plot3d - three-dimensional plotting

Calling Sequence:

\[
\text{plot3d}(\text{expr1}, x=a..b, y=c..d) \\
\text{plot3d}(f, a..b, c..d) \\
\text{plot3d}([\text{exprf}, \text{exprg}, \text{exprh}], s=a..b, t=c..d) \\
\text{plot3d}([f, g, h], a..b, c..d)
\]

Parameters:

\[
\begin{align*}
f, g, h & \quad \text{function(s) to be plotted} \\
\text{expr1} & \quad \text{expression in } x \text{ and } y. \\
\text{exprf}, \text{exprg}, \text{exprh} & \quad \text{expressions in } s \text{ and } t. \\
a, b & \quad \text{real constants.} \\
c, d & \quad \text{real constants, procedures or expressions in } x \\
x, y, s, t & \quad \text{names}
\end{align*}
\]

Description:

- The four different calling sequences to the plot3d function above all define surfaces. Facilities for plotting curves and other objects are in the plots package. The first two calling sequences describe surface plots in cartesian co-ordinates while the second two describe parametric surface plots.

- In the first call, \( \text{plot3d}(\text{expr1}, x=a..b, y=c..d) \), the expression \( \text{expr1} \) must be a Maple expression in the names \( x \) and \( y \). The range \( a..b \) must evaluate to real constants. The range \( c..d \) must either evaluate to real constants or be expressions in \( x \). They specify the range over which \( \text{expr1} \) will be plotted. In the second call, \( \text{plot3d}(f, a..b, c..d) \), \( f \) must be a Maple procedure or operator which takes two arguments. Operator notation must be used, i.e. 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. The second range \( c..d \) can have arguments evaluating to real constants or procedures of one-variable.

- A parametric surface can be defined by three expressions \( \text{expr1}, \text{expr2}, \text{expr3} \) in two variables. In the third call, \( \text{plot3d}([\text{expr1}, \text{expr2}, \text{expr3}], s=a..b, t=c..d) \), \( \text{expr1}, \text{expr2}, \) and \( \text{expr3} \) must be Maple expressions in the names \( s \) and \( t \). Finally, in the fourth call, \( \text{plot3d}([f, g, h], a..b, c..d) \), \( f, g \) and \( h \) must be Maple procedures or operators taking two arguments. Here again, operator notation must be used.

- Any additional arguments are interpreted as options which are specified as equations of the form \( \text{option} = \text{value} \). For example, the option \( \text{grid} = [m,n] \) where \( m \) and \( n \) are positive integers specifies that the plot
is to be constructed on an \( m \times n \) grid at equally spaced points in the ranges \( a..b \) and \( c..d \) respectively. By default a 25 by 25 grid is used, thus 625 points are generated. Other options include specification of alternate coordinate systems, rendering styles, etc. See the help page for \( \text{plot3d}[\text{options}] \).

- When \( \text{plot3d} \) evaluates its arguments, any errors generated during the evaluation are suppressed by the plot package. A symptom that something has gone wrong with the evaluation of your expression is a resulting empty plot.

- The result of a call to \( \text{plot3d} \) is a PLOT3D data structure containing enough information to render the plot. The user may assign a PLOT3D value to a variable, save it in a file, then read it back in for redisplay. See the help page for \( \text{plot3d}[\text{structure}] \).

- All plotted expressions are evaluated numerically, that is, as floating point expressions, rather than symbolically.

- Note: for other types of 3d plots, see \( \text{plots} \).

### Examples:

```plaintext
> plot3d(sin(x+y),x=-1..1,y=-1..1);
plot3d(binomial,0..5,0..5,grid=[10,10]);
```

- can have alternate coordinate systems

```plaintext
> plot3d((1.3)^x * sin(y),x=-1..2*Pi,y=0..Pi,coords=spherical,style=patch);
plot3d([1,x,y],x=0..2*Pi,y=0..2*Pi,coords=toroidal(10),scaling=constrained);
plot3d(sin(x*y),x=-Pi..Pi,y=-Pi..Pi,style=contour);
```

- can have variable endpoints in some cases

```plaintext
> plot3d(sin(x*y),x=-Pi..Pi,y=-x..x);
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);
plot3d([x*sin(x)*cos(y),x*cos(x)*cos(y),x*sin(y)],x=0..2*Pi,y =0..Pi);
plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2,grid=[49,49]);
```

- can specify a color function (or procedure)

```plaintext
> plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2,color=x);
plot3d(p,-2..2,-1..h,color=h);
```

- multiple 3d plots can also be done
> 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], style=patch);

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

See Also:

plot, plot3d[option], plot3d[structure], plot3d[coords], plots