plot - create a two-dimensional plot of functions

Calling Sequence:

plot(f, h, v)

plot(f, h, v,...)

Parameters:

f - function(s) to be plotted

h - horizontal range

v - vertical range (optional)

Description:

- A typical call to the plot function is plot(f(x),x=a..b), where f is a real function in x and a..b specifies the horizontal real range on which f is plotted.

- Parametric plots are specified in the form \([fx,fy,range]\). The presence of a range in the third position of the list distinguishes parametric plots from multiple function plots.

- Multiple functions can be plotted by placing a set or list of functions in the first argument. Options such as color, thickness and style can be specified for each function by placing a list in the appropriate option argument (see below).

- The plot function provides support for two-dimensional plots of one or more functions specified as expressions, procedures, or parametric functions. (When plotting a procedure, operator notation must be used.) See plot[function] for more information on plotting functions. For three-dimensional plots, see plot3d. A call to plot produces a PLOT data structure, which is then printed. For information on the PLOT data structure, see plot[structure].

- The horizontal and vertical range arguments h and v define the axis labels and the range over which the function(s) are displayed. They take one of the following forms: string, low..hi, or string=low..hi, where low and hi are real constants. See plot[range] for further information.

- Remaining arguments are interpreted as options which are specified as equations of the form option = value. In particular, the style option allows one to plot the points as points only, or to interpolate them using line mode. See plot[options] for more information.

- Note that plot now automatically attempts to determine the locations of discontinuities before plotting. For expressions, this is achieved via discont; in the case of procedures and functions, fdiscont is used. Discontinuity searching can be turned off by setting the discont option to false. See plot[options] for more details.
• Setting the interface variable plotdevice (via the plotsetup command) to one of the values listed under plot[device] controls the type of plot which will be produced. See plot[device] and plot[setup] for information on how to set up plots on a particular device (printer or terminal).

• An empty plot may result if errors occur during the evaluation of the arguments to plot. This sometimes confuses new users because error messages are suppressed during the evaluation of these arguments. Errors generated during the evaluation of functions are replaced with the value undefined to allow the plots system to better handle singularities.

• Note that the arguments to the plot function are evaluated numerically, not symbolically.

--- Examples:

```plaintext
> plot(cos(x) + sin(x), x=0..Pi);
plot(tan(x), x=-Pi..Pi);
plot([sin(t), cos(t), t=-Pi..Pi]);
plot(sin(t), t);
```

Since no domain is specified in the last example, we use t=-10..10.

Same four plots, but as procedures or operators:

```plaintext
> plot(cos + sin, 0..Pi);
plot(tan, -Pi..Pi);
plot([sin, cos, -Pi..Pi]);
plot(sin);
```

For expressions having discontinuities over finite intervals one can do:

```plaintext
> plot(tan(x), x = -2*Pi..2*Pi, y = -4..4, discont = true);
plot(-1 + 2*Heaviside(x-1), x = -1..2, discont = true);
```

Multiple plots (in a set or list):

```plaintext
> plot([sin(x), x-x^3/6], x=0..2, color=[red,blue],
    style=[point,line]);
```

infinity plots

```plaintext
> plot(sin(x), x=0..infinity);
```

point plots

```plaintext
> l := [[ n, sin(n)] $n=1..10];
plot(l, x=0..15, style=point,symbol=circle);
```

some other plots

```plaintext
> s := t->100/(100+(t-Pi/2)^8): r := t ->
```
s(t)*(2-sin(7*t)-cos(30*t)/2):
plot([r(t), t=-Pi/2..3/2*Pi], numpoints=2000, coords=polar, axes=none):
plot([x, tan(x), x=-Pi..Pi], -4..4, -5..5, tickmarks=[8,10]);

polar coordinates (with thickened curve)

> plot([sin(4*x), x, x=0..2*Pi], coords=polar, thickness=3);

See Also:

plot[spec] where spec is one of infinity, polar, parametric, multiple, ranges, function, options, structure, setup, device, replot, style, color. Use plot3d for plotting surfaces. See discont and fdiscont regarding discontinuities.