

# Maple: Plotting graphs

System Maple provides many possibilities for graphical representation of data and mathematical relations, using different modifications of function `plots` for plotting graphs. Among many other there are available also functions for interactive plotting and animations. One of the most frequently used from all other plot functions is the well-known function `plot` for plotting graph of function in one variable, characteristic for a wide variability of different options of plotting parameters and graph features.

For successful work with plot function of the system Maple it is necessary to understand two basic principles of their construction. First one is the fundamental difference between provided plot functions, as standard plot functions - standard plots and additional graphical functions - smart plots. Next important difference is the way, in which parameter settings and their modifications in separate plot functions are controlled, and how these influence plot of individual curves and surfaces.

## Standard plot functions

Standard plot functions of the system Maple are used for construction of detailed, complex, or special graphical representations of data or mathematical expressions. Several standard basic plot functions exist.

They contain few obligatory parameters and several optional ones with default setting values to specify attributes of the plot – thickness, style and colour of lines, to plot or to suppress drawing of plot axes and settings of their view, setting of the plot scale, size and type of symbols used in the graphics, etc.

Function `plot(f, h, v)` plots graph of function of one variable defined on certain interval. It consists of the following necessary parameters:

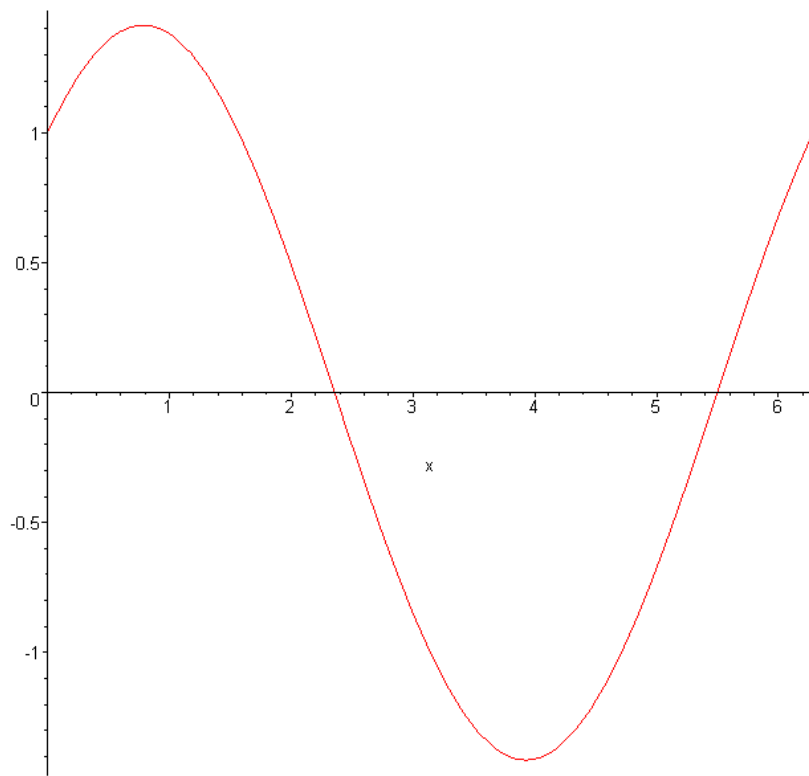
`f` – real function in  $x$ , graph of which will be plotted,

`h` – range on the horizontal axis

and an optional parameter

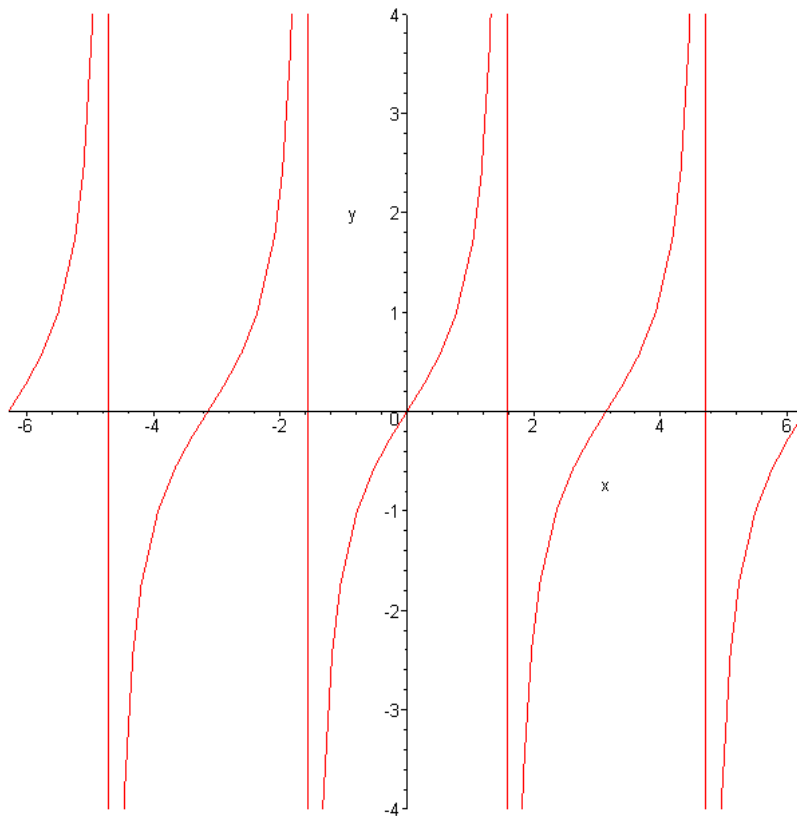
`v` – range on the vertical axis.

```
> plot(cos(x) + sin(x), x=0..2*Pi);
```



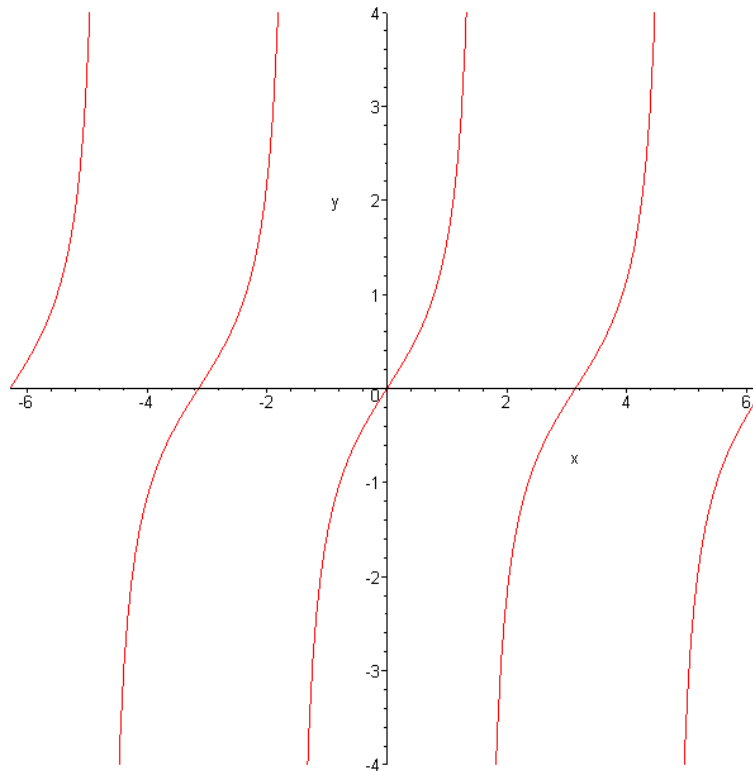
System Maple tries to plot graph of function as continuous curve.

> `plot(tan(x), x = -2*Pi..2*Pi, y = -4..4);`



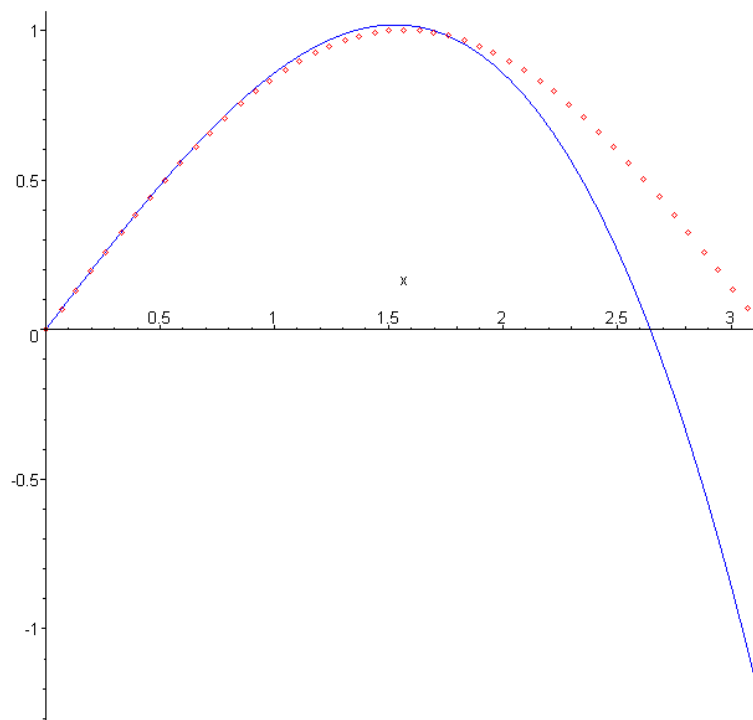
Parameter `discont = true` secures discontinuity of the plotted function graph on the chosen interval

```
> plot(tan(x), x = -2*Pi..2*Pi, y = -4..4, discont = true);
```



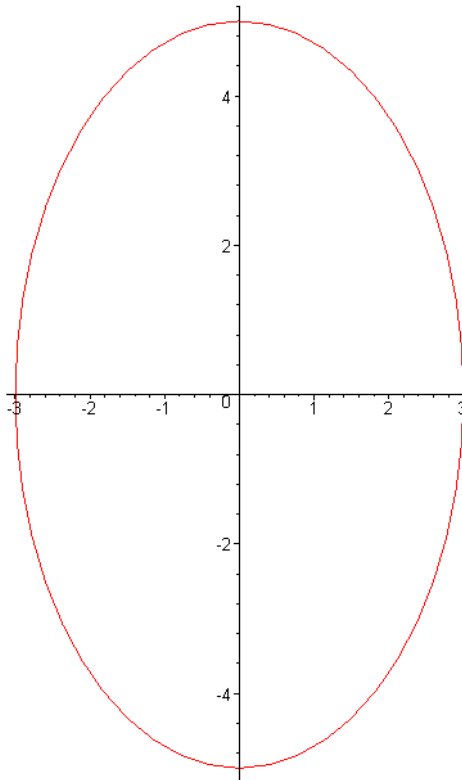
Graphs of two functions plotted in one graphic distinguished by colour and line style.

```
> plot([sin(x), x-x^3/7], x=0..3.14, color=[red,blue], style=[point,line]);
```



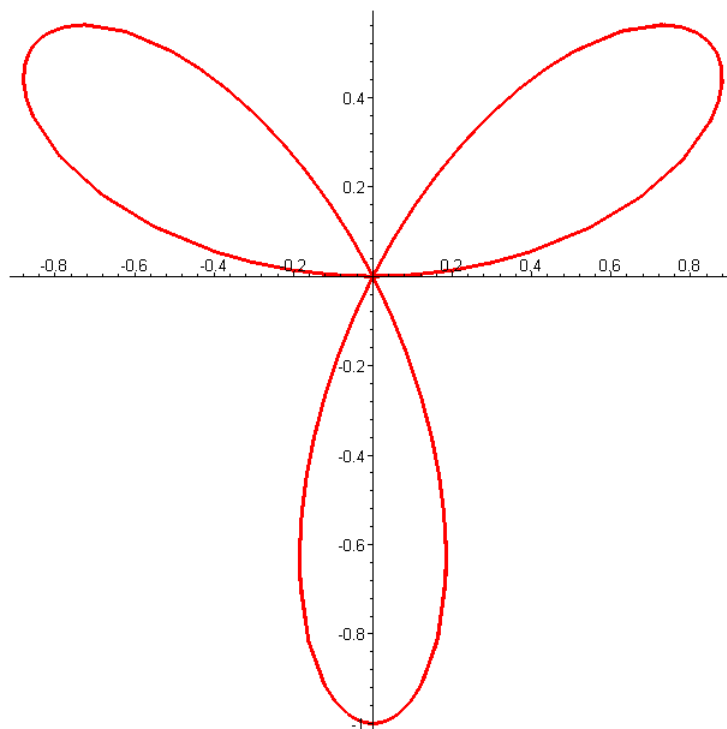
Plot of a parametric curve.

```
> plot([3*sin(t), 5*cos(t), t=-Pi..Pi]);
```



Plot of a curve in polar coordinates in chosen colour and line thickness.

```
> plot([sin(3*x),x,x=0..2*Pi],coords=polar,thickness=3);
```



Function `plot3d(expr1, x=a..b, y=c..d)` shows graph of function in two variables, surface patch defined over a region in coordinate plane  $xy$ .

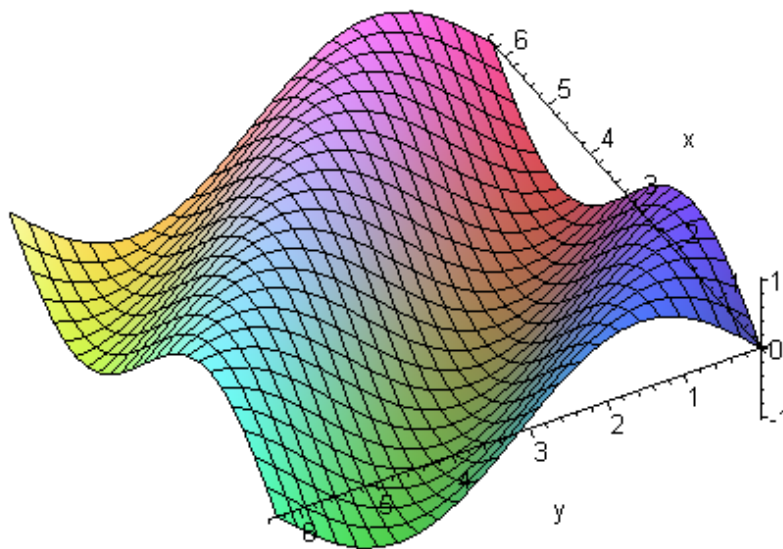
Parameters are as follows:

`expr1` – function, graph of which will be plotted,

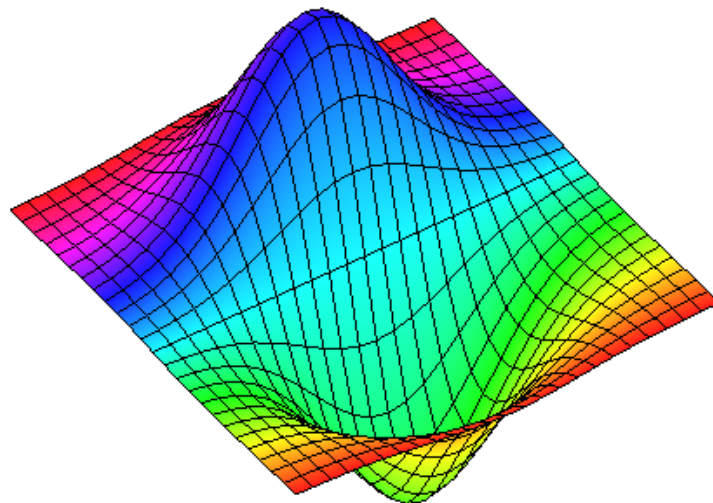
`x=a..b` – range on the coordinate axis  $x$

`y=c..d` – range on the coordinate axis  $y$

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

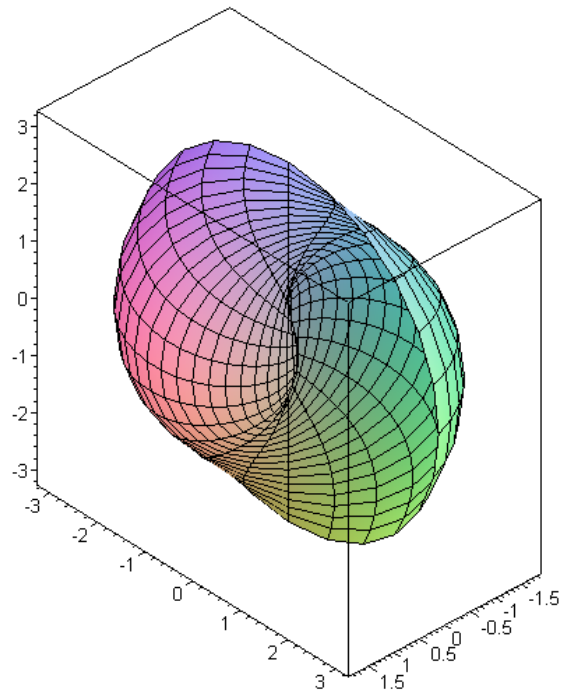


> `plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2,color=x);`



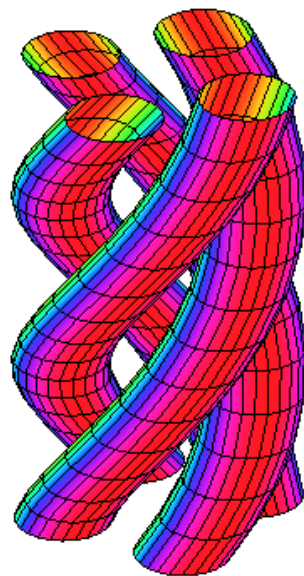
Plot of a parametric surface patch.

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



Plot of more surface patches in one graphics.

```
> 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],style=patch,color=sin(x));
```



Plot functions in the system Maple include default definition of the Cartesian coordinate system. Parameter `coords` enables to define also other coordinate systems, e.g. polar, cylindrical spherical, bispherical, cardioidal, cardioidal-cylindrical, conical, a i. Standard forms of plot functions for some of the coordinate systems are given below:

- Cartesian coordinate system

```
plot3d(z(x,y), x=a..b, y=c..d);
```

- Cylindrical coordinate system

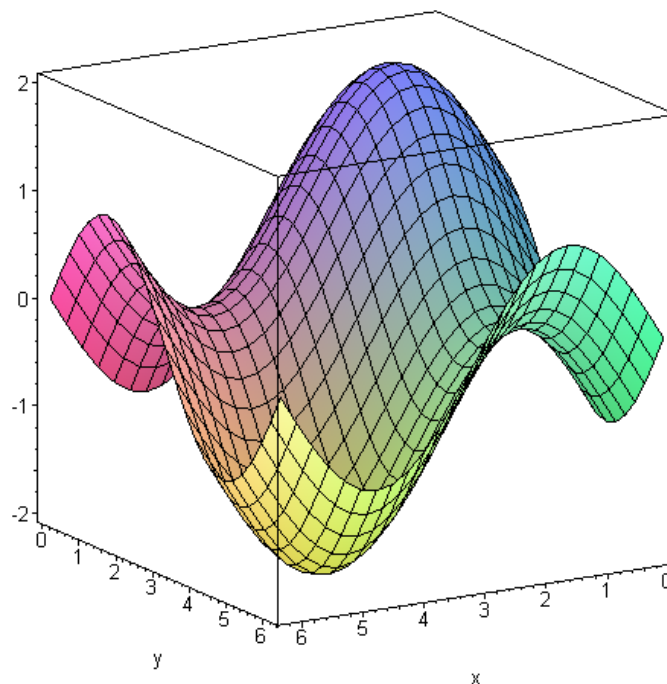
```
plot3d(r(theta,z), theta=a..b, z=c..d, coords=cylindrical);
```

- Spherical coordinate system

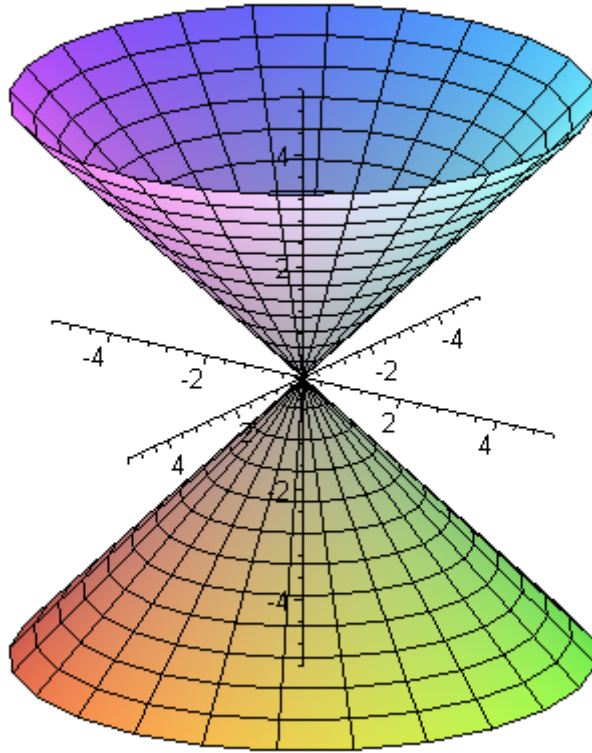
```
plot3d(r(theta,phi), theta=a..b, phi=c..d, coords=spherical);
```

Several examples of graphics with views of some surfaces in different coordinate systems follow.

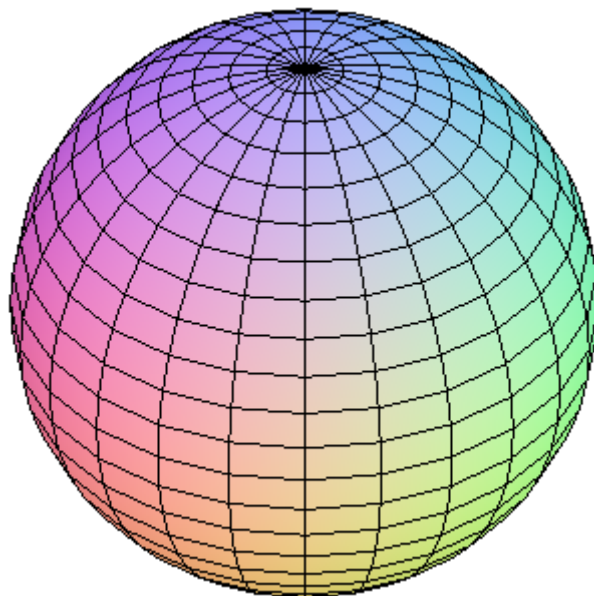
```
. > plot3d(sin(x)+sin(y),x=0..2*Pi, y=0..2*Pi, axes=boxed);
```



```
> plot3d(height,angle=0..2*Pi,height=-5..5, coords=cylindrical, axes=normal);
```

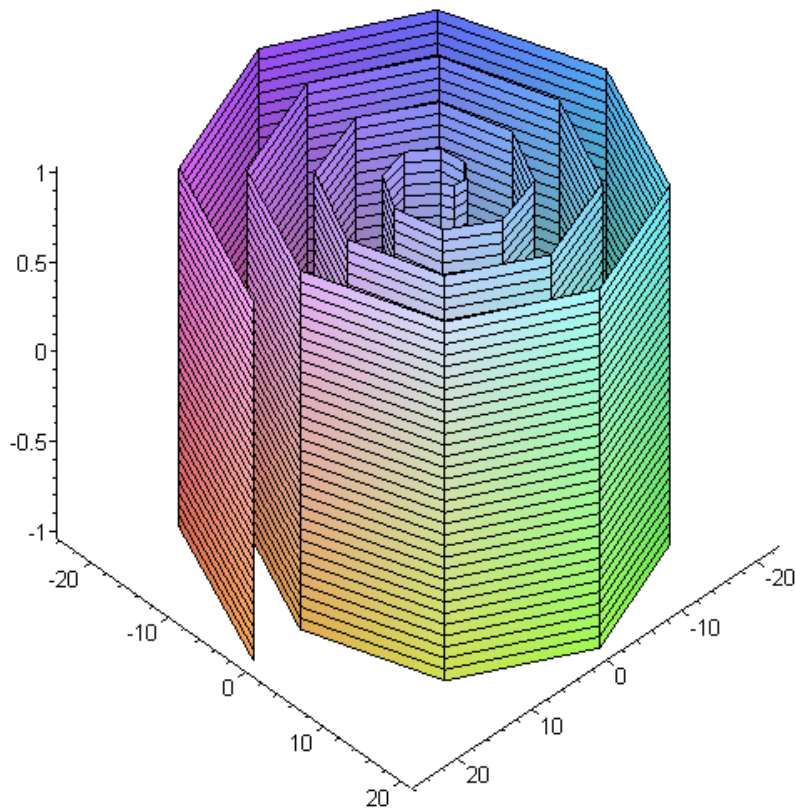


```
> plot3d(1,t=0..2*Pi,p=0..Pi,coords=spherical,scaling=constrained);
```





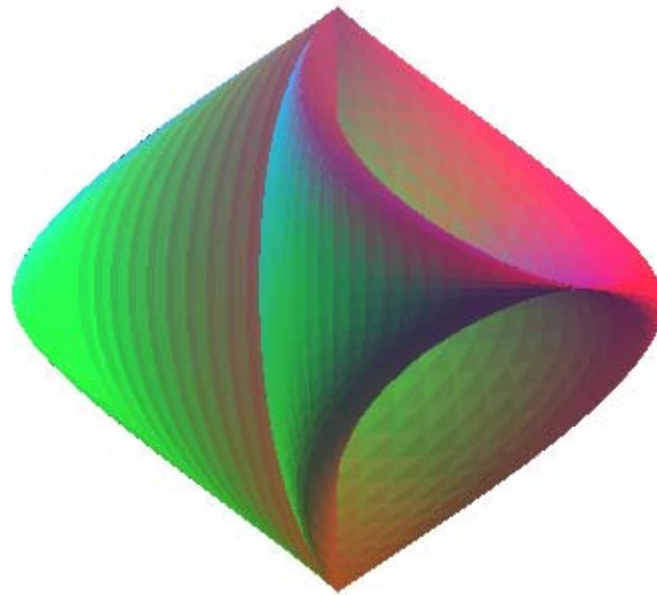
```
> plot3d(theta,theta=0..8*Pi,z=-1..1, coords=cylindrical, style=patch,  
numpoints=1000, axes=framed);
```



Specific attributes can be set either by direct explicit quotation in the function plot or plot3d syntax, or these can be set interactively, from the menu in the palette, which can be viewed by the right mouse click. The following options are in the menu:

- Copy – to copy plot to the clipboard
- Style – settings of the surface patch view (patch, patch grid, patch and contour, hidden line, contour, wireframe, point, default), size and type of used symbols, line width and style (solid, dot, dash, dash-dot, default) and grid style (triangular, or rectangular)
- Colour – settings of the colour palette and light scheme
- Axes – available viewing styles are: normal, boxed, framed, or none for no axes plotted
- Projection – type of perspective and scale on axes (constrained, unconstrained)
- Export – export of graphics to some of the graphical files: DXF, EPS, GIF, JPG, POV, BMP, WMF.

```
> plot3d([(-3*cos(2*Pi*u))^4*sin(2*Pi*v)*cos(2*Pi*v),
          (-3*cos(2*Pi*u))^4*sin(2*Pi*v)*sin(2*Pi*v),
          7*cos(2*Pi*v)*sin(2*Pi*u)], u=0..1,v=0..1,numpoints=4000,);
```



Additional graphical package of the system Maple – plots provides functions for generating complex graphical representations and different nonstandard functions, as e.g. animate, animate3d, animatecurve, changeccord, contourplot3d, display, implicitplot3d, listplot, pointplot, pointplot3d, polyhedra, spacecurve, tubeplot, ao.

```
> with(plots):
   tubeplot( [-10*cos(t) - 2*cos(5*t) + 15*sin(2*t),
             -15*cos(2*t) + 10*sin(t) - 2*sin(5*t),
             10*cos(3*t) ], t= 0..2*Pi,radius=5*cos(t*Pi/3));
```

