SVKM'S NMIMS Deemed to be UNIVERSITY | NAVI MUMBAI

MATLAB

Unit 4-Lecture 11

BTech (CSBS) -Semester VII

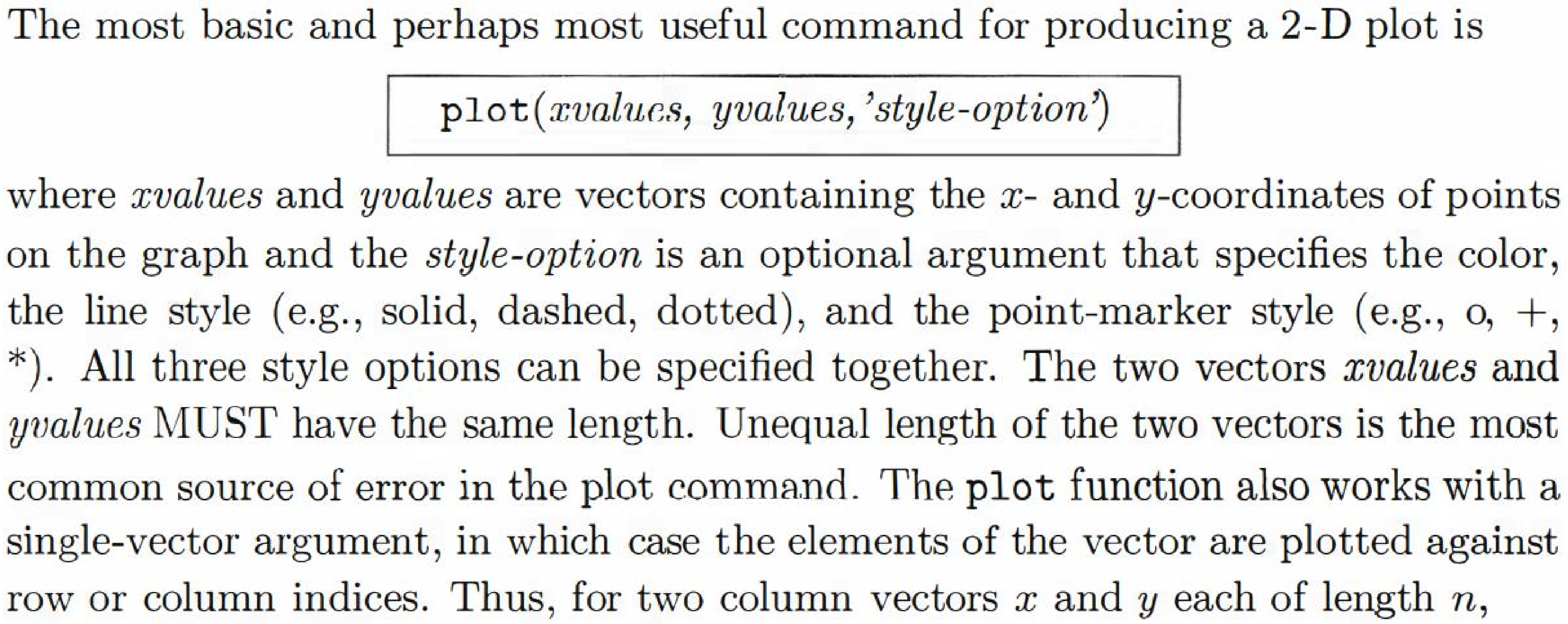
16 August 2022, 09:35AM

*few figures/content have been prepared/referred from internet source/books

Basic plotting

- Overview,
- axis labels, and annotations,
- adding titles,
- specifying line styles and colours.
- creating simple plots,
- multiple data sets in one plot,

Overview

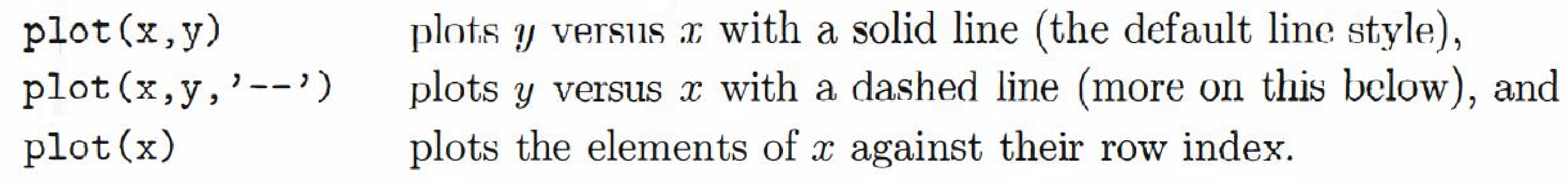


The most basic and perhaps most useful command for producing a 2-D plot is

`plot`(*xvalucs*, *yvalues*,'*style-option*')

where *xvalues* and *yvalues* are vectors containing the x- and y-coordinates of points on the graph and the *style-option* is an optional argument that specifies the color, the line style (e.g., solid, dashed, dotted), and the point-marker style (e.g., o, +, *). All three style options can be specified together. The two vectors *xvalues* and *yvalues* MUST have the same length. Unequal length of the two vectors is the most common source of error in the plot command. The `plot` function also works with a single-vector argument, in which case the elements of the vector are plotted against row or column indices. Thus, for two column vectors x and y each of length n,

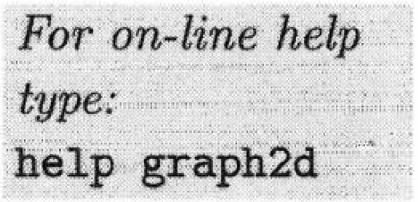
Overview



`plot(x,y)` plots y versus x with a solid line (the default line style),

`plot(x,y,'--')` plots y versus x with a dashed line (more on this below), and

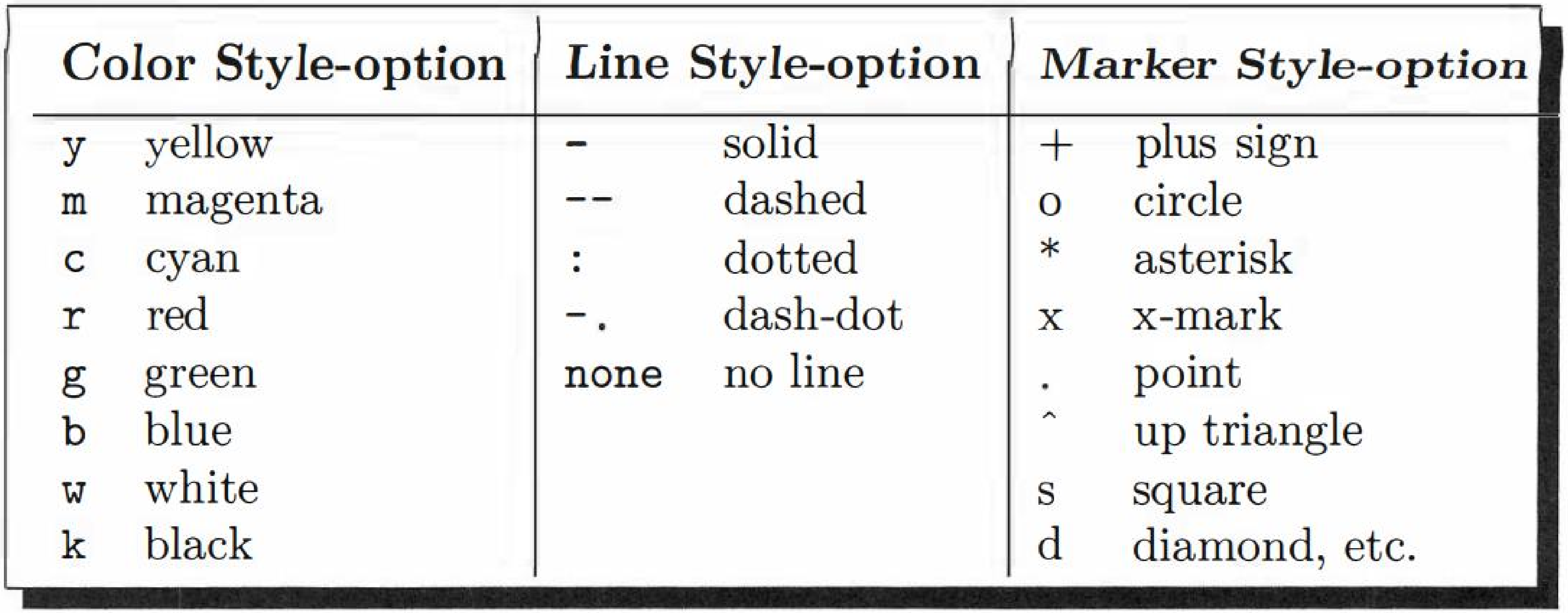
`plot(x)` plots the elements of x against their row index.



For on-line help type:

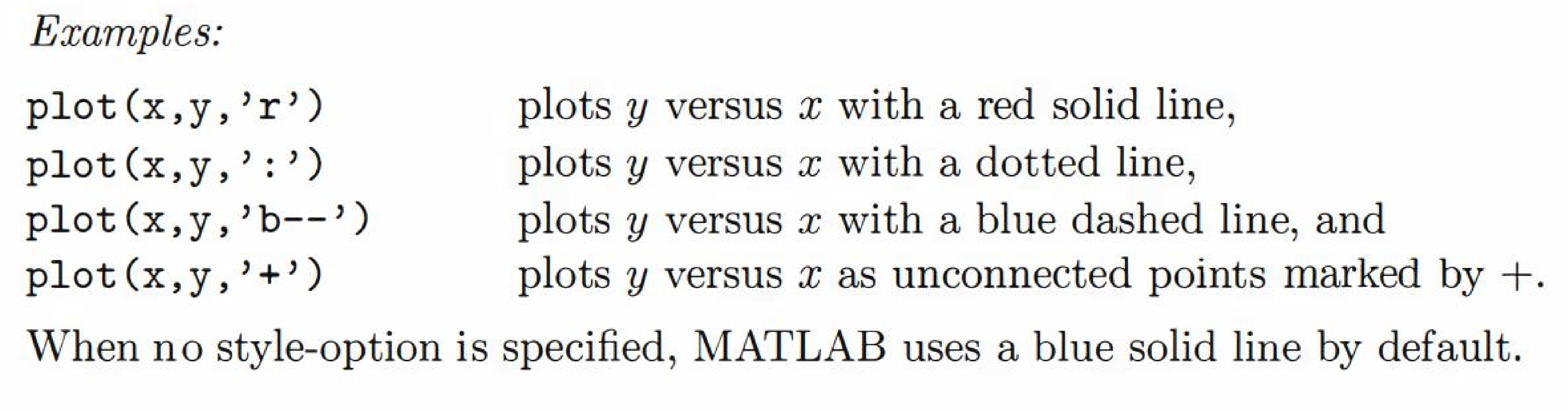
```
help graph2d
```

Style Options



Color Style-option		Line Style-option		Marker Style-option	
y	yellow	-	solid	+	plus sign
m	magenta	--	dashed	o	circle
c	cyan	:	dotted	*	asterisk
r	red	-.	dash-dot	x	x-mark
g	green	none	no line	.	point
b	blue			^	up triangle
w	white			s	square
k	black			d	diamond, etc.

Style Options



Examples:

`plot(x,y,’r’)` plots y versus x with a red solid line,

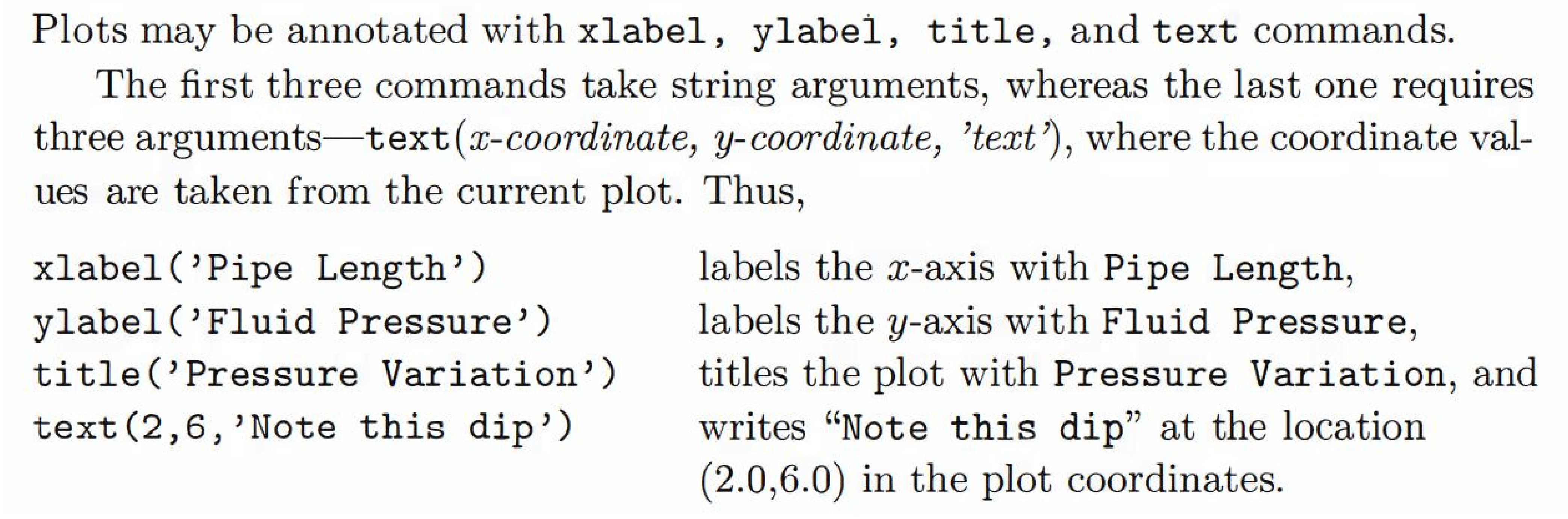
`plot(x,y,’:’)` plots y versus x with a dotted line,

`plot(x,y,’b--’)` plots y versus x with a blue dashed line, and

`plot(x,y,’+’)` plots y versus x as unconnected points marked by $+$.

When no style-option is specified, MATLAB uses a blue solid line by default.

Label and title

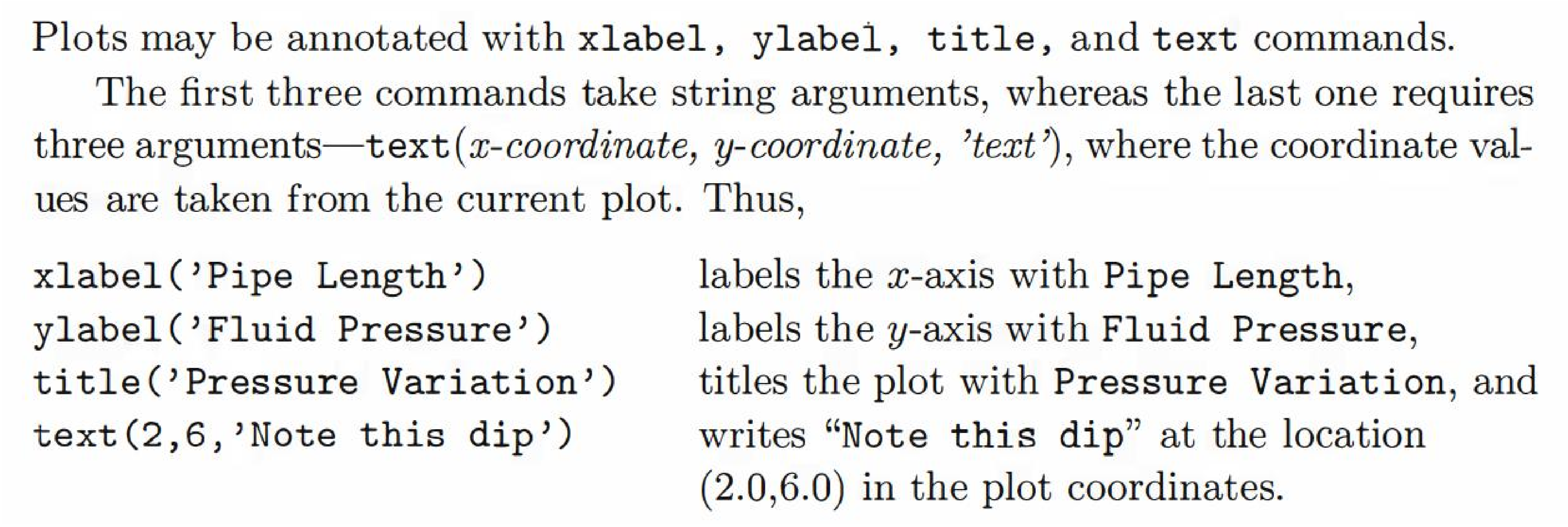


Plots may be annotated with `xlabel`, `ylabel`, `title`, and `text` commands.

The first three commands take string arguments, whereas the last one requires three arguments—`text`(*x-coordinate*, *y-coordinate*, *'text'*), where the coordinate values are taken from the current plot. Thus,

`xlabel('Pipe Length')`	labels the x-axis with `Pipe Length`,
`ylabel('Fluid Pressure')`	labels the y-axis with `Fluid Pressure`,
`title('Pressure Variation')`	titles the plot with `Pressure Variation`, and
`text(2,6,'Note this dip')`	writes "`Note this dip`" at the location (2.0,6.0) in the plot coordinates.

Label and title

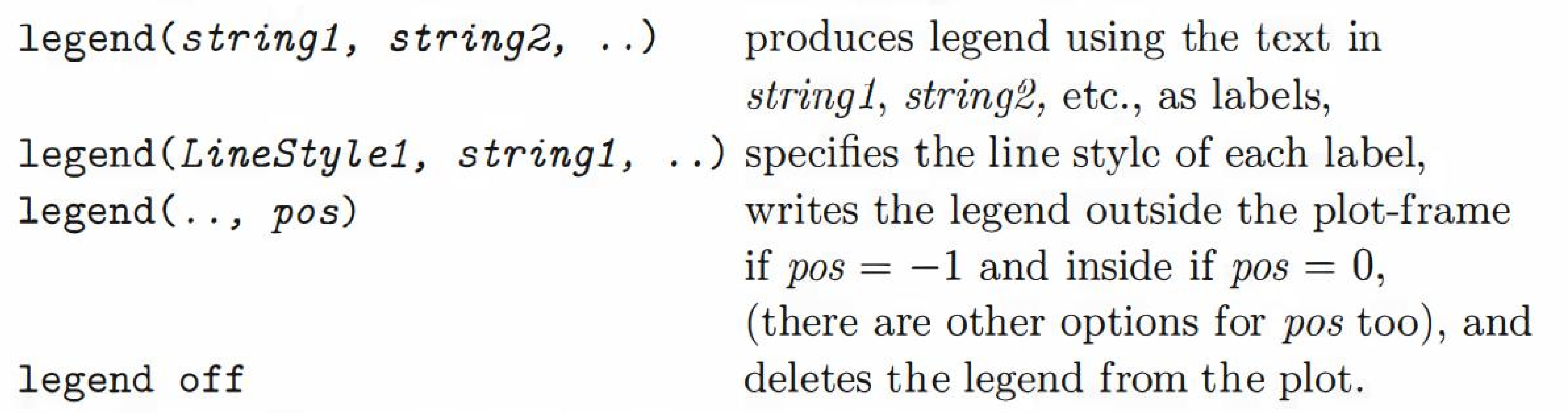


Plots may be annotated with `xlabel`, `ylabel`, `title`, and `text` commands.

The first three commands take string arguments, whereas the last one requires three arguments—`text`(*x-coordinate, y-coordinate, 'text'*), where the coordinate values are taken from the current plot. Thus,

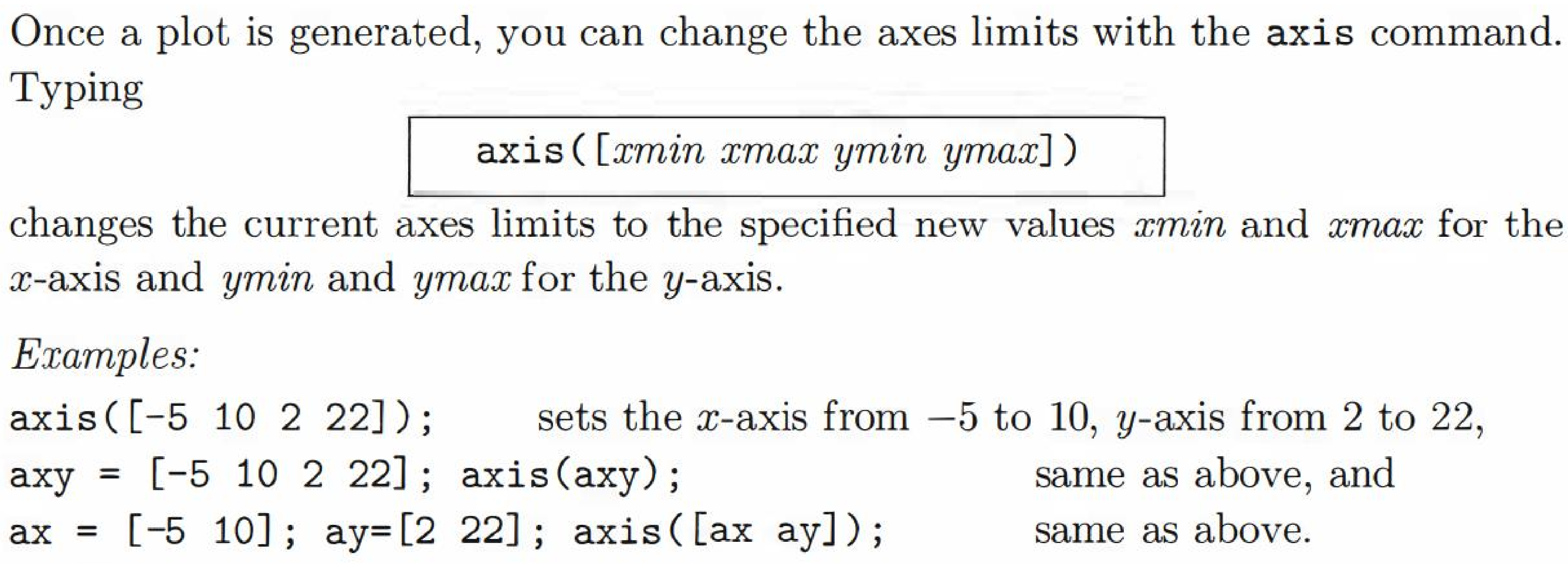
`xlabel('Pipe Length')`	labels the x-axis with `Pipe Length`,
`ylabel('Fluid Pressure')`	labels the y-axis with `Fluid Pressure`,
`title('Pressure Variation')`	titles the plot with `Pressure Variation`, and
`text(2,6,'Note this dip')`	writes "`Note this dip`" at the location (2.0,6.0) in the plot coordinates.

Legend



`legend(`*`string1, string2,`*` ..)`	produces legend using the text in *string1*, *string2*, etc., as labels,
`legend(`*`LineStyle1, string1,`*` ..)`	specifies the line stylc of each label,
`legend(.., `*`pos`*`)`	writes the legend outside the plot-frame if *pos* $= -1$ and inside if *pos* $= 0$, (there are other options for *pos* too), and
`legend off`	deletes the legend from the plot.

Axis Control



Once a plot is generated, you can change the axes limits with the **`axis`** command. Typing

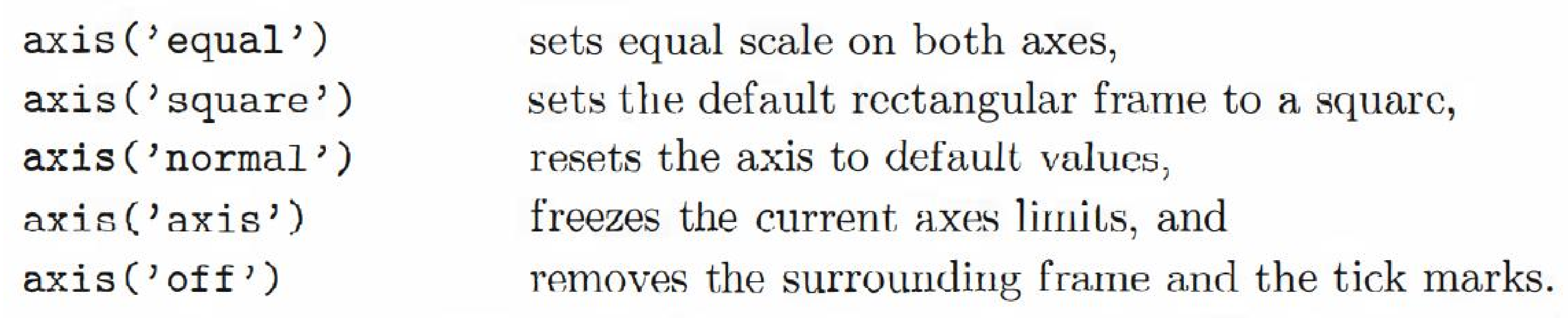
`axis([`*xmin xmax ymin ymax*`])`

changes the current axes limits to the specified new values *xmin* and *xmax* for the x-axis and *ymin* and *ymax* for the y-axis.

Examples:

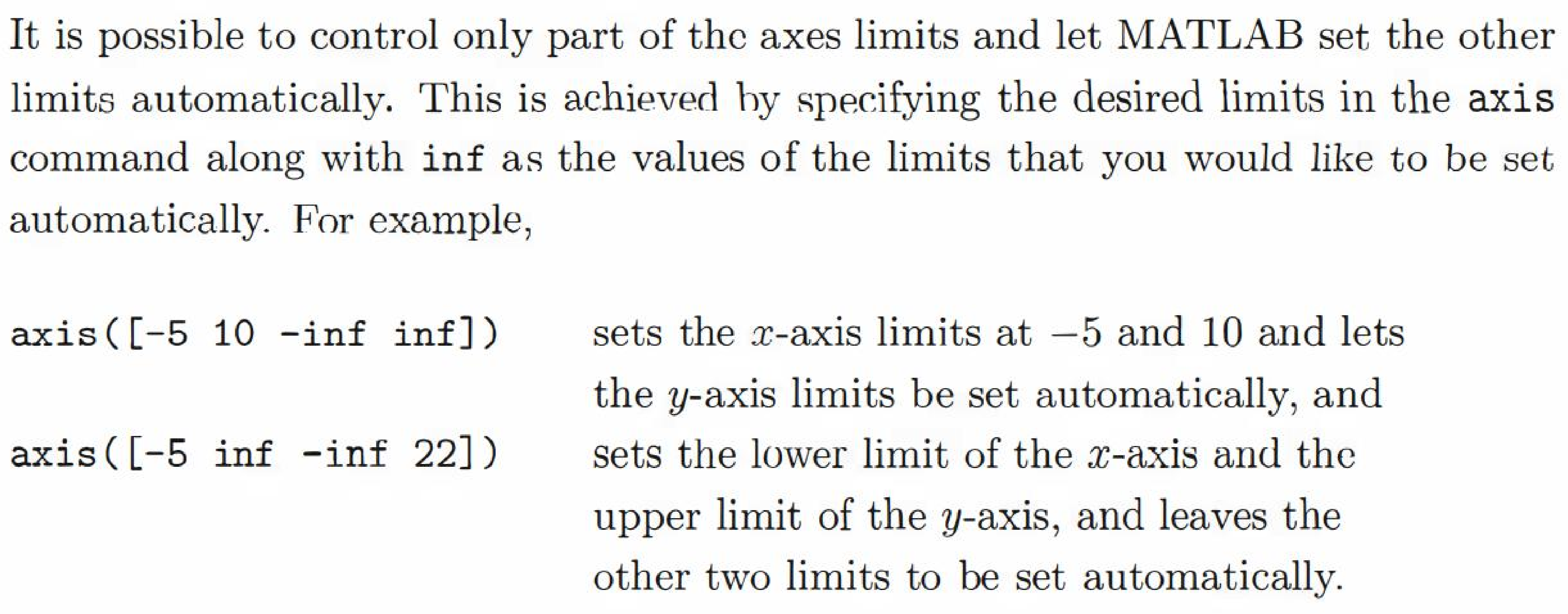
`axis([-5 10 2 22]);`	sets the x-axis from -5 to 10, y-axis from 2 to 22,
`axy = [-5 10 2 22]; axis(axy);`	same as above, and
`ax = [-5 10]; ay=[2 22]; axis([ax ay]);`	same as above.

Axis Control



`axis('equal')`	sets equal scale on both axes,
`axis('square')`	sets the default rectangular frame to a square,
`axis('normal')`	resets the axis to default values,
`axis('axis')`	freezes the current axes limits, and
`axis('off')`	removes the surrounding frame and the tick marks.

Semi control of Axis

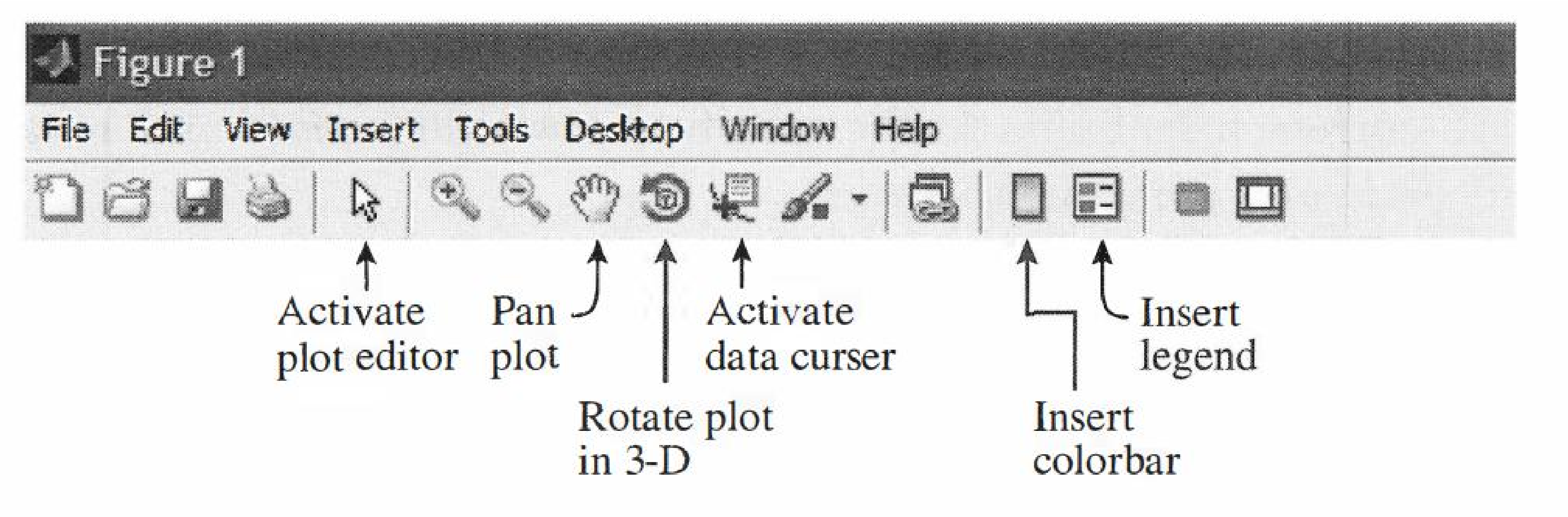


It is possible to control only part of the axes limits and let MATLAB set the other limits automatically. This is achieved by specifying the desired limits in the `axis` command along with `inf` as the values of the limits that you would like to be set automatically. For example,

`axis([-5 10 -inf inf])` sets the x-axis limits at -5 and 10 and lets the y-axis limits be set automatically, and

`axis([-5 inf -inf 22])` sets the lower limit of the x-axis and the upper limit of the y-axis, and leaves the other two limits to be set automatically.

Modify plot with Plot Editor



Activate plot editor

Pan plot

Rotate plot in 3-D

Activate data curser

Insert colorbar

Insert legend