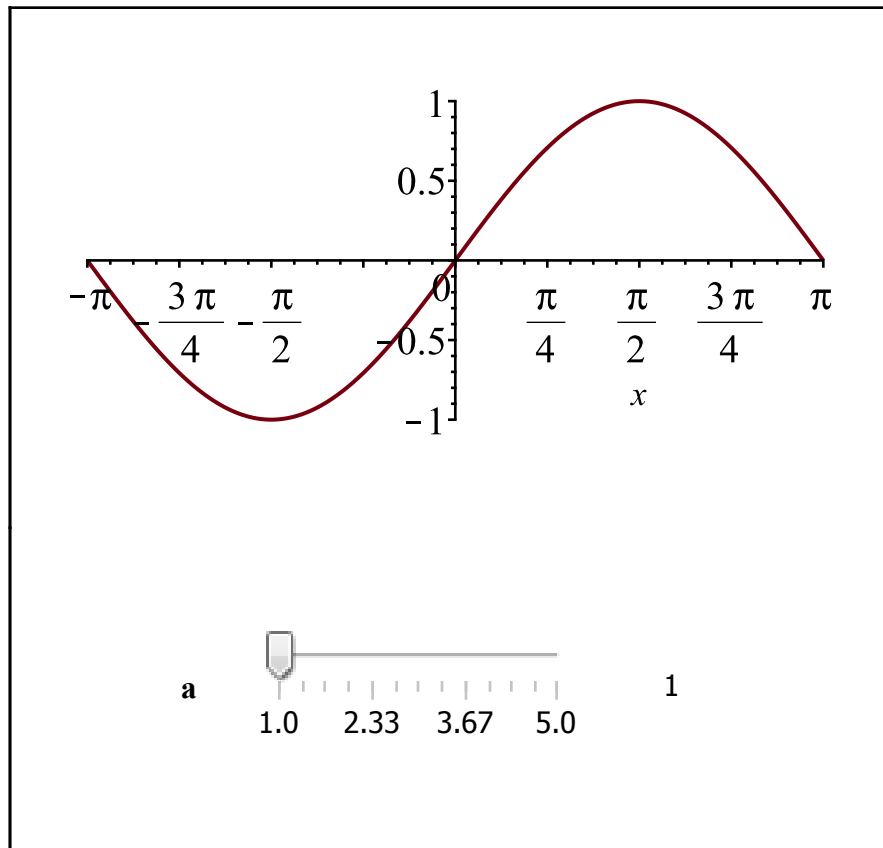
```
plot(sin(a·x), x=-Pi..Pi, size = [300, 200])
```

Warning, expecting only range variable x in expression sin(a\*x) to be plotted but found name a



The *Explore* command accommodates this by not evaluating the explored command until it has numeric values for *a*.

```
Explore( plot(sin(a·x), x=-π..π, size = [400, 200]), a = 1.0 .. 5.0 )
```



In versions previous to Maple 2016, all exploration parameters had to be explicitly given in *expr* in order for the exploration to work as expected. That is, if you had assigned  $w$  to  $\sin(a \cdot x)$  in the preceding example and then tried to explore  $w$ , you would have received unintended results or a warning about the unexpected presence of the name  $w$ . Therefore, in order to explore an expression that depended on a previously assigned name, it was necessary to use a more involved process, such as using a custom procedure that accepted  $a$  as its parameter and would resolve  $w$  during its execution.

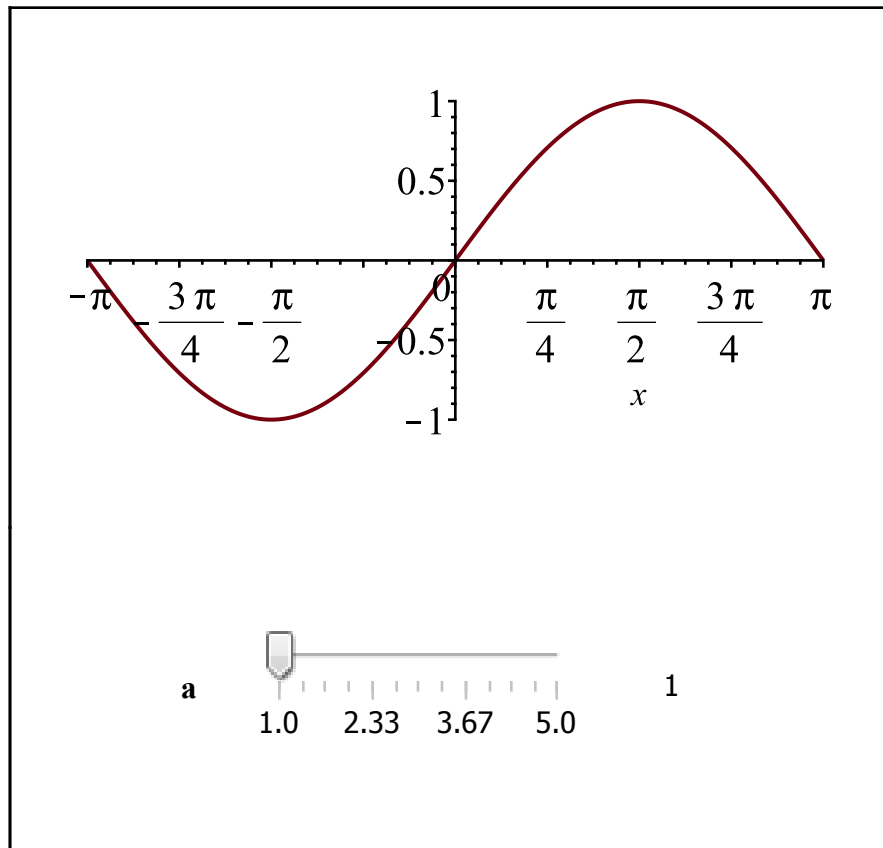
*restart;*

$w := \sin(a \cdot x)$

$\sin(a \cdot x)$

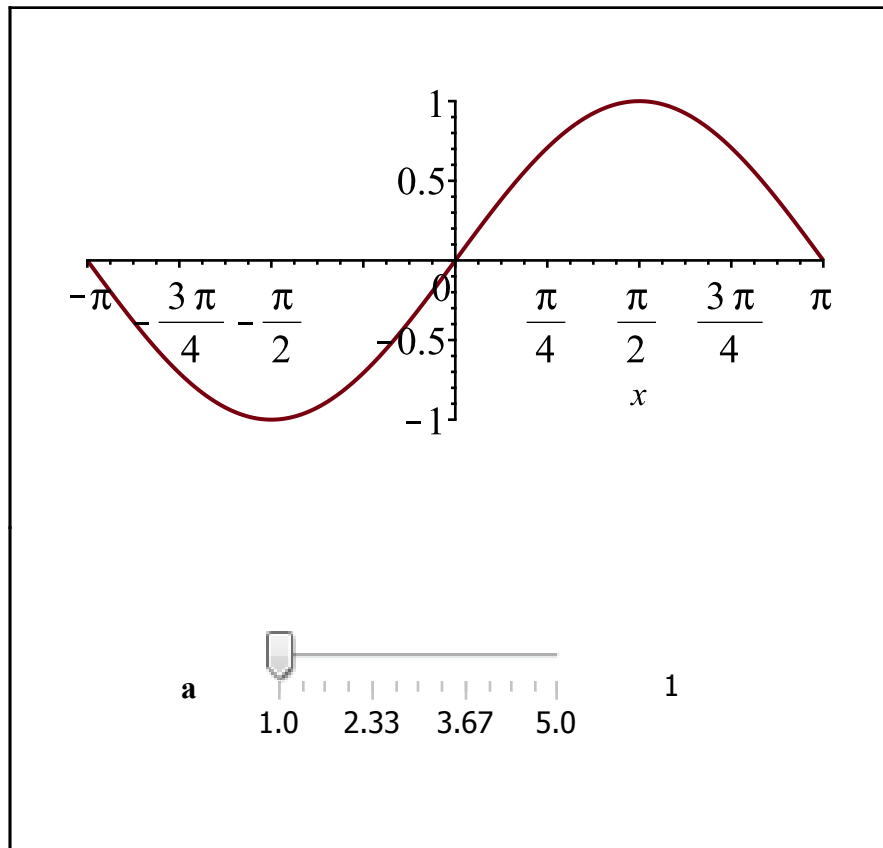
$F := \mathbf{proc}(A) \text{ plot}(eval(w, a = A), x = -\pi .. \pi, size = [400, 200]) \mathbf{end proc};$

$Explore(F(a), parameters = [a = 1.0 .. 5.0])$



In **Maple 2016** the name  $w$  is resolved to its assigned value by *Explore*, while the plotting command is still usefully left evaluated until parameter name  $a$  gets its numeric values. This eliminates the need for a custom procedure. Also note that the following exploration will continue to operate as intended after a restart.

`Explore(plot(w, x = -π..π, size = [400, 200]), parameters = [a = 1.0 .. 5.0] )`



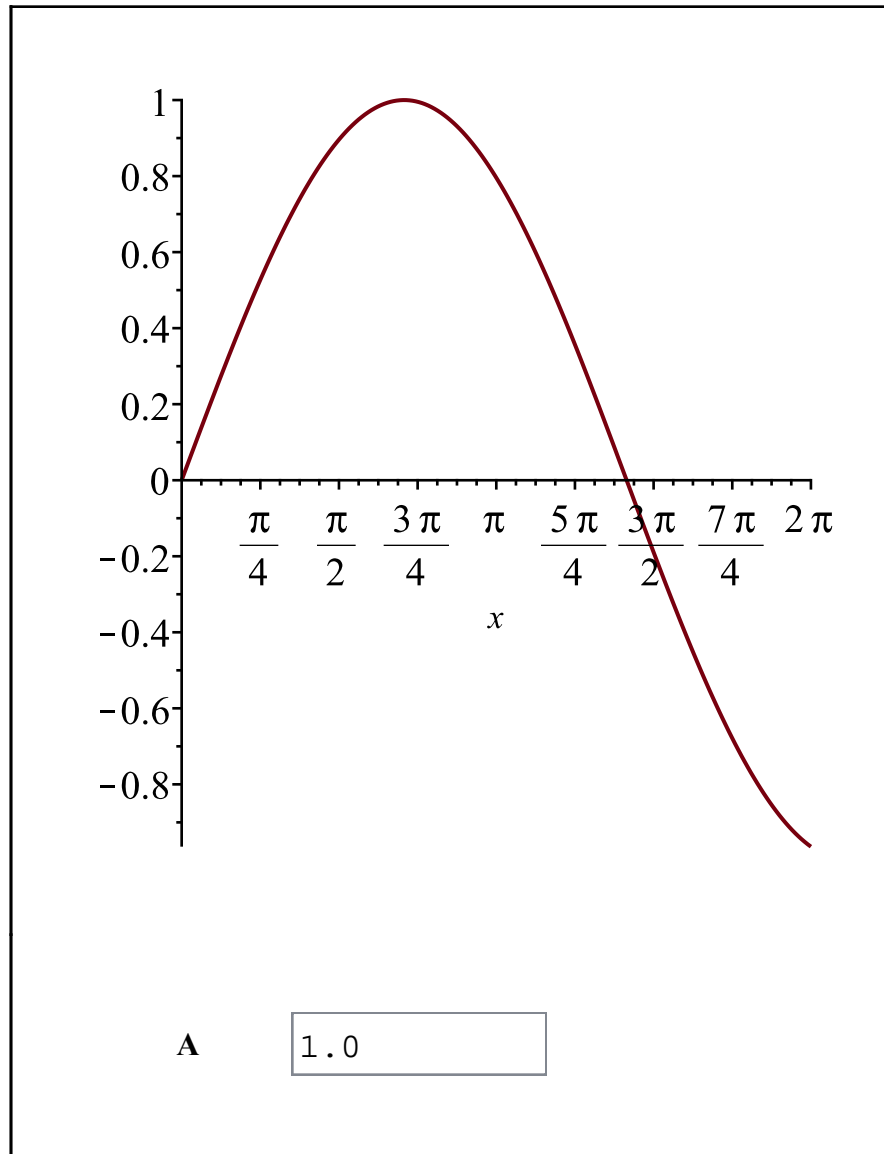
Names are resolved, recursively, to algebraic expressions involving other names. However, resolution of a name is not done if the value of the name is a Matrix/Vector/Array with hardware datatype or something of type [last\\_name\\_eval](#), such as a procedure. Hence in the previous example the assigned name  $F$  is not resolved and replaced, since its value is a procedure.

## ▼ **TextArea controllers for evaluating at custom values of a parameter**

With a [TextArea](#) controller the value for a parameter can be any custom value or expression. In the following exploration, type a value in the embedded TextArea component for parameter  $A$ , and then press the **Enter** key to update the plot.

For example, the supplied value could be  $\text{sqrt}(2)$ , or an expression in  $x$  such as  $x^2$ .

*Explore*  $\left( \text{plot} \left( \sin \left( \frac{A}{\sqrt{2}} \cdot x \right), x = 0 .. 2 \cdot \pi \right), \text{parameters} = [[A, \text{controller} = \text{"textarea"}]], \text{initialvalues} = [A = 1.0] \right)$



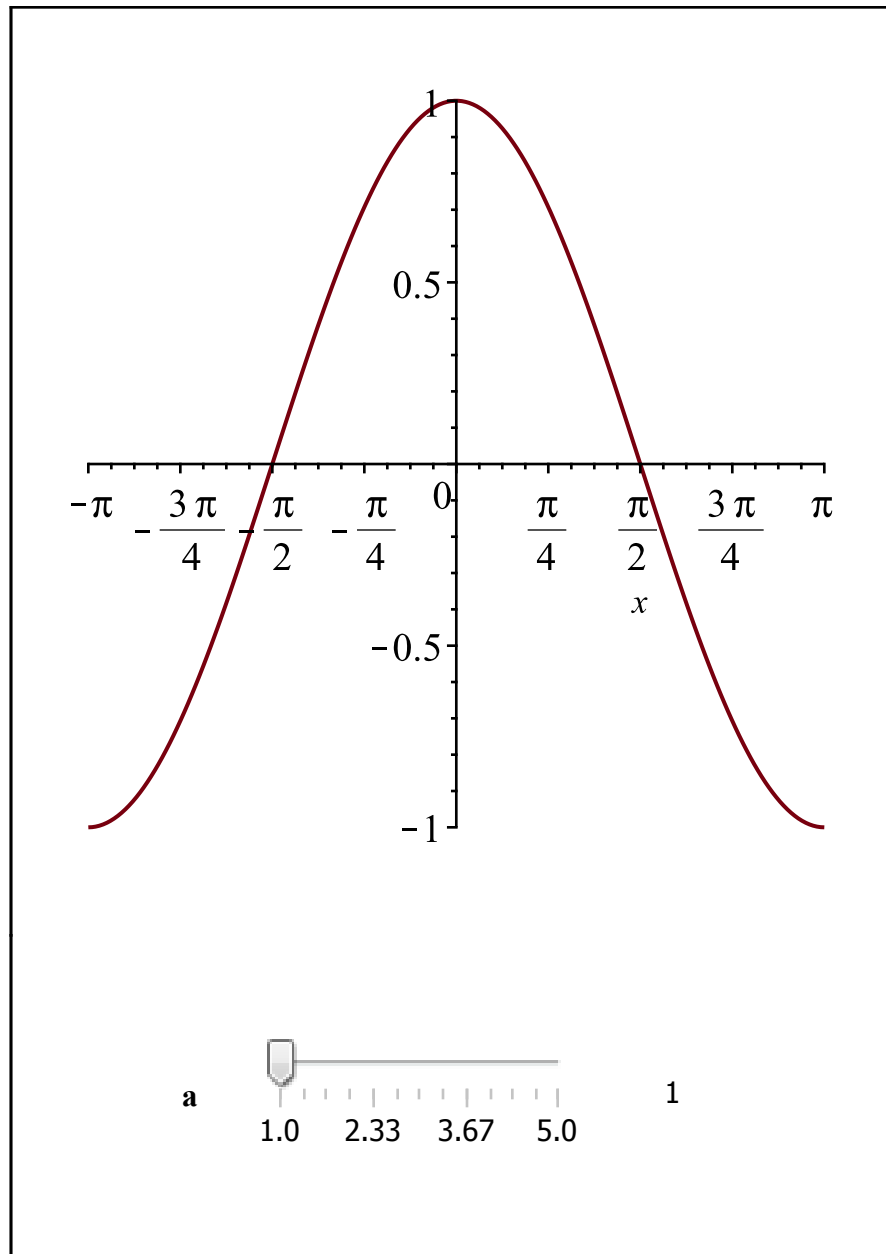
## ▼ Controller customization

Parameters given using the *parameters* option can be further specified with optional arguments that are valid for the constructor of the relevant controller.



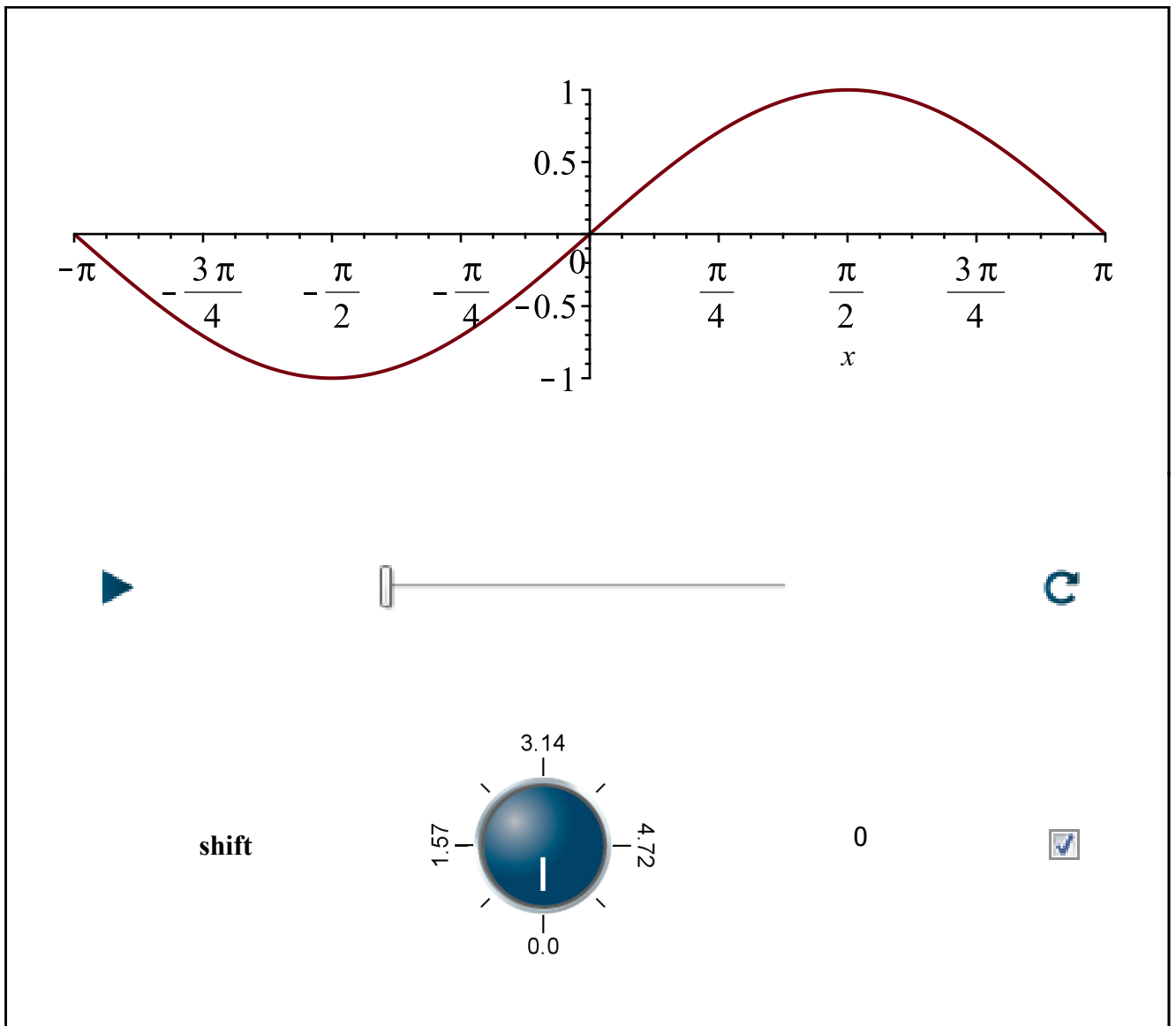
In the following example, the *continuous* option of the Slider component is used. The effect is that when the Slider is moved manually using the mouse pointer, the plot will not recompute and update until the sliding movement is finished.

`Explore(plot(cos(a·x), x=-π..π), parameters = [[a = 1.0 ..5.0, continuous = false]])`



In the following example the options *startangle*, *anglerange*, *majorticks*, *height*, and *width* are passed through to the [Dial](#) programmatic constructor.

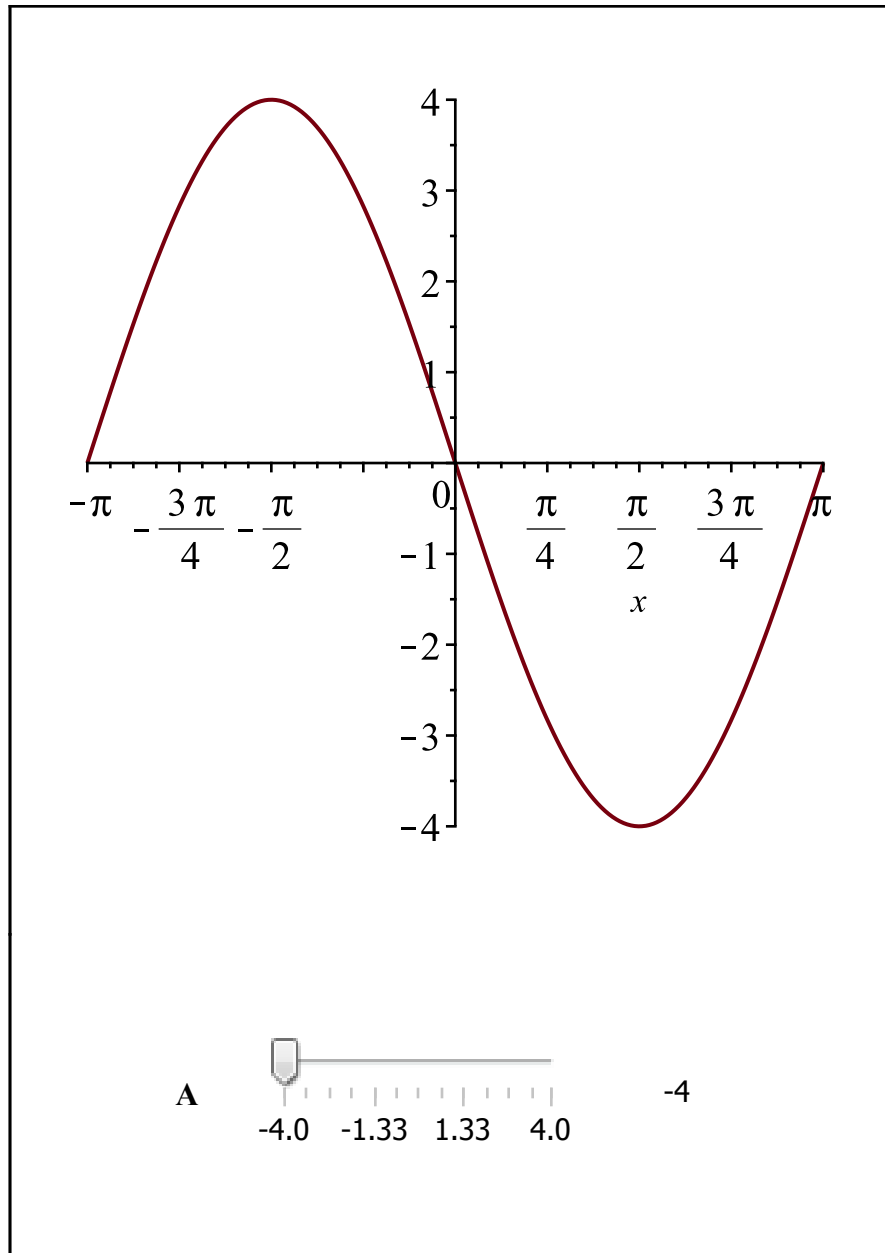
```
Explore( plot( sin( (shift * (50 - 1) / 50) + x ), x = -pi..pi ), loop, numframes = 50, size = [600, 200],
  parameters = [ [ shift = 0..2 * pi, controller = "dial", startangle = 0, anglerange = 360,
    majorticks = pi / 2 + 0.001, height = 100, width = 100 ] ] )
```



In the next example the *tooltip* option is passed to the programmatic constructor of the controller component, which by default is a [Slider](#).

In the following exploration, hovering over the Slider component with the mouse pointer will reveal the tooltip string.

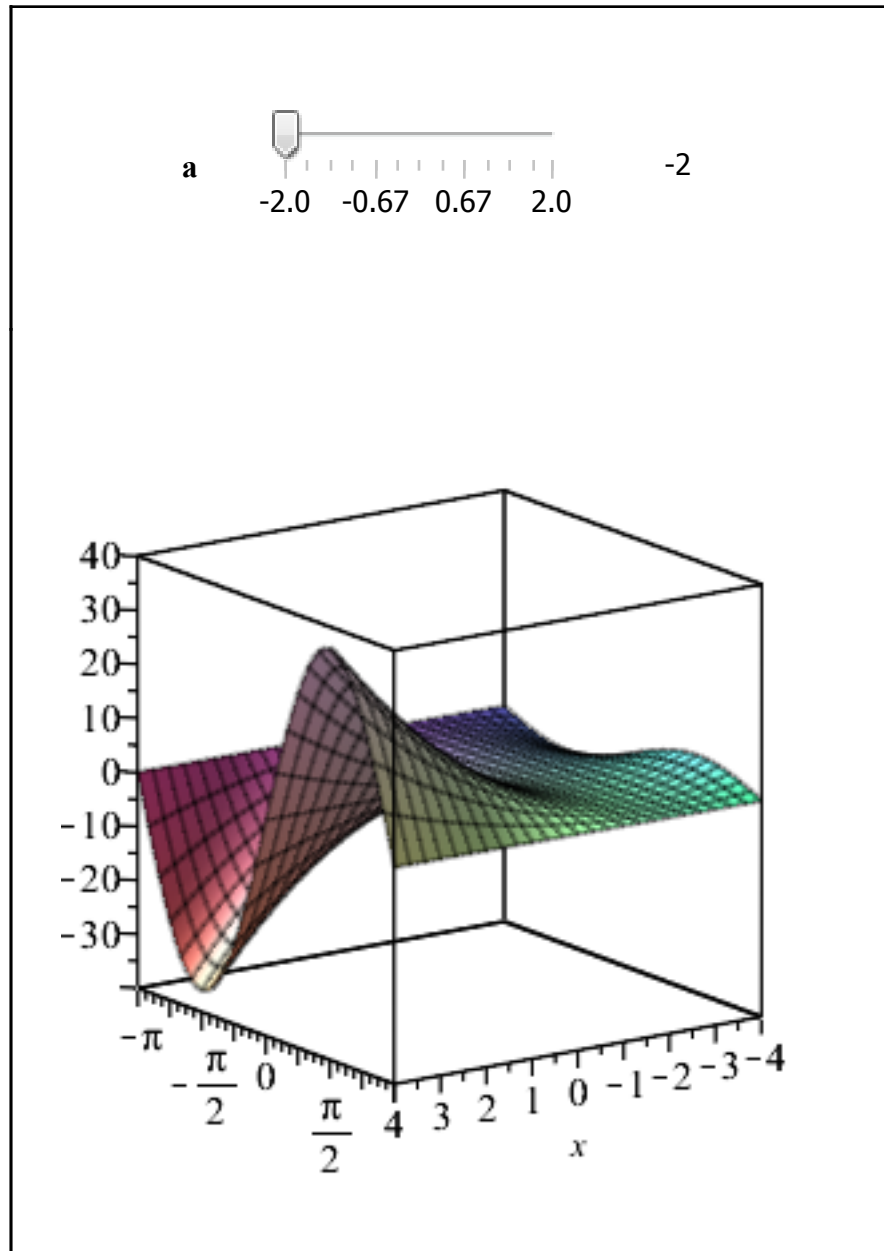
```
Explore( plot(A·sin(x), x=-π..π, view=-4..4), parameters = [[A=-4.0..4.0, tooltip  
= "vertical scaling factor"]])
```



The component identity and the action code for the controllers cannot be specified in this way, and are set internally by the *Explore* command.

## ▼ Optional Controller placement on top

*Explore*( *plot3d*(  $(x - a)^2 \cdot \sin(y)$ ,  $x = -4..4$ ,  $y = -\pi..pi$ ,  $view = -40..40$ ),  $a = -2.0..2.0$ , *placement = top* )



## ▼ Recording plots

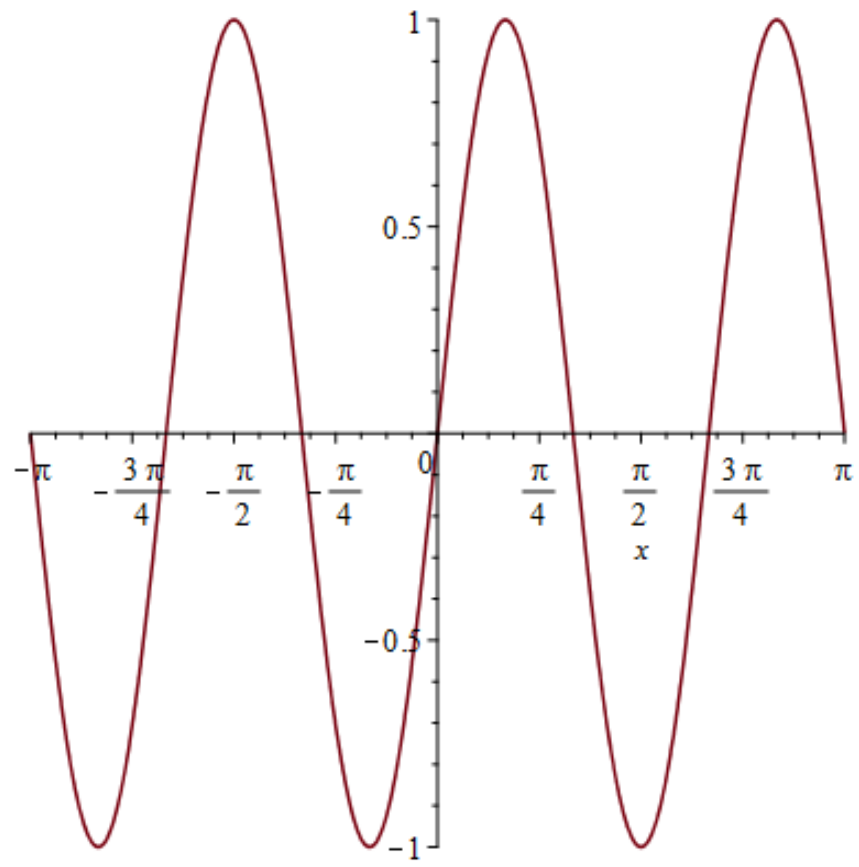
The new *recordto* and *autorecord* options provide the ability to assign plots generated by the exploration into a named [table](#). The entries of the table can then be played in sequence as an animation outside of the exploration using the [animation toolbar](#).

The recording state can be toggled on or off, or the name of the table can be unassigned, using two new buttons which appear in the exploration whenever the *recordto* option is supplied. The round button toggles recording on an off. The square button unassigns the name to which the table is assigned.

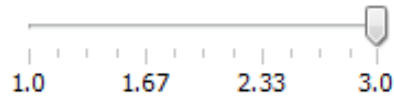
The saved plots are indexed in the table in the order they are generated.

None of the Explore parameters have to be specified with the *animate* option, in order to record. Manually adjusting any of the controllers which causes a new plot to be generated will also cause that plot to be saved if the record button is toggled on (solid red).

```
Explore( plot(sin(a*x), x=-Pi..Pi), a = 1.0 ..3.0, recordto ='myanim', autorecord, autorun,  
        numframes = 10)
```



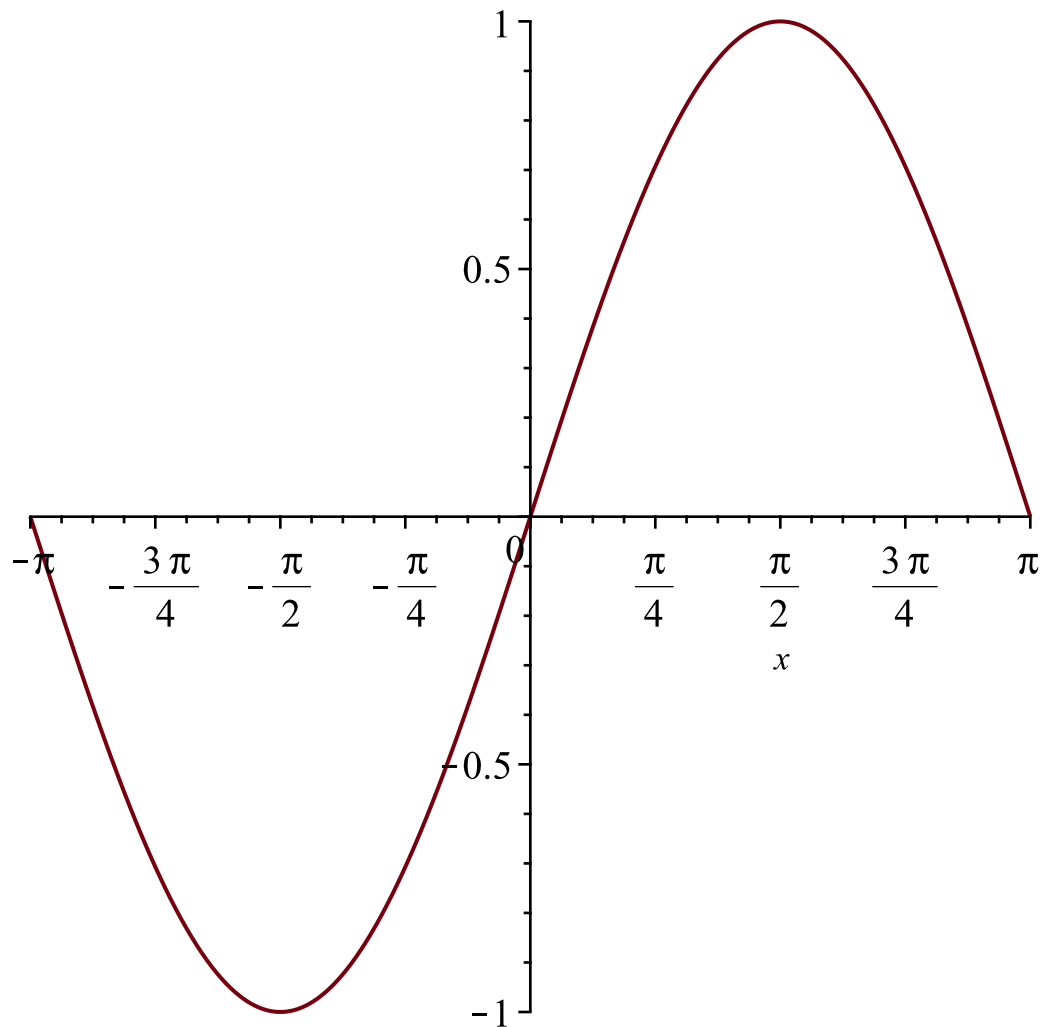
**a**



3



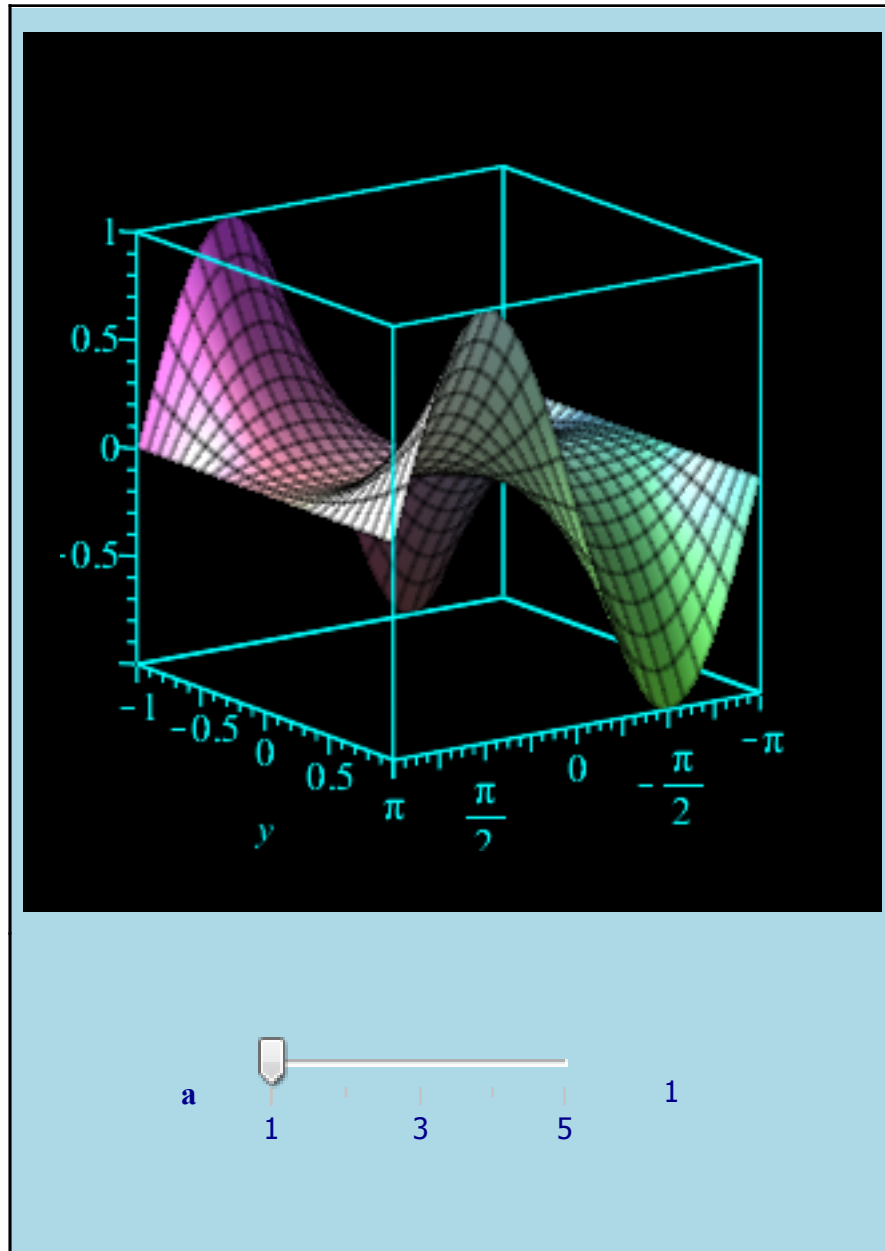
```
plots:-display(convert(myanim, list) [ ], insequence)
```



## ▼ Color customization

The new *fillcolor*, *fontcolor*, and *componentbackground* options provide the ability to customize the colors in the exploration. The values of these options can be any [named color](#) known to the plot command, or a list of three positive integers representing a color in the RGB color space.

```
Explore( plot3d( sin(a * x) * y^2, x = -Pi .. Pi, y = -1 .. 1, axis = [color = cyan], view = -1 .. 1 ),  
        a = 1 .. 5, componentbackground = "Black", fillcolor = "LightBlue", fontcolor  
        = "DarkBlue" )
```

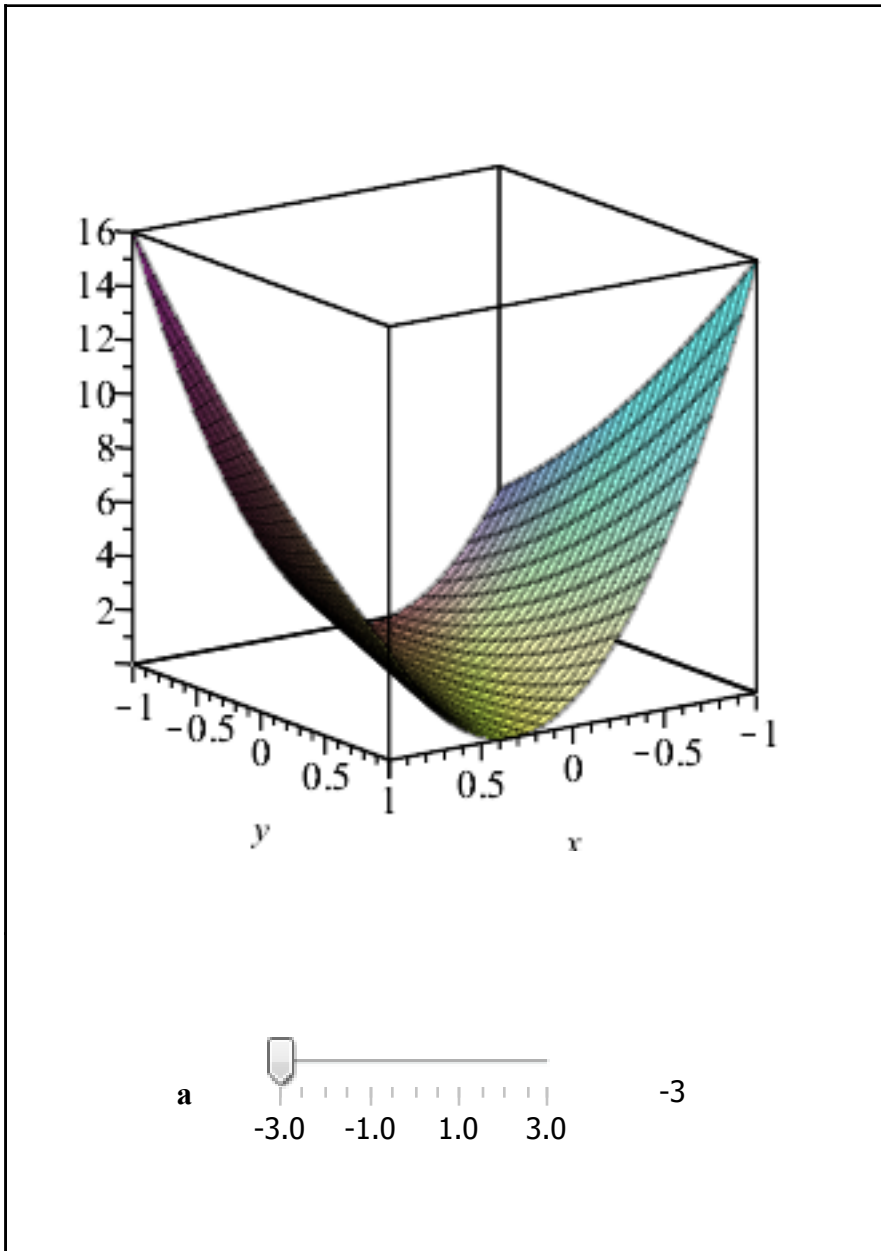


## ▼ Exploration alignment and width

The new option *tablealignment* specifies the horizontal placement in the document of the Table containing the exploration. By default, the exploration is centered horizontally.

*Explore*( *plot3d*(  $(a \cdot x + y)^2$ ,  $x = -1 .. 1$ ,  $y = -1 .. 1$ , *view* = 0 .. 16),  $a = -3.0 .. 3.0$ , *tablealignment* = left)

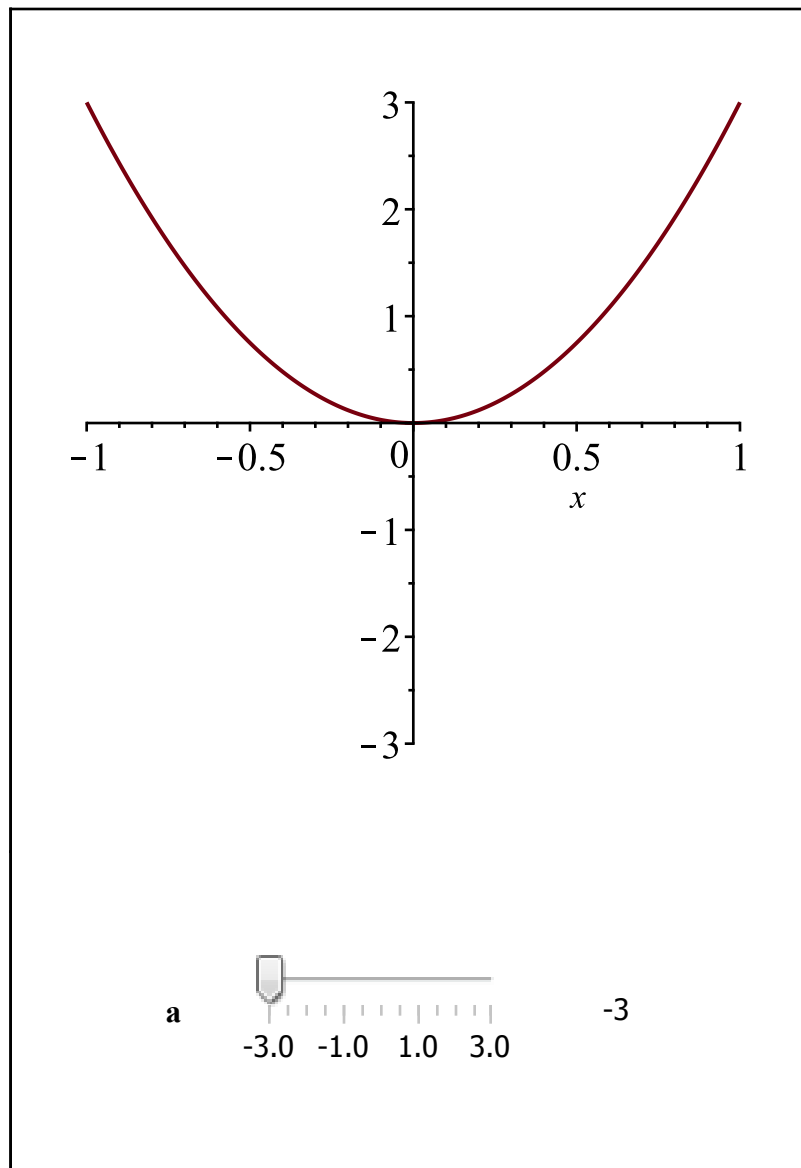




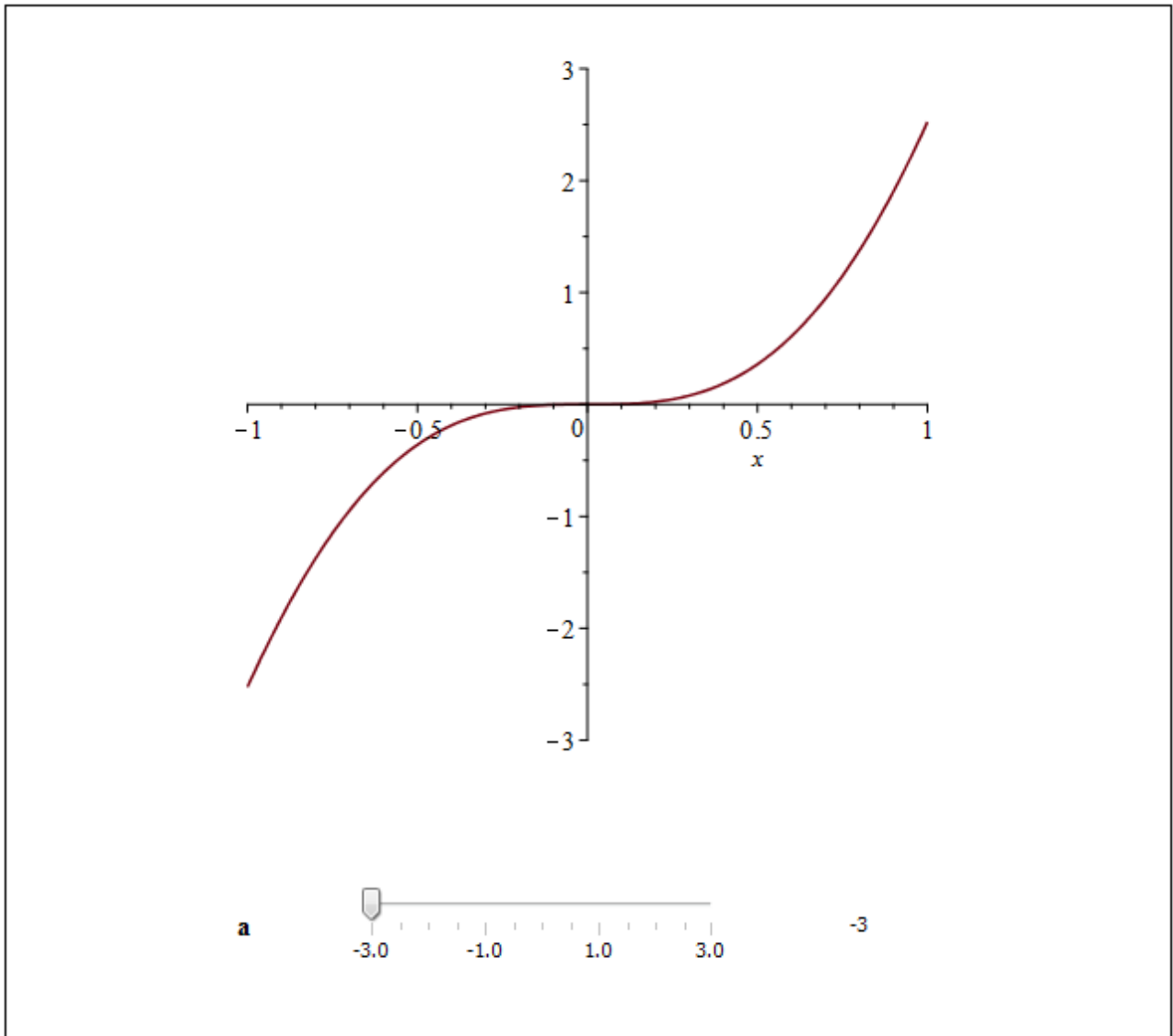
The new options *width* and *widthmode* specify the horizontal size of the exploration in the document.

The *widthmode* option denotes how the *width* option is interpreted. With *widthmode = pixels* the *width* option specifies the number of pixels wide the exploration is, at default 100% zoom level. In this case the exploration Table will not automatically resize horizontally when the document's window is manually resized using the mouse pointer. With *widthmode = percentage* the *width* option specifies the percentage of the current document window's width that the exploration will occupy horizontally. In this case the exploration's Table will automatically resize when the document's window is manually resized using the mouse pointer.

*Explore( plot(  $-a \cdot x^2$ ,  $x=-1 \dots 1$ ,  $view=-3 \dots 3$ ),  $a=-3.0 \dots 3.0$ , **widthmode = pixels**, **width = 300** )*



*Explore( plot(  $-a \cdot x^2 \cdot \sin(x)$ ,  $x=-1 \dots 1$ ,  $view=-3 \dots 3$ ),  $a=-3.0 \dots 3.0$ , **widthmode = percentage**, **width = 40** )*

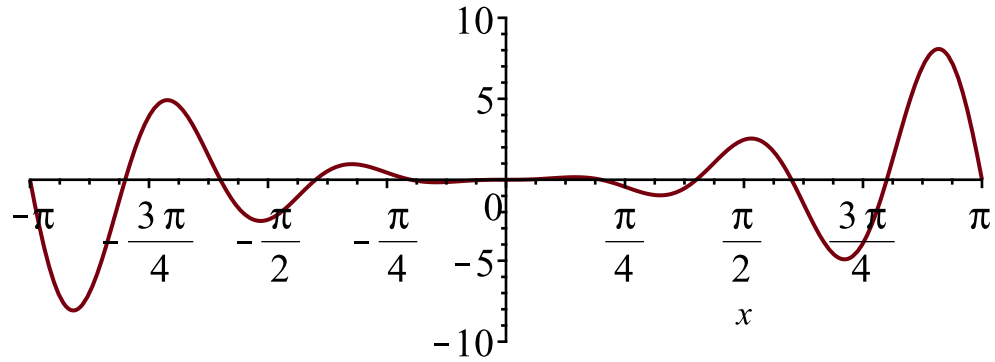
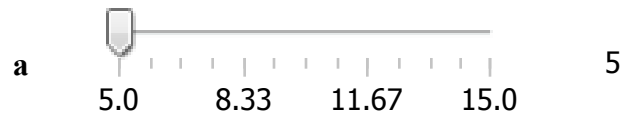


## ▼ Table and component borders

Several qualities of the borders in the exploration can be specified in Maple 2016.

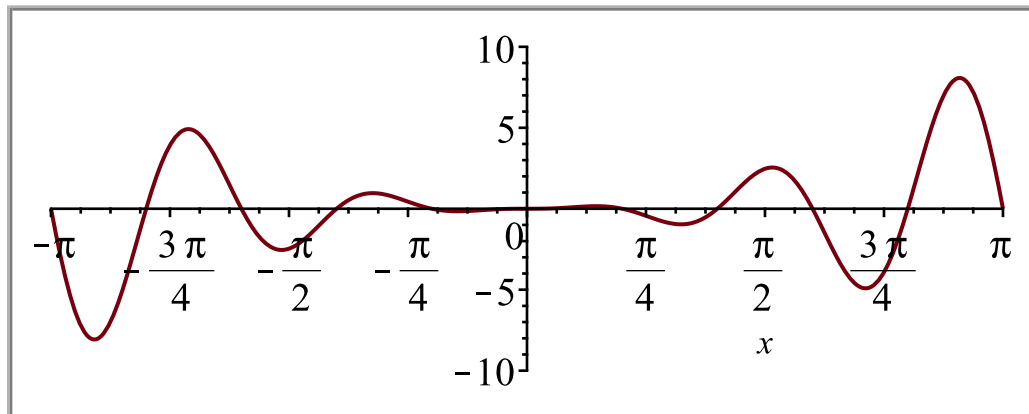
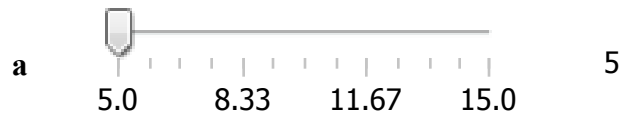
The exterior border of the outermost Table can be displayed, or not, using the new *tableborders* option.

```
Explore( plot(sin(a * x) * x^2, x=-Pi..Pi, view=-10..10, size = [500, 200]), a = 5.0..15.0,
         placement = top, tableborders = false )
```



The borders surrounding the [PlotComponent](#) (or [MathContainer](#)) can be displayed, or not, using the new *componentborders* option.

```
Explore( plot(sin(a * x) * x^2, x=-Pi..Pi, view=-10..10, size=[500, 200]), a=5.0..15.0,
         placement=top, tableborders=false, componentborders=true )
```



Interior borders of a Table that are not displayed have three modes of behavior when the mouse cursor temporarily hovers over the Table. They can be revealed in light gray by respecting the property of the worksheet, never revealed, or always revealed. These choices can now be specified using the new *showhiddenborders* option. The default for an exploration embedded into the current worksheet is to never reveal these interior Table borders.

When the mouse cursor hovers over the Tables of the following exploration the interior borders will be revealed in light gray.

*Explore( plot(sin(a \* x) \* x^2, x=-Pi..Pi, view=-10..10, size = [500, 200]), a = 5.0..15.0, showhiddenborders = always)*

