

# plotplots

plots data along a discontinuous or multiscaled axis

## Syntax

```
plotplots(x, y, intervals, cutaxis)
plotplots(.., "ticksmode", "alt" | "shift" | "labels_shift")
plotplots(.., "widths", "isoscaled" | widths)
[curves, axes] = plotplots(..)
```

## Arguments

### x

vector or matrix of real abscissae.

- If  $x$  and  $y$  are vectors, we must have  $\text{length}(x) = \text{length}(y)$
- If it is a vector and  $y$  is a matrix, the same abscissae are shared for all curves given as  $y$  columns, and we must have  $\text{length}(x) = \text{size}(y, 1)$ .
- If both  $x$  and  $y$  are matrices, they must have the same sizes, and  $x(:, i)$  are abscissae of  $y(:, i)$ .

### y

real numbers: ordinates of curves to be plotted. If  $y$  is a matrix, each of its columns represents a separate curve.

### intervals

( $N \times 2$ ) matrix of real numbers: Each row provides both  $[\text{start}, \text{end}]$  bounds of an interval over which all given curves must be plotted. According to the `cutaxis` value, given bounds are along the X axis or along the Y one. Any number of intervals may be specified as as many `intervals` rows.

On the figure, intervals are plotted from left to right in the order they are specified in `intervals`. The user may organize and present them as wished.

When an interval is specified such that  $\text{start} > \text{end}$ , the axis is reversed on this part.

### cutaxis

"x" or "y", specifying the discontinuous axis.

### "ticksmode"

By default, ticks and labels of the different parts of the discontinuous axis are drawn "aligned", without specific attribute to distinguish consecutive parts. "ticksmode" is an optional flag allowing to manage ticking and labelling of alternate pieces of the cut axis. It must be followed by one of the following values:

"alt" Ticks and labels are drawn alternately on the bottom and top sides for the "x" discontinuous axis, or on the left and right sides for the "y" one.

"shift" Ticks and labels are drawn with a small alternative shift for one interval over two of the discontinuous axis.

### "widths"

By default, if  $N$  intervals are defined, the full plot width is split into  $N$  sections of equal widths, whatever are the  $|\text{end} - \text{start}|$  data ranges. This default plotting mode may be changed using the "widths" keyword, followed by one of the following specifications:

"-" As the default mode, except that interspaces separating sections are removed.

"isoscaled" This mode plots data with the same scale along the discontinuous axis, for all intervals. The plotting width is proportional to the  $|\text{end} - \text{start}|$  range.

"-isoscaled" will do the same, except that interspaces separating sections are removed.

widths Vector of numbers giving the relative graphical widths of plotting sections along the discontinuous axis. The vector must have as many values as there are defined intervals.

Only  $|\text{widths}| / \text{sum}(|\text{widths}|)$  is considered, in such a way that  $[1, 2, 2]$  and  $[0.5, 1, 1]$  or  $[0.2, 0.4, 0.4]$  are for instance equivalent.


To remove the separating space before a section, set its width as negative. Only its absolute value will be actually taken into account.

## curves

list of vectors of pointers on plotted curves:

- `curves(i)` is a vector of graphic pointers on all sections drawn for the curve #i defined in `y(:,i)`.
- `curves(i)(j)` points to the section #j drawn for the curve #i over `intervals(j,:)`.

These identifiers allow to modify curves attributes after plotting. For instance, setting in red all pieces of the curve #2 will be done with `curves(2).foreground = color("red")`. Setting only its section #3 in dash style is done with `curves(2)(3).line_style = 3`. An example is given and illustrated below.

 The vector of handles of curves #i1 to #i2 > #i1 of the same section #s may be retrieved with `axes(s).children($).children($-i1+1:-1:$-i2+1)`.

## axes

vector of pointers on plotted sub-axes. The components allow to modify subaxes attributes, like the log mode of the piece of discontinuous axis, etc. See examples for illustrations.

`axes($)` points to the common continuous axes. `plotplots()` makes it the current axes before returning.

## Description

`plotplots()` allows to plot some curves along a non-regular axis, like having some hidden intervals. Consecutive visible intervals are drawn without loosing space in-between.

The different parts of the discontinuous axis may have distinct scales, distinct linear or logarithmic modes, distinct direct or reversed modes.

A symmetrical axis may be plotted by repeating an interval along the discontinuous axis and reverting the clone. See examples.

 `plotplots()` can be used in conjunction to the following features:

- `subplot(..)` or `xsetech(..)` can be used to define the plotting area before calling `plotplots()`.
- A zoom box spanning over several pieces of the discontinuous axis may be selected interactively.

## Titles, legends and datatips

**General title, title of the continuous common axis:** `title()`, `xtitle()`, `xlabel()` when `cutaxis` is "y" or `ylabel()` when `cutaxis` is "x" can be used as usual after calling `plotplots()`.

**Title of the discontinuous axis:** It is highly recommended to set a title to the axis of the most central plotted section, rather than to the axis of the overall axes (which introduces a parasitic shift). The handle of the targeted section will be used in conjunction to `xlabel()` or `ylabel()`. Please see examples.

**Datatips:** can be set as usual on curves drawn with `plotplots()`. However, a given datatip can't be dragged from a section of the plot to other ones.

**Block of curves legends:** Such a block can be defined in a given plotting interval, by targeting the related axes:

```
sca(axes(i));
legend(["Curve 1" "Curve 2"], "in_upper_left");
```

The chosen position refers to the bounds of the targeted section. There is no way to refer to the whole axes defining the shared axis.

## Examples

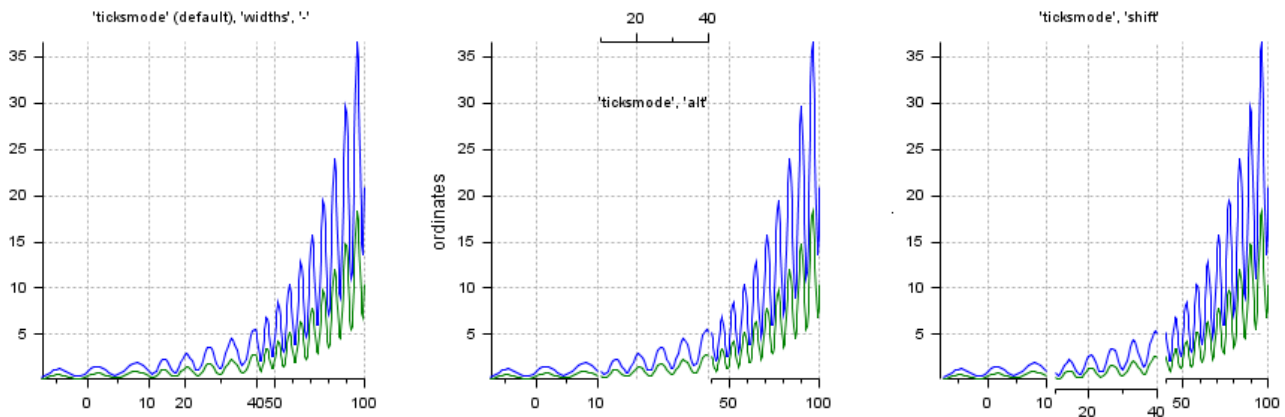
```
x = [-7.5:0.5:200]';
y = exp(x/30).*(1+0.5*sin(x));

f =(gcf()); f.axes_size = [900 300];
clf

// "ticksmode" along "x"
subplot(1,3,1)
intervals = [-7.5 10 ; 10 40 ; 40 100];
[c, a] = plotplots(x, [y y/2], intervals, "x", "widths", "-");
title(['''ticksmode'' (default), ''widths'', ''-'''])

subplot(1,3,2)
[c, a] = plotplots(x, [y y/2], intervals, "x", "ticksmode", "alt");
ylabel("ordinates")
xstring(47, 30, '''ticksmode'', 'alt'');
e = gce(); e.text_box_mode = "centered";

subplot(1,3,3)
plotplots(x, [y y/2], intervals, "x", "ticksmode", "shift");
title('''ticksmode'', 'shift'')
```



```
x = [-7.5:0.5:200]';
y = exp(x/30).*(1+0.5*sin(x));

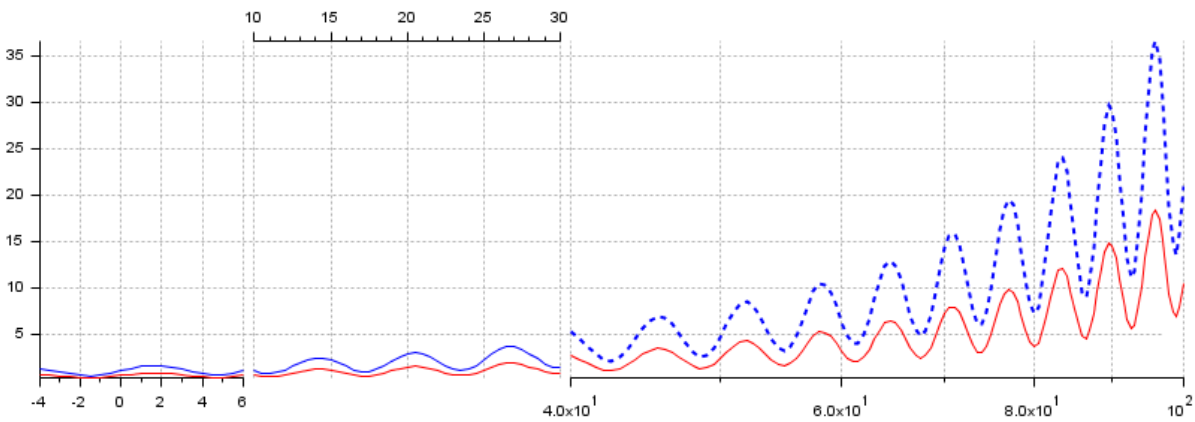
f = gcf(); f.axes_size = [1000 300];
clf

intervals = [-4 6; 10 30; 40 100];
[c, a] = plotplots(x, [y y/2], intervals, "x", "ticksmode", "alt", "widths", [1 1.5 3]);

// Special curves settings:
c(2).foreground = color("red");           // 2nd cuve in red
c(1)(3).line_style = 8;                   // 3rd section of 1st curve in small dashes
c(1)(3).thickness = 2;                    // .. and thicker

// Invert the common Y axis
//a.axes_reverse = ["off" "on"];

// Setting X-axis of the third section in Log:
a(3).log_flags = "lnn";
```



### Other usages of intervals:

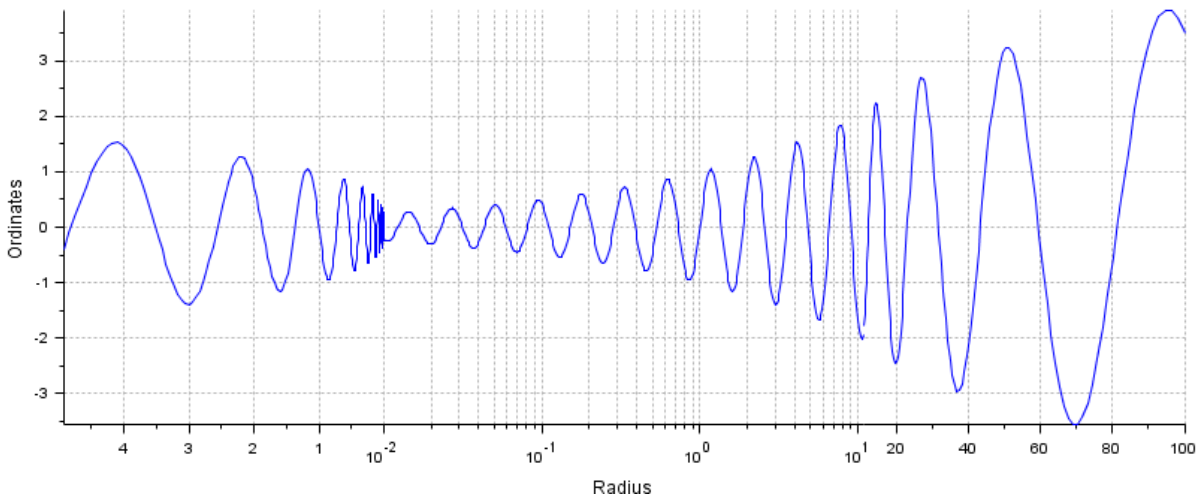
plotplots() allows to specify overlaying intervals, or the same interval several times, or twice the same but in reverse order. This may be useful in various situations. For a positive radial coordinate, it is thus possible to tick a positive distance from the center on both sides. A side can be plotted in radial linear scale, and the other in radial logarithmic scale to show the asymptotical behavior:

```
x = logspace(-2,2,500)';
y = sin(10*log(x)).*(x.^0.3);

f = gcf(); f.axes_size = [1000 370];
clf

intervals = [5 0; 0.01 11; 11 max(x)];
[c, a] = plotplots(x, y, intervals, "x", "widths", -[2 3 2]);
a(2).log_flags = "lnn";
a(2).sub_ticks(1) = 8;
title("Radial singularity on 0", "fontsize", 3)
ylabel("Ordinates", "fontsize", 2)
xlabel(a(2), "Radius", "fontsize", 2);
```

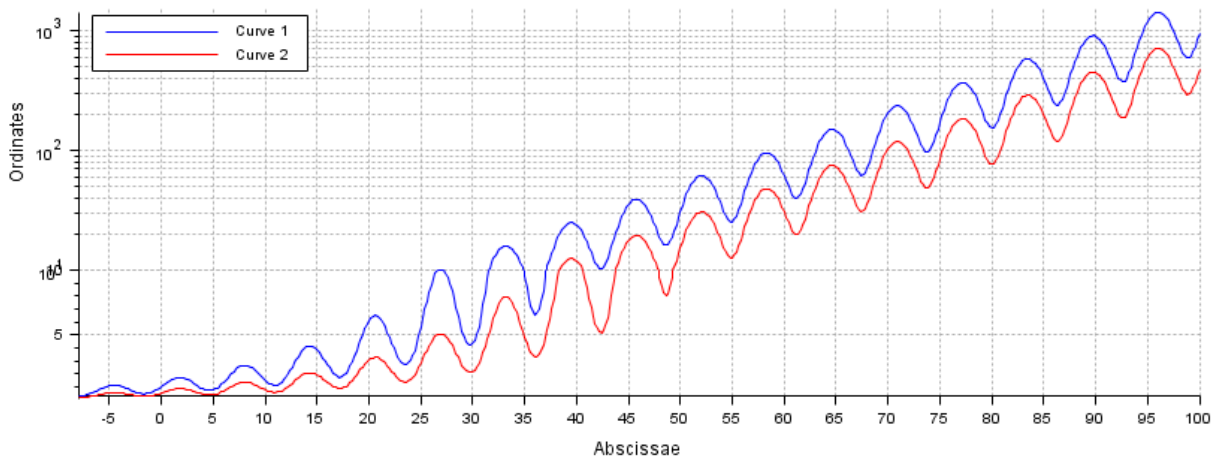
Radial singularity on 0



**Multiscaled Y axis:**

```
x = [-8:0.2:100]';
y = exp(x/14).*(1+0.5*sin(x));
intervals = [0.01 10 ; 10 max(y)];

f =(gcf()); f.axes_size = [1000 400];
clf
[c, a] = plotplots(x, [y y/2], intervals, "y", "widths", -[1 2]);
c(2).foreground = color("red");
a(2).log_flags = "nln";
a(2).sub_ticks(2) = 8;
xlabel("Abcissae", "fontsize", 2)
ylabel(a(2),"Ordinates", "fontsize", 2) // a(2) is now the current axes
legend(["Curve 1" "Curve 2"], "in_upper_left")
```



**See Also**

- <http://atoms.scilab.org>
- [plot2d](#)
- [plot](#)

**Author**

(C) Samuel GOUGEON - France

**History**

Version	Description
1.0	Initial release