

# plot3d

three-dimensional plotting

---

[Calling Sequence](#)

[Parameters](#)

[Description](#)

---

[Examples](#)

[Compatibility](#)

## Calling Sequence

`plot3d(expr, x=a..b, y=c..d, opts)`

`plot3d(f, a..b, c..d, opts)`

`plot3d([exprf, exprg, exprh], s=a..b, t=c..d, opts)`

`plot3d([f, g, h], a..b, c..d, opts)`

## Parameters

<code>expr</code>	- expression in <b>x</b> and <b>y</b>
<code>f, g, h</code>	- procedures or operators
<code>exprf, exprg, exprh</code>	- expressions in <b>s</b> and <b>t</b>
<code>a, b</code>	- real constants, procedures, or expressions in <b>y</b>
<code>c, d</code>	- real constants, procedures, or expressions in <b>x</b>
<code>x, y, s, t</code>	- names
<code>opts</code>	- (optional) equations of the form <b>option=value</b> where <b>option</b> is described in <a href="#">plot3d/option</a>

## Description

- The **plot3d** command computes the plot of a three-dimensional surface. The first two calling sequences describe surface plots

## See Also

[coords](#)

[plot](#)

[plot/colormes](#)

[plot/colourscheme](#)

[plot/computation](#)

[plot/device](#)

[plot/interface](#)

[plot3d/colorfunc](#)

[plot3d/coords](#)

[plot3d/option](#)

[plot3d/structure](#)

[plots](#)

[plotsetup](#)

[Plotting Guide](#)

[plottools](#)

[Using the Interactive](#)

in Cartesian coordinates, while the second two describe parametric surface plots.

Other plotting facilities include the [plot](#) command for 2-D plotting, the [plots](#) package for specialized plots and the [plottools](#) package for plotting objects.

For further resources for plotting, and a pictorial listing of the available types of plots, see the [Plotting Guide](#). **Note** that this guide is only available in the Standard interface.

- Maple includes the **Interactive Plot Builder**, which provides a point-and-click interface to the plotting functionality including two and three-dimensional plots, animations, and interactive plots with sliders. To launch the **Plot Builder**, run the **plots [interactive]** command. You can also launch the **Plot Builder** in the Standard Worksheet from the **Tools** menu. Select **Assistants**, and then **Plot Builder**. For more information, see [Using the Interactive Plot Builder](#).
- In the first calling sequence, **plot3d(expr, x=a..b, y=c..d)**, the expression **expr** must be an expression in the names **x** and **y**. The range **a..b** must evaluate to real constants and the range **c..d** must either evaluate to real constants or be expressions in **x**. Alternatively, the range **c..d** must evaluate to real constants and the range **a..b** must either evaluate to real constants or be expressions in **y**. These specify the range over which **expr** is plotted.
- In the second calling sequence, **plot3d(f, a..b, c..d), f** must be a procedure or operator that takes two arguments. Operator notation must be used, that is, the procedure name is given without parameters specified, and the ranges must be given simply in the form **a..b**, rather than as an equation. At least one of the ranges must have arguments evaluating to real constants; the other range may have arguments evaluating to real constants or be procedures of one variable.
- A parametric surface can be defined by three expressions **exprf, exprg, exprh** in two variables. In the third calling sequence, **plot3d([exprf, exprg, exprh], s=a..b, t=c..d), exprf, exprg, and exprh** must be expressions in the names **s** and **t**. In the fourth calling sequence, **plot3d([f, g, h], a..b, c..d), f, g, and h** must be procedures or operators taking at most two arguments. As with the second calling sequence, operator notation must be used.
- With any of these calling sequences, the range arguments may be omitted. In that case, the **plot3d** command assumes default ranges of -10 to 10, or  $-2\pi$  to  $2\pi$  in the case where a trigonometric function is detected. The first argument **f** or **expr** can also be omitted or set to the empty list **[]**, in which case an empty plot is created.

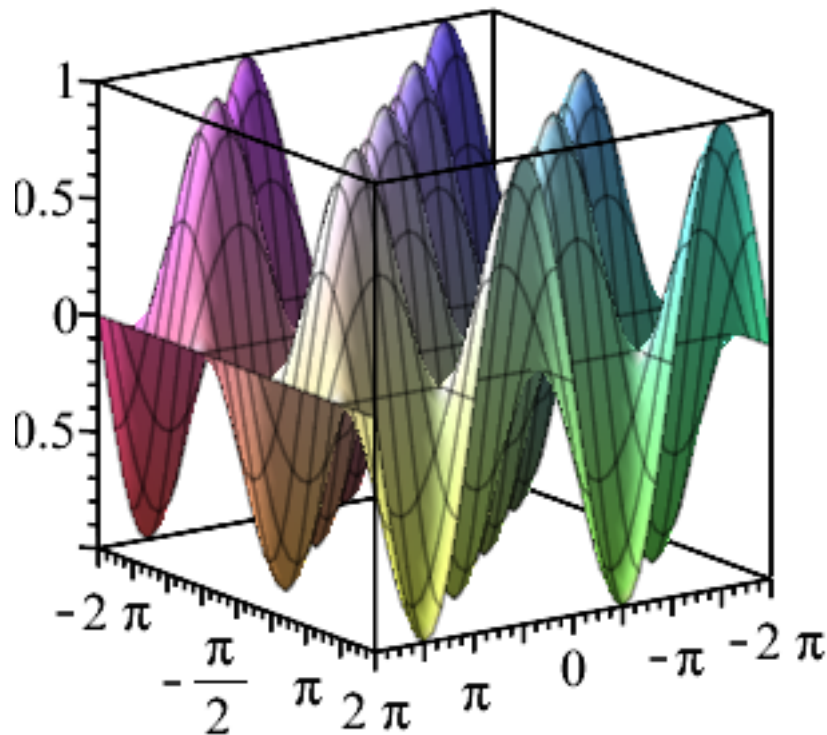
- Any additional arguments are interpreted as options, which are specified as equations of the form **option = value**. For example, the option **grid = [m, n]** where **m** and **n** are positive integers, specifies that the plot is to be constructed on an **m** by **n** grid at equally spaced points in the ranges **a..b** and **c..d** respectively. By default, a 49 by 49 grid is used and 2401 points are generated. Other options include specification of alternate coordinate systems and rendering styles. For more information, see [plot3d/options](#).
- If the first argument in any of the calling sequences is a set or list of surfaces, then the surfaces are plotted together. If a list is provided, then particular option values can also be given as lists, with elements corresponding to elements of the list of surfaces. The options that can take lists as values are: **color**, **coords**, **grid**, **linestyle**, **numpoints**, **shading**, **style**, **symbol**, **symbolsize**, **thickness**, and **transparency**. A list of three algebraic expressions or procedures is always interpreted as a parametric plot. To specify a list of three distinct plots, the option **plotlist=true** (or simply **plotlist**) must be provided.
- Plots in alternative coordinate systems, such as spherical and cylindrical systems, can be generated by using the **coords** option. For more information, see the examples below or the [plot3d/coords](#) help page.
- There are several ways to color 3-D surfaces created by the **plot3d** command. See the [plot/color](#), [plot3d/colorfunc](#) and [plot/colorscheme](#) help pages for more information.
- When **plot3d** evaluates its arguments, any errors generated during the evaluation are suppressed. A symptom that something has gone wrong with the evaluation of your expression is a resulting empty plot.
- Help pages describing plotting commands and interactive plotting features are written with the assumption that you are using the Standard Worksheet interface. If you are using a different interface, see [plot/interface](#).
- An output device may be specified using the [plotsetup](#) command. See [plot/device](#) for a list of supported devices.
- The result of a call to **plot3d** is a **PLOT3D** data structure containing enough information to render the plot. The user can assign a **PLOT3D** value to a variable, save it in a file, then read it in for redisplay. For more information, see [plot3d/structure](#).
- All plotted expressions are evaluated numerically, that is, as floating point expressions, rather than symbolically. For more information about the computational environment used by the **plot3d** function, see [plot/computation](#).

## Examples

### Default ranges in 3-D plots

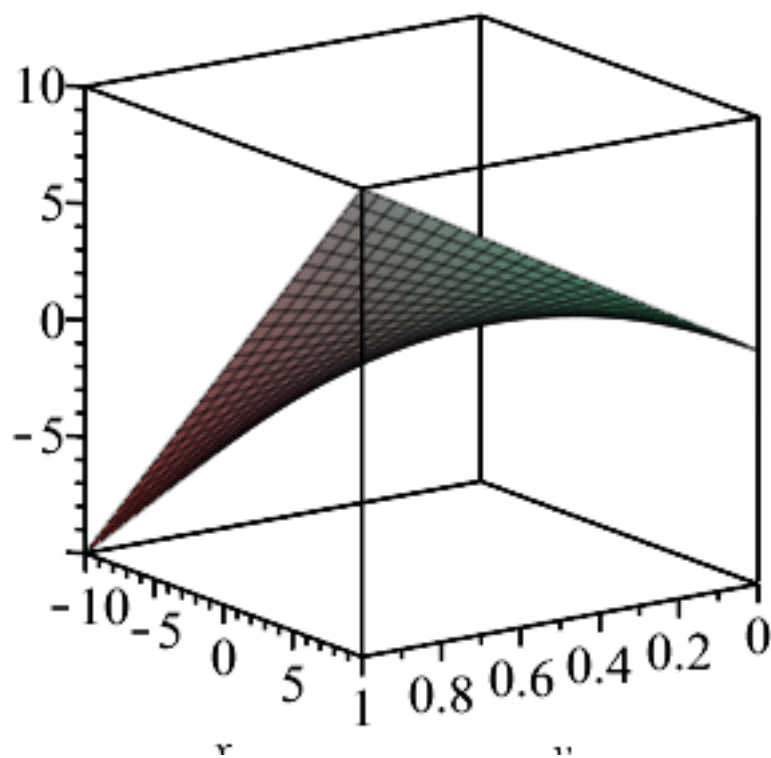
For trigonometric functions, default ranges of  $-2\pi \dots 2\pi$  is used.

```
> plot3d(sin(x) * cos(y));
```



Default ranges of  $-10 \dots 10$  are used when the range arguments are not provided.

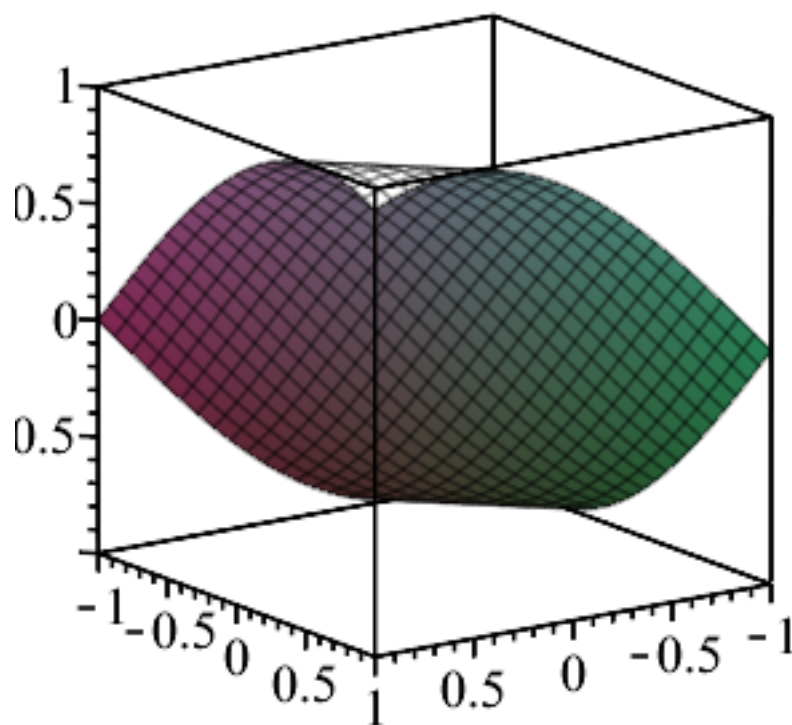
```
> plot3d(x*y, y = 0..1);
```



Generating three-dimensional surfaces using expressions or procedures

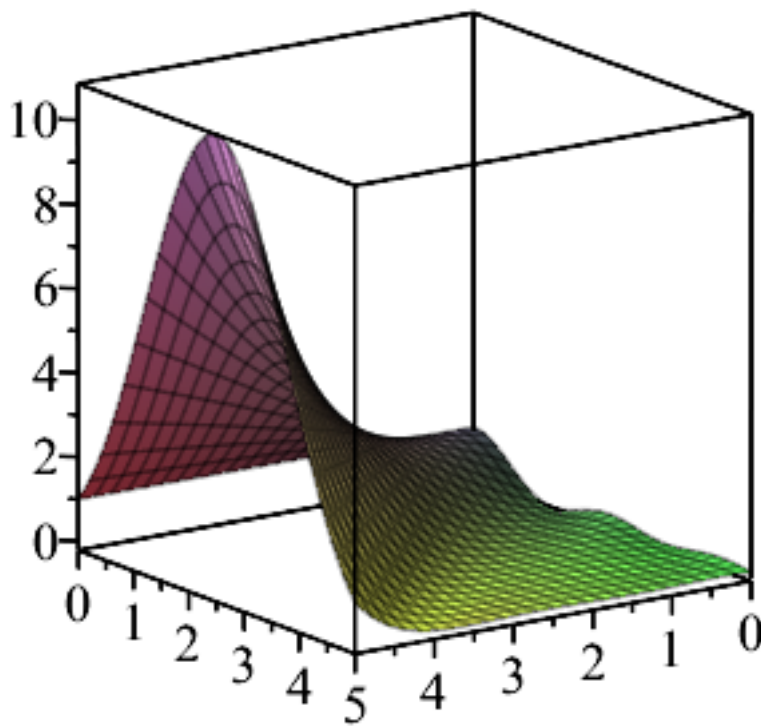
When plotting an expression in two variables, the range for each variable must be provided in the form of **x=a..b** and **y=c..d**, where **x** and **y** are the variables used in the expression.

```
> plot3d(sin(x+y), x=-1..1, y=-1..1);
```



To plot a procedure that is a function of two variables (for example, [binomial](#)), give the procedure in operator notation (that is, without parameters). Also, the ranges must be given in the form **a..b**, not as equations.

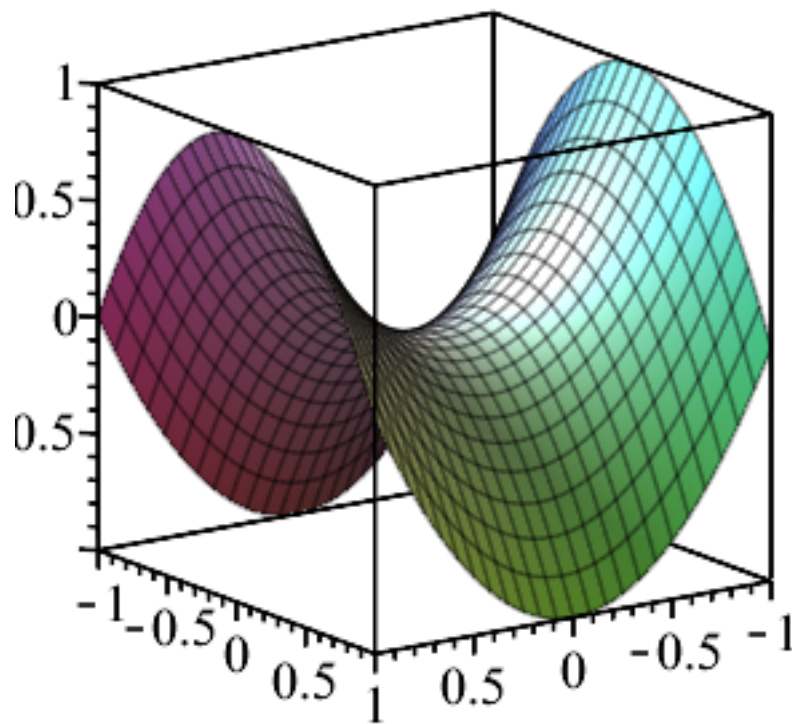
```
> plot3d(binomial, 0..5, 0..5);
```



The following example defines a functional operator of two variables and plots it.

```
> f:=(x,y)->x^2-y^2:
```

```
> plot3d(f,-1..1,-1..1);
```

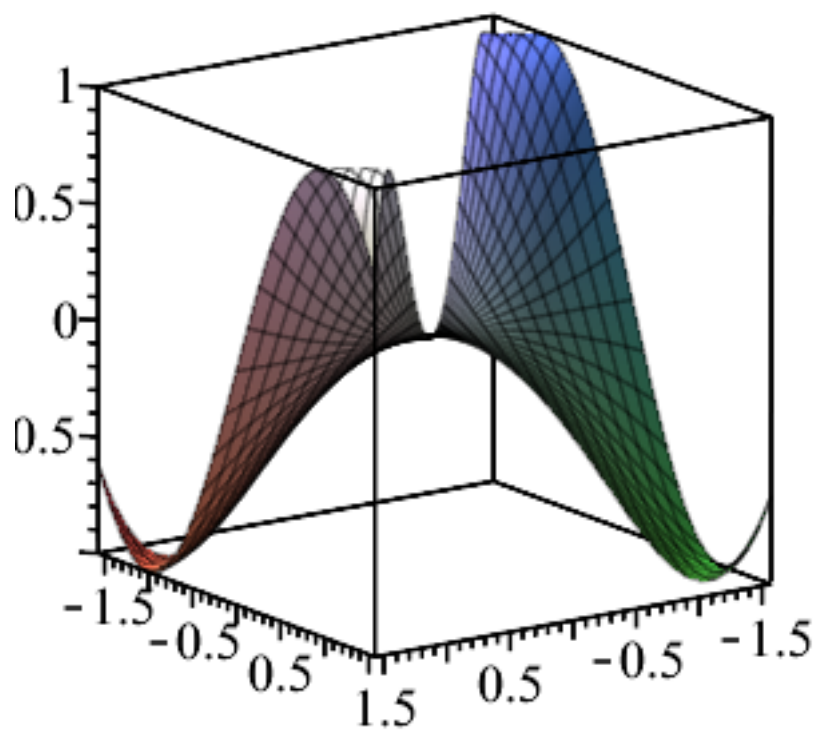


Using variable expressions or procedures in a range

You can specify a variable expression in one or both endpoints of a range as long as the other range contains real constants for both its endpoints. In the following example, the endpoints for **x** are given as real constants while the endpoints for **y** are expressions in **x**.

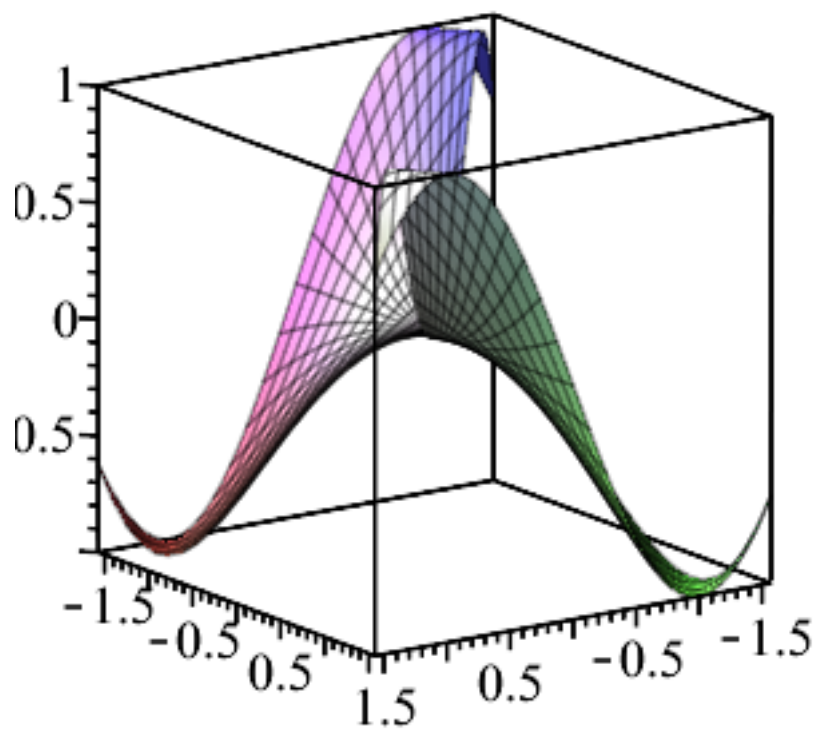
```
> plot3d(sin(x*y), x=-Pi/2..Pi/2, y=-x..x);
```





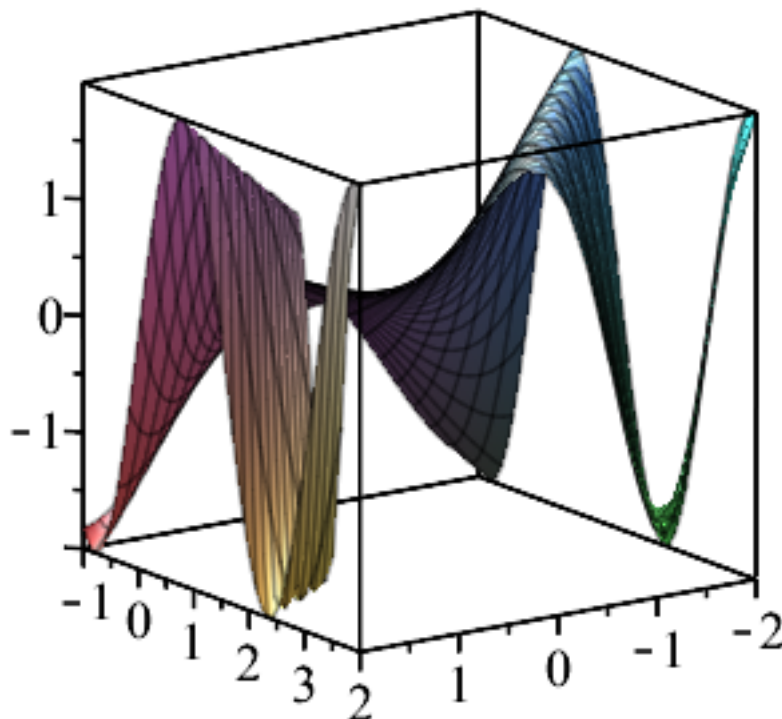
Alternatively, the endpoints for **y** can be specified as real constants and the endpoints for **x** given as expressions in **y**.

```
> plot3d(sin(x*y),x=-y..y,y=-Pi/2..Pi/2);
```



The following example uses one procedure for the surface to be plotted and another procedure in the range for the second variable.

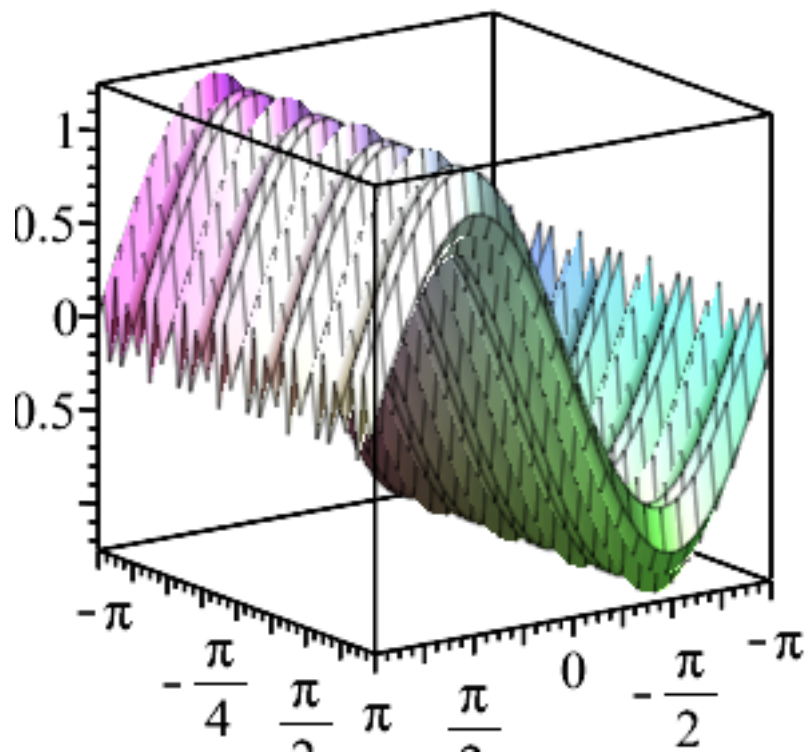
```
> p:= proc(x,y) if x^2 < y then cos(x*y) else  
  x*sin(x*y) end if end proc:  
  
> h:= proc(x) x^2 end proc:  
  
> plot3d(p, -2..2, -1..h);
```



### Drawing smoother curves with the grid option

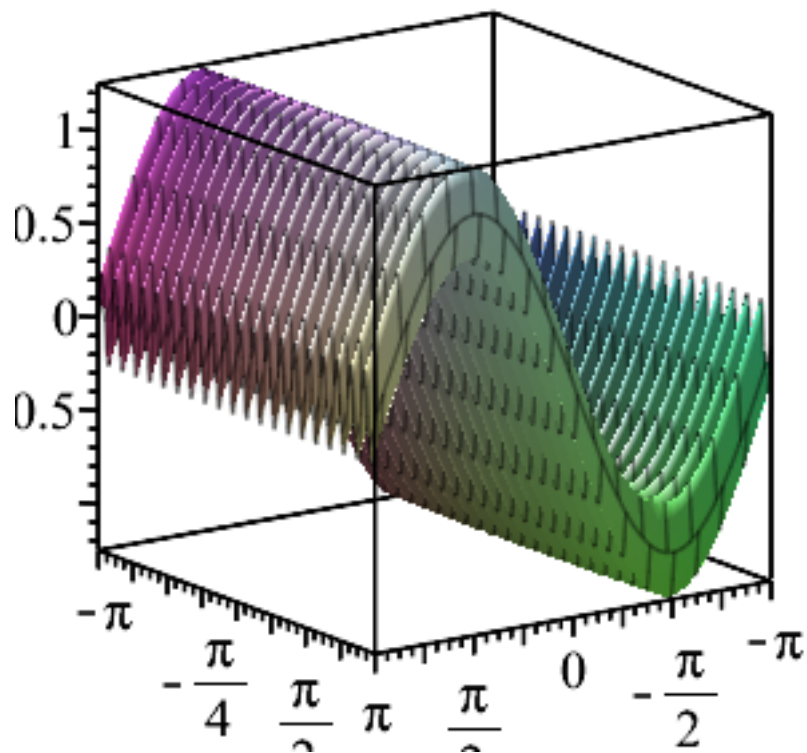
The default 49 by 49 grid may be too coarse for a plot, especially if the surface changes rapidly over the plotting range.

```
> plot3d(sin(x) + sin(20*y)/4, x=-Pi..Pi, y=-Pi..  
  Pi);
```



Use the **grid = [m,n]** option to specify a finer grid and show more detail in your plot.

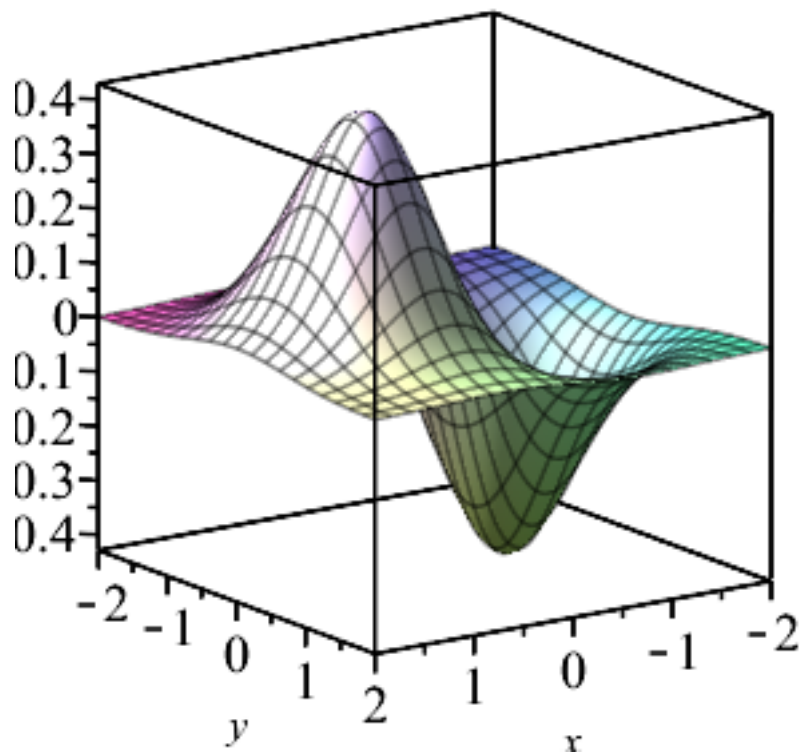
```
> plot3d(sin(x) + sin(20*y)/4,x=-Pi..Pi,y=-Pi..
    Pi,grid=[50,200]);
```



### Specifying the surface color

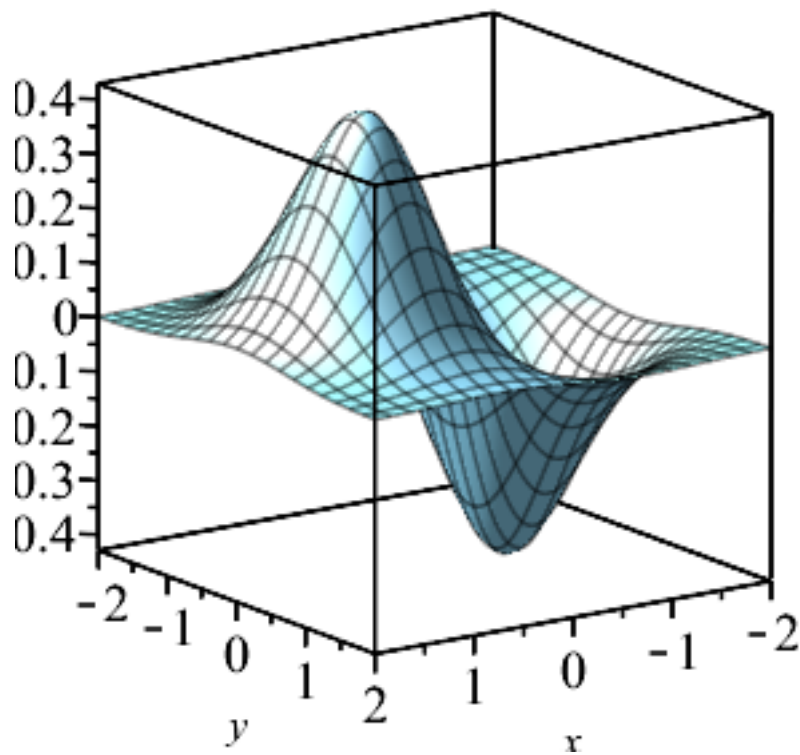
The following command shows the default color for a three-dimensional surface.

```
> plot3d(x*exp(-x^2-y^2), x=-2..2, y=-2..2,
  grid=[100,100]);
```



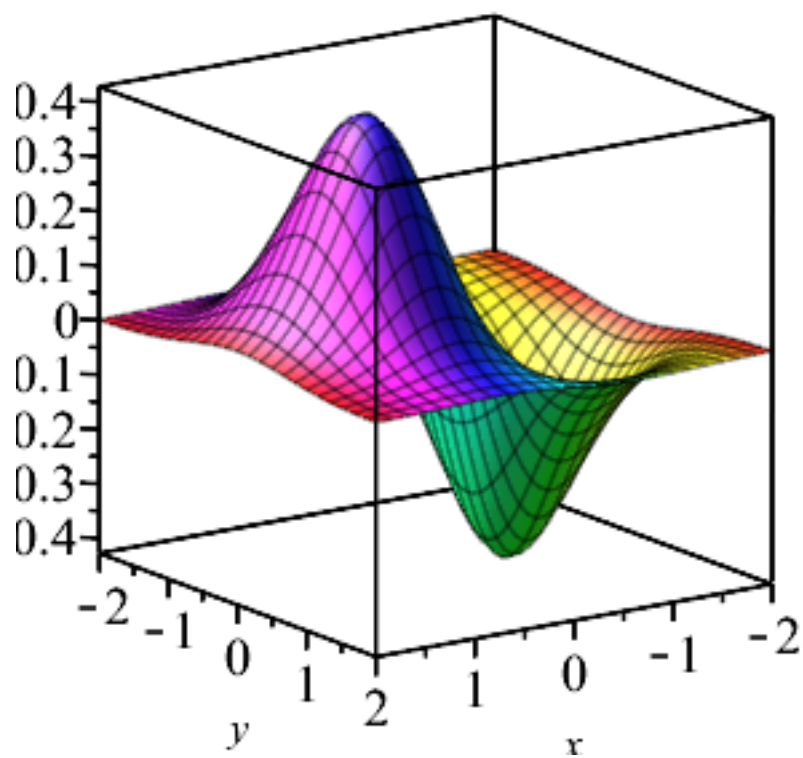
To set the color, use the **color=cname** option, where **cname** is one of the predefined [plot color names](#).

```
> plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2,grid=[100,100],color="SkyBlue");
```

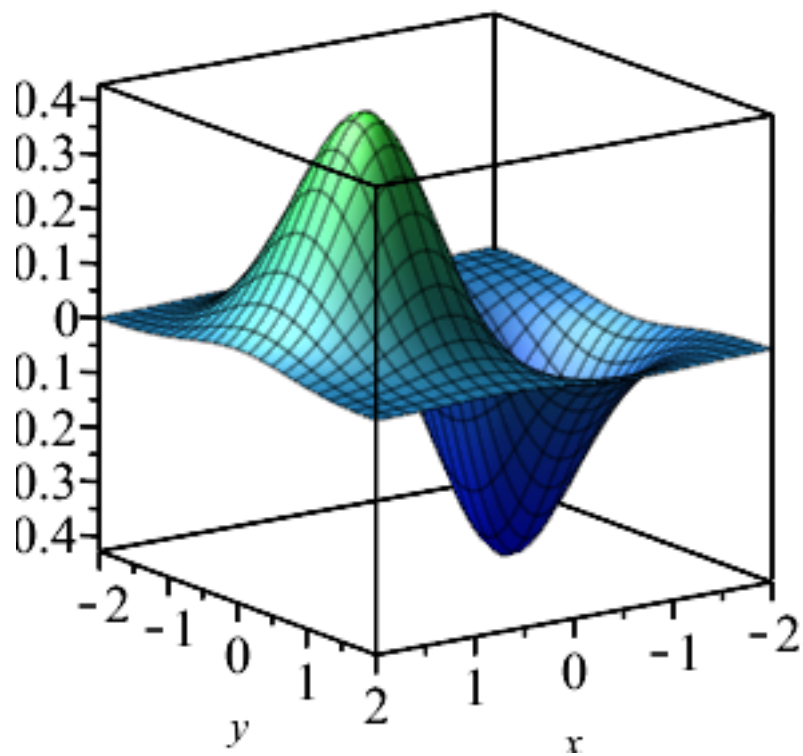


You can also set the **color** option to an expression or procedure, or use the **colourscheme** option to color by  $z$  values.

```
> plot3d(x*exp(-x^2-y^2), x=-2..2, y=-2..2,  
  color=x);
```



```
> plot3d(x*exp(-x^2-y^2), x=-2..2, y=-2..2,
  colorscheme=["Blue", "LimeGreen"]);
```

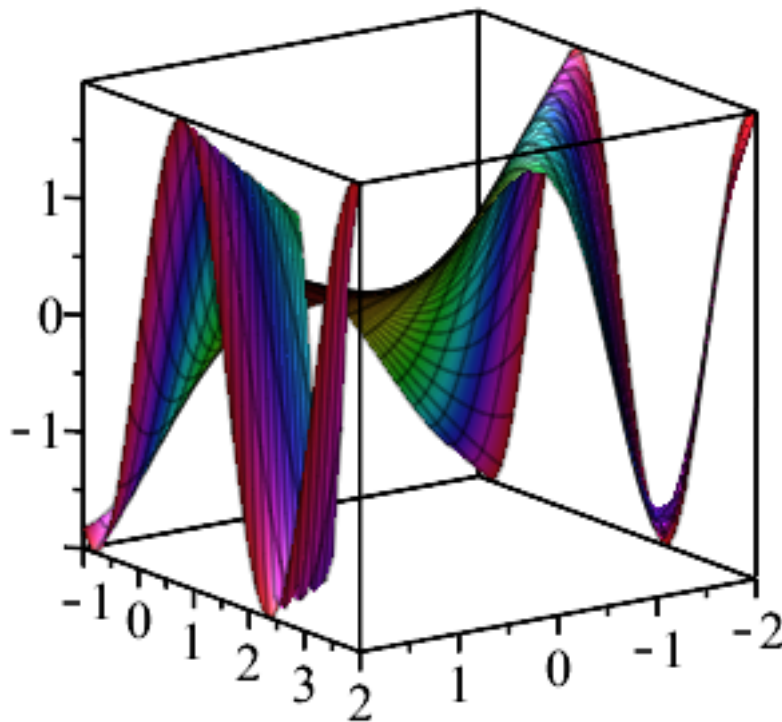


```
> p:= proc(x,y) if x^2 < y then cos(x*y) else
```

```

    x*sin(x*y) end if end proc:
> h:= proc(x) x^2 end proc:
> plot3d(p, -2..2, -1..h, color=h);

```



For more information on using a procedure or expression to color your surface, see [plot3d/colorfunc](#).

## Displaying multiple surfaces in one plot

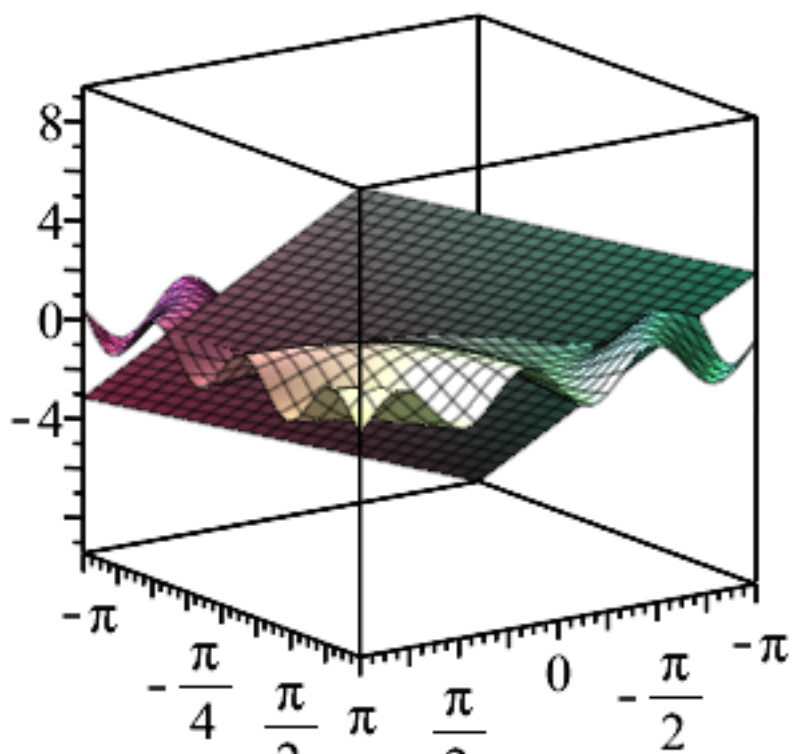
Use sets or lists to display more than one surface in the same three-dimensional plot.

```

> plot3d([sin(x*y), x + 2*y], x=-Pi..Pi, y=-Pi.
    .Pi);

```





```

> c1:= [cos(x)-2*cos(0.4*y), sin(x)-2*sin(0.4*
y), y]:

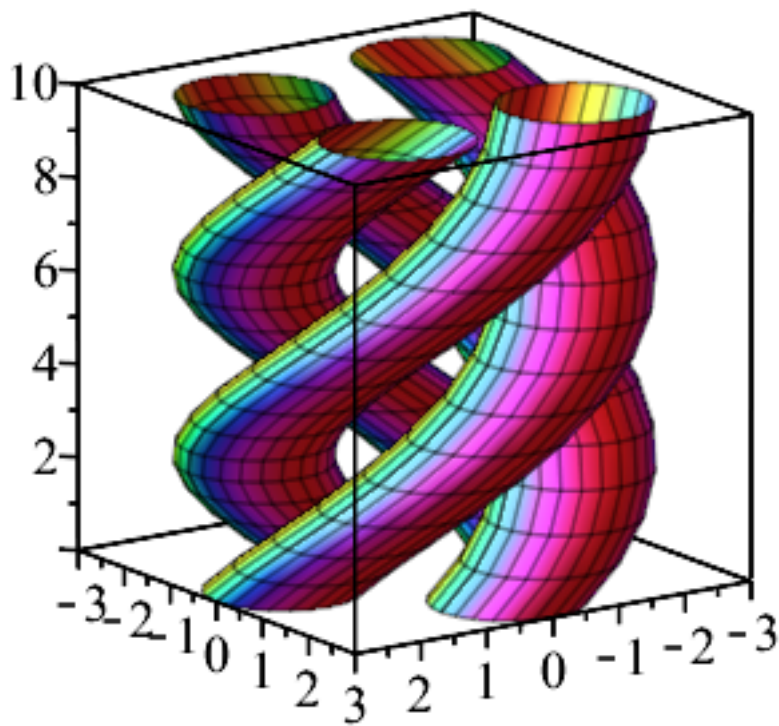
> c2:= [cos(x)+2*cos(0.4*y), sin(x)+2*sin(0.4*
y), y]:

> c3:= [cos(x)+2*sin(0.4*y), sin(x)-2*cos(0.4*
y), y]:

> c4:= [cos(x)-2*sin(0.4*y), sin(x)+2*cos(0.4*
y), y]:

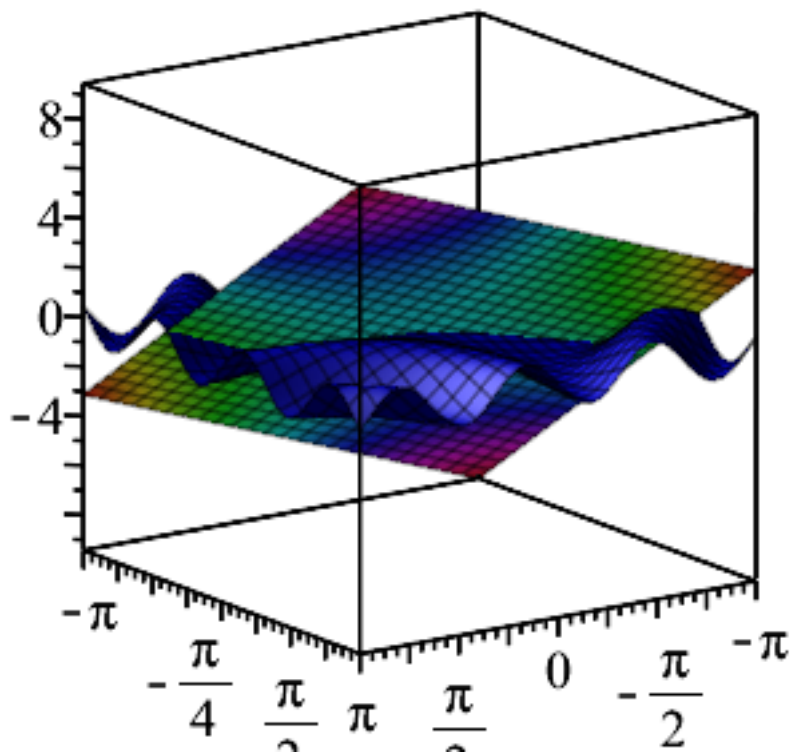
> plot3d({c1, c2, c3, c4}, x=0..2*Pi, y=0..10,
grid=[25,15], color=sin(x));

```



To specify a different color for each surface, set the **color** option to a list of colors or expressions.

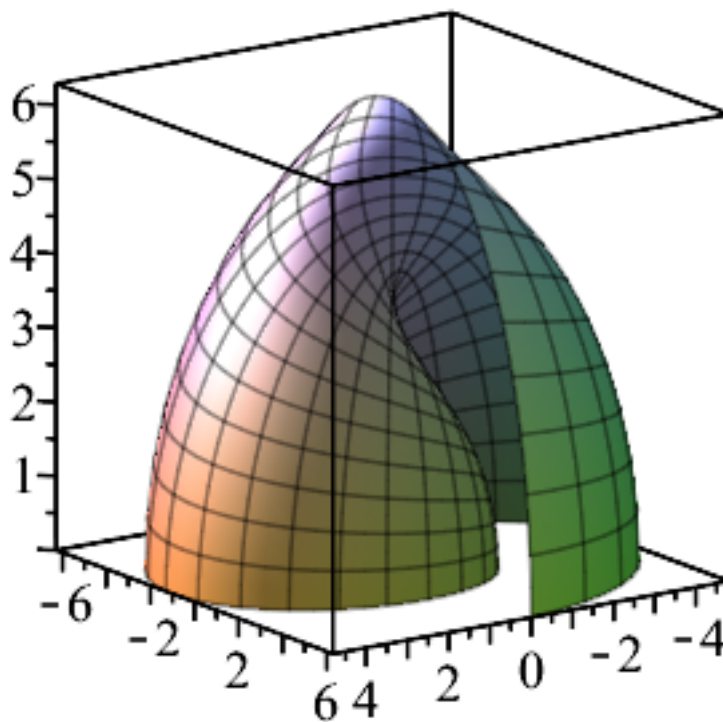
```
> plot3d([sin(x*y), x + 2*y], x=-Pi..Pi, y=-Pi.  
      .Pi, color=["Navy",x*y]);
```



### Plotting a parametric surface

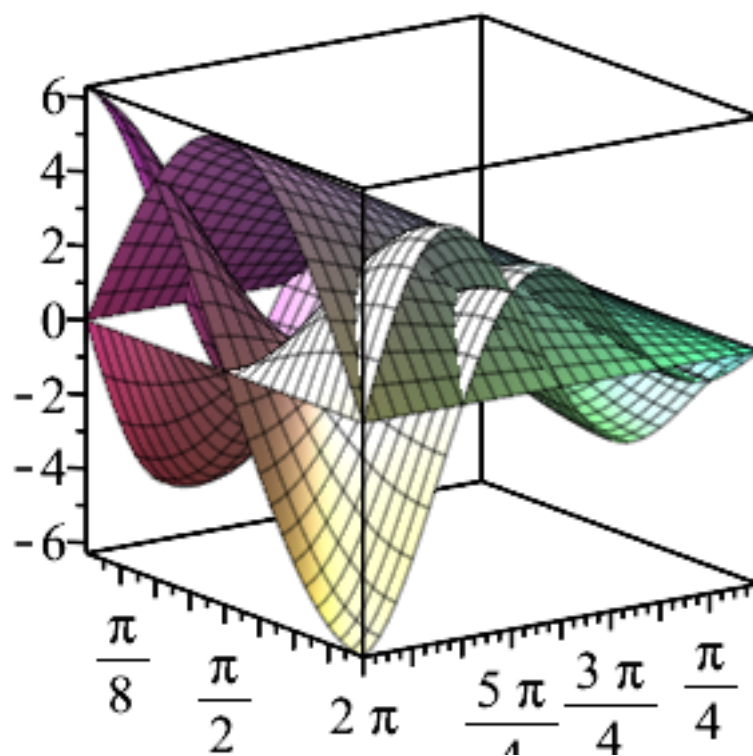
If the first argument is a list of three surfaces, **plot3d** produces a plot of the parametric surface.

```
> plot3d([x*sin(x)*cos(y), x*cos(x)*cos(y), x*
  sin(y)], x=0..2*Pi, y=0..Pi);
```

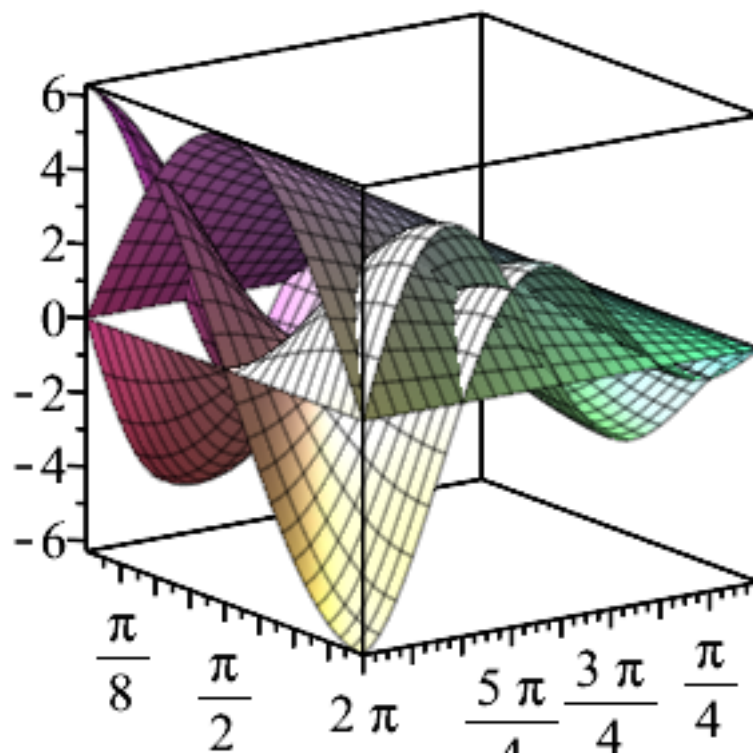


To not have a list of three surfaces interpreted as a parametric plot, either provide the **plotlist=true** option or use a set for the surfaces.

```
> plot3d([x*sin(x)*cos(y), x*cos(x)*cos(y), x*
  sin(y)], x=0..2*Pi, y=0..Pi, plotlist=true);
```



```
> plot3d({x*sin(x)*cos(y), x*cos(x)*cos(y), x*
sin(y)}, x=0..2*Pi, y=0..Pi);
```

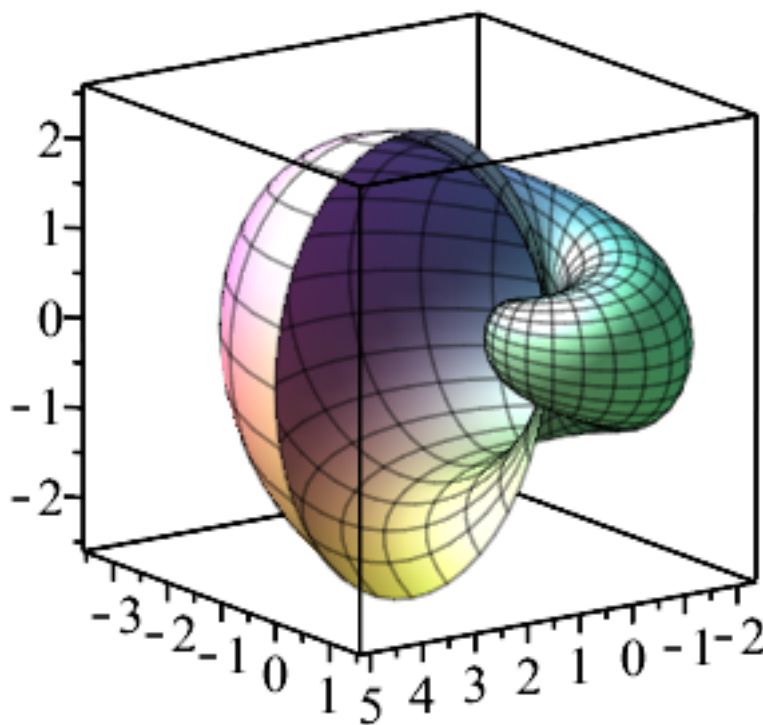


Using different coordinate systems

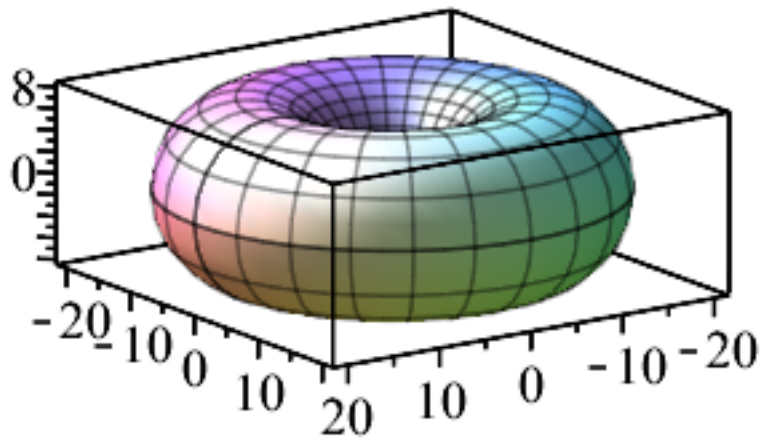
Use the **coords** option to specify a different coordinate system for your plot. See [plot3d/coords](#) for a list of the available coordinate systems and information on how they are interpreted by **plot3d**. The [coords](#) page gives a description for each of these coordinate systems.

The following three commands show plots using spherical and toroidal coordinates.

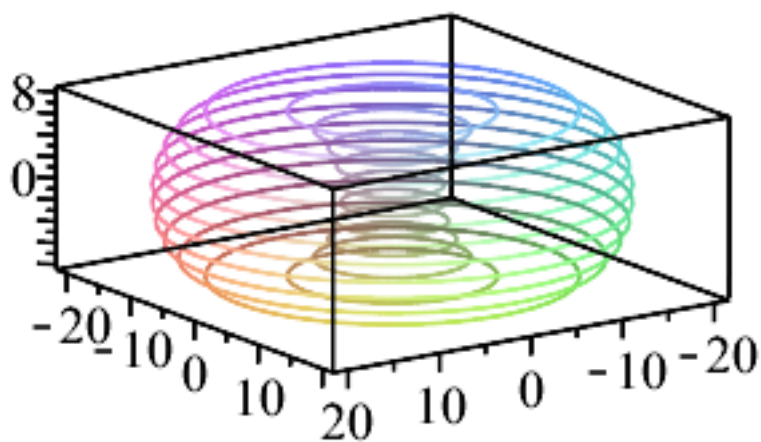
```
> plot3d((1.3)^x * sin(y), x=-1..2*Pi, y=0..Pi,  
  coords=spherical);
```



```
> plot3d([1,x,y], x=0..2*Pi, y=0..2*Pi, coords=  
  toroidal(10), scaling=constrained);
```

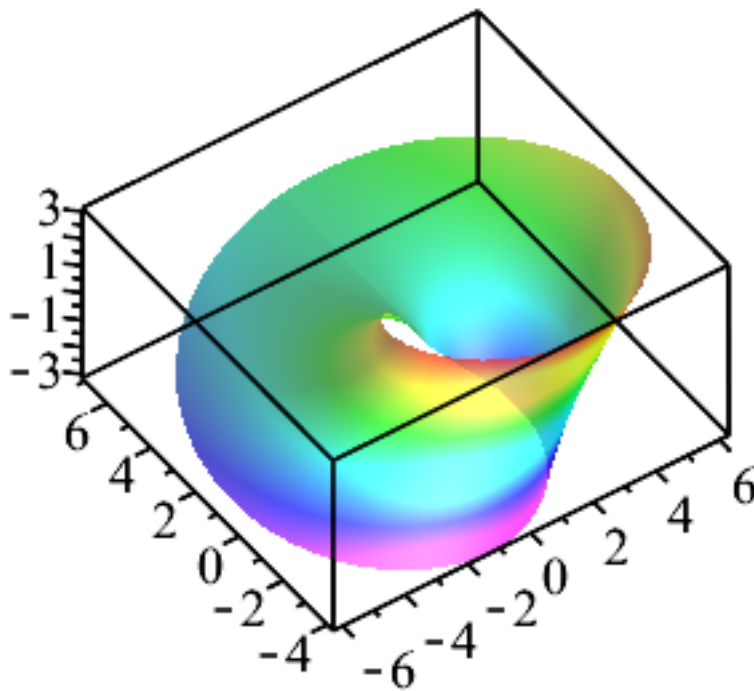


```
> plot3d([1,x,y], x=0..2*Pi, y=0..2*Pi, coords=  
toroidal(10), scaling=constrained, style=  
contour);
```



The following command generates the plot of the Möbius strip from the [Plotting Guide](#) using cylindrical coordinates.

```
> plot3d([4+x*cos(1/2*y), y, x*sin(1/2*y)], x=-  
  Pi..Pi, y=0..2*Pi, coords=cylindrical, style=  
  patchnogrid, grid=[60,60], orientation=[35,  
  135], lightmodel=light4, shading=zhue,  
  scaling=constrained, transparency=0.3);
```

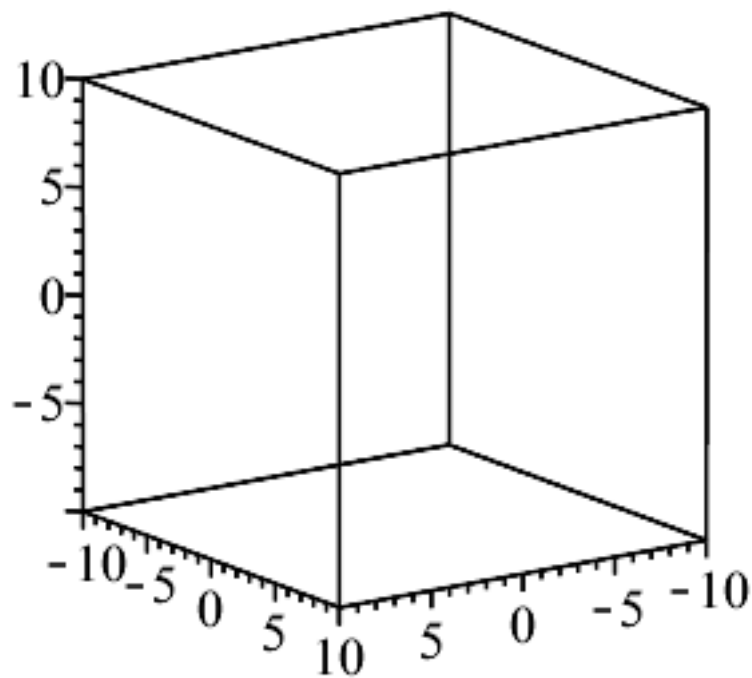


Generating empty plot

```
> plot3d(title="An empty plot");
```



## An empty plot



>

## Compatibility

- The **plot3d** command was updated in Maple 2015.