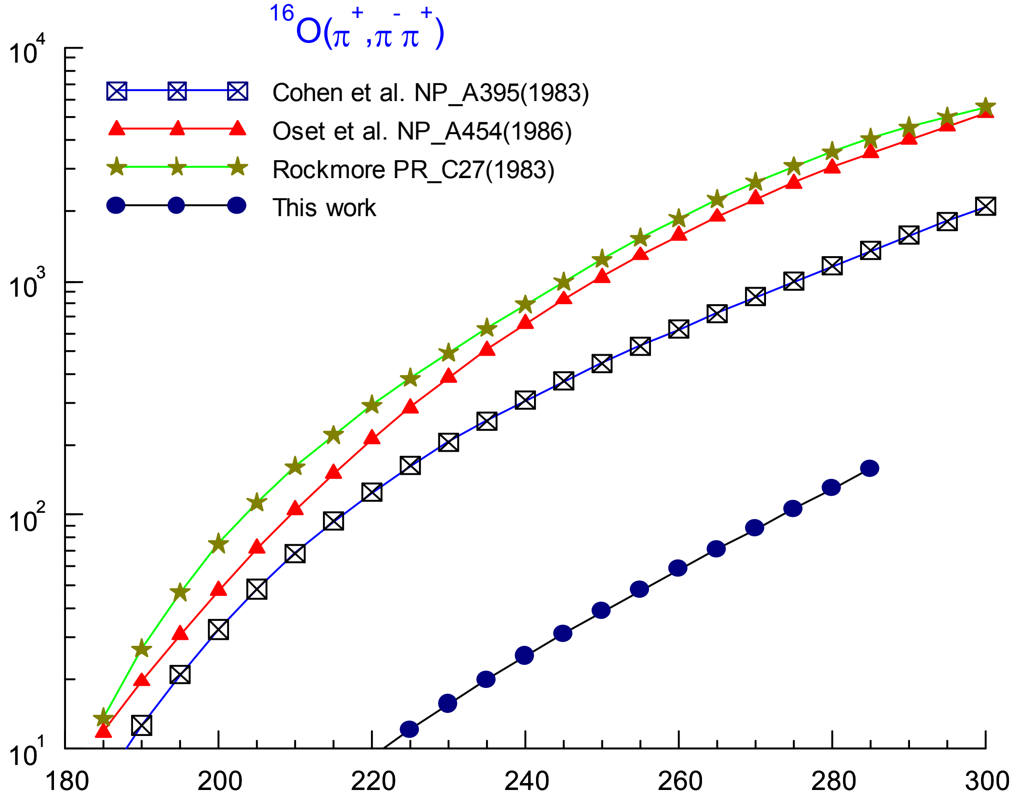
Extrema Tutorial

Customizing Graph Presentation



Introduction

Extrema has a large number of internal parameters used to control the drawing details. By altering these parameters you can vary the appearance of your drawing in a great variety of ways.

The most commonly used parameters can be easily set from the GUI, simply by checking off the desired options from those that are presented. The more obscure parameters may not have any convenient checkboxes, however, and will have to be set manually using a typed command.

Each drawing parameter has a name. To get the value of a parameter, use the function

```
GET characteristicname
```

This returns a value that can be viewed interactively, or stored in a variable. To set the value of a parameter, use

```
SET characteristicname value
```

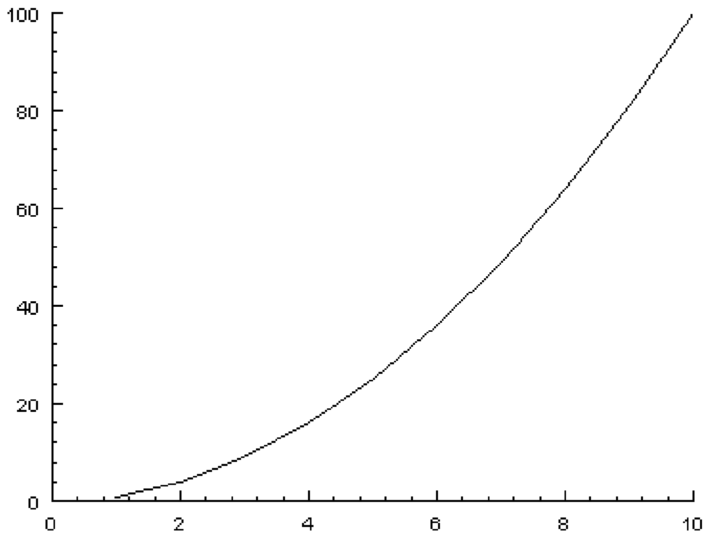
Many drawing parameters refer to positions on the drawing, which can be expressed in various units, including percentages. To interactively determine which position you would prefer, simply move your mouse over the drawing in the visualization window and the positions will be displayed below in whatever units have been selected.

Plot symbols

The graph below shows the result of using the default values for all the characteristics. The commands to produce this graph are on the left below. The first line creates a vector `X` with values `{1,2,3,…,10}` and the second line draws the graph.

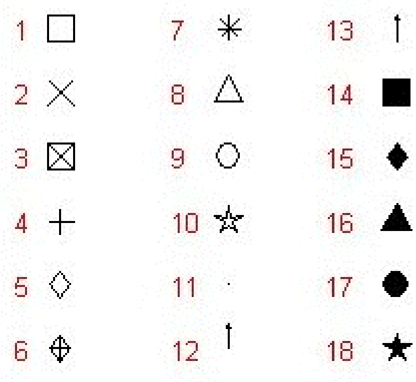
Note If you do not provide an independent variable to graph against, **Extrema** will use the vector index as the independent variable.

```
X = [1:10]
GRAPH X^2
```



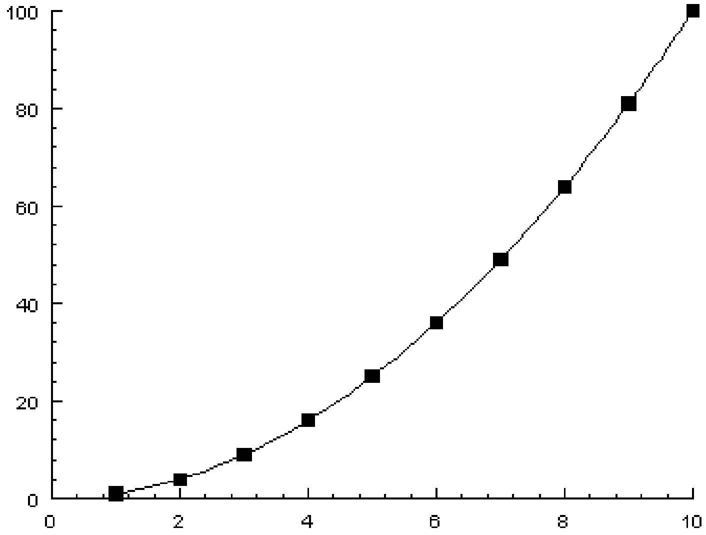
The plotting symbol can be manually selected in the `GRAPH` window. In the command language, use: SET PLOTSYMBOL n where `n` is the symbol number.

- If `n` is positive, successive points are connected by lines.
- If `n` is negative, the absolute value is used, but points will not be connected.
- If `n` is zero, no plotting symbol is used (the data is drawn as a simple curve).



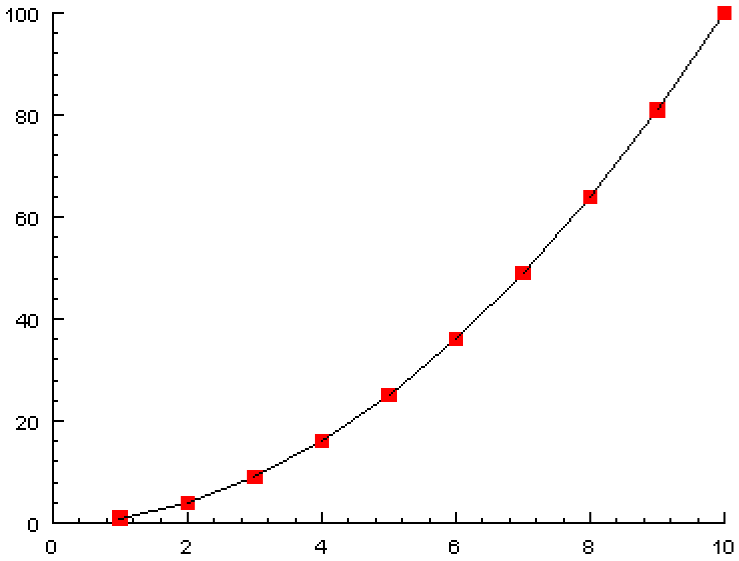
The default value of the PLOTSYMBOL characteristic is zero, which means to have no plot symbols. If you would prefer to have plot symbols drawn at the data locations, this is easily done by setting the PLOTSYMBOL characteristic, as in the example below.

```
X = [1:10]
SET PLOTSYMBOL 14
GRAPH X^2
```



If you would like the plot symbols to be red, use the PLOTSYMBOLCOLOR characteristic.

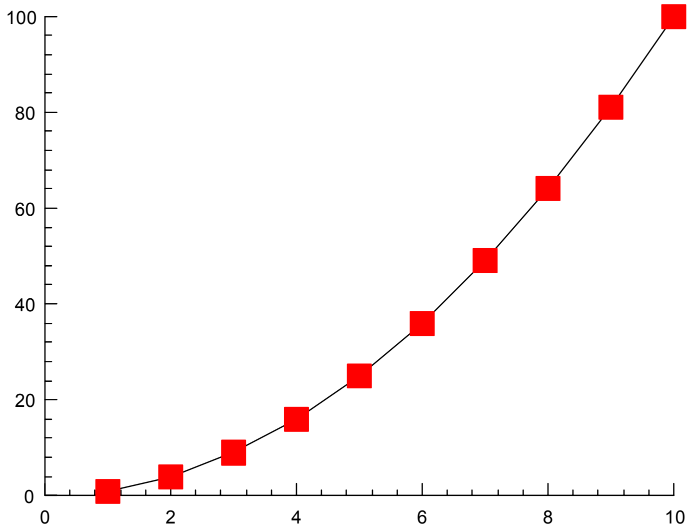
```
X = [1:10]
SET PLOTSYMBOL 14
SET PLOTSYMBOLCOLOR RED
GRAPH X^2
```



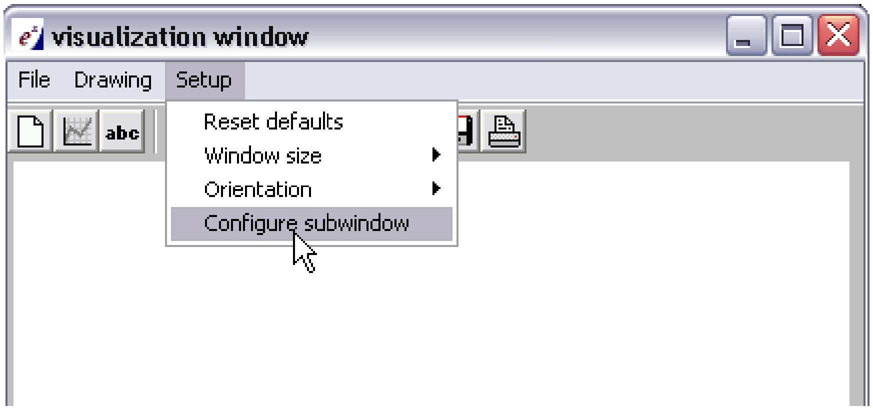
Note There are two groups of colors: a standard set of colors that are always defined, and a dynamically loaded colormap that can be changed at will. Dynamically loaded colors use color indexes starting at 1 and increasing to the number of colors in the colormap. Standard colors can be specified by name or by index number. The standard color indices start at 0 and decrease to –22 (since there are 23 standard colors).

If you want the plot symbols to be larger, change the value of the `PLOTSYMBOLSIZE` characteristic, as in the example below.

```
X = [1:10]
SET PLOTSYMBOL 14
SET PLOTSYMBOLCOLOR RED
SET %PLOTSYMBOLSIZE 4
GRAPH X^2
```



Note Characteristics which represent sizes or heights can be specified in either world units (inches) or as percentages of the height of the current graphics sub-window. Graphics sub-windows are chosen with the `WINDOW` command or with the SETUP ↪Configure subwindow menu item on the visualization window.



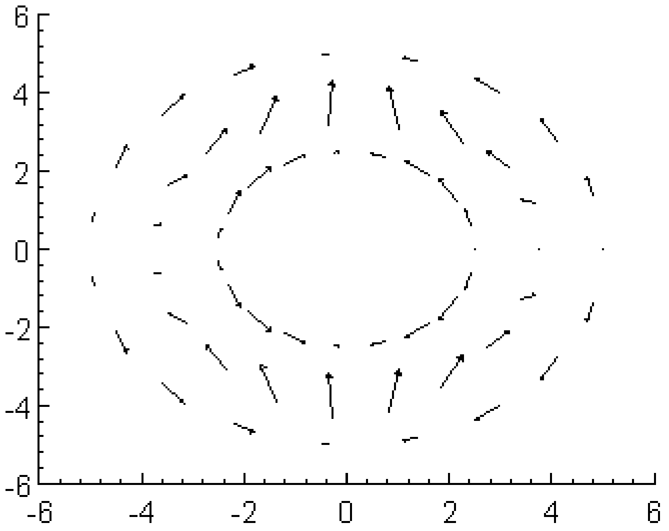
In addition to the plotting symbol code, size, and color, you can also specify the angle (in degrees). If scalar values are used for these characteristics, the value will apply to every data point. If vector values are used, the corresponding values for each point are used to set the plotting style for that point. The vectors should be the length as the data vectors.

In the GUI, you can simply enter the size, color, and angle in the appropriate fields. In the command language, use:

```
SET PLOTSYMBOL symbol
SET %PLOTSYMBOLSIZE size
SET PLOTSYMBOLCOLOR color
SET PLOTSYMBOLANGLE angle
```

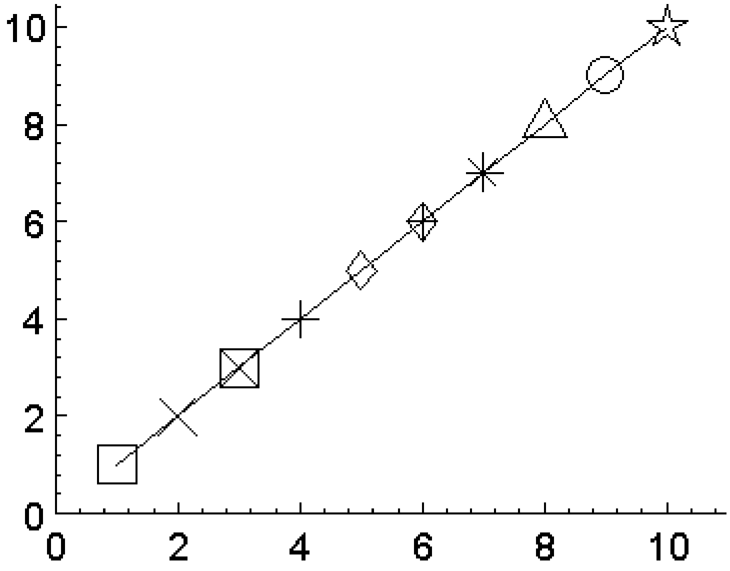
For example, to plot a vector field, we could select an arrow symbol where the arrow is centred at on the data value (#13), and then set the sizes and angles according to two vectors, `magnitude` and `direction`:

```
SET PLOTSYMBOL -13
SET %PLOTSYMBOLSIZE magnitude
SET PLOTSYMBOLCOLOR black
SET PLOTSYMBOLANGLE direction
GRAPH x y                               ! draw the vector field
```



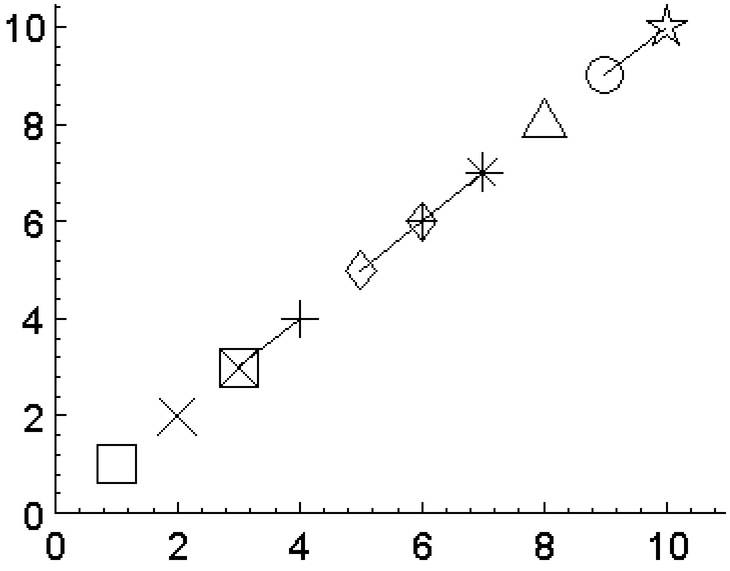
It is also possible to have each data point marked with a different plot symbol. This is accomplished by entering a vector instead of a scalar for the plot symbol. See the example below.

```
SET PLOTSYMBOL [1:10]
SET %PLOTSYMBOLSIZE 6
SET %XNUMBERHEIGHT 6
SET %YNUMBERHEIGHT 6
SCALES 0 11 0 10.5
GRAPH [1:10]
```



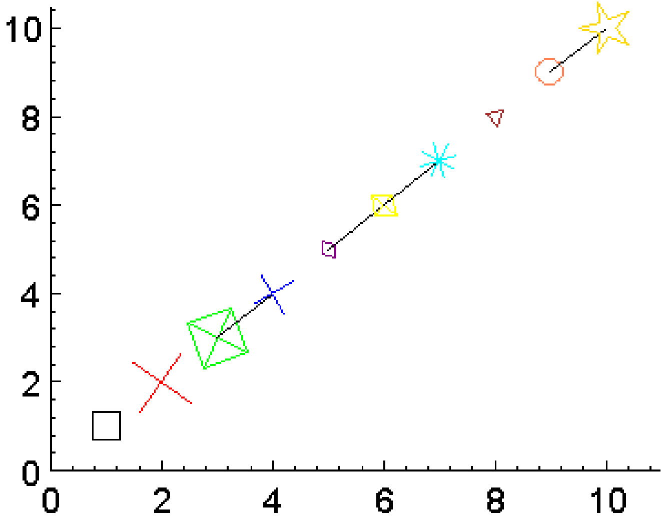
Sometimes you might want to have disconnected plot symbols. This is easily accomplished by using a negative value for the plot symbol. You can have a single negative scalar, in which case every symbol is the same, and they are all disconnected. It is also possible to specify different symbols by using a vector, as above. Any symbols with negative values will not be connected to the previous plot symbol. See the example below.

```
PC= [1;-2;-3;4;-5;6;7;-8;-9;10]
SET PLOTSYMBOL PC
SET %PLOTSYMBOLSIZE 6
SET %XNUMBERHEIGHT 6
SET %YNUMBERHEIGHT 6
SCALE 0 11 0 10.5
GRAPH [1:10]
```



You can also control the color and rotation angle of the plot symbols by changing the values for `PLOTSYMBOLCOLOR` and `PLOTSYMBOLANGLE.` These characteristics can be set as scalars or vectors, giving you complete control over each individual plot symbol. The following example puts it all together.

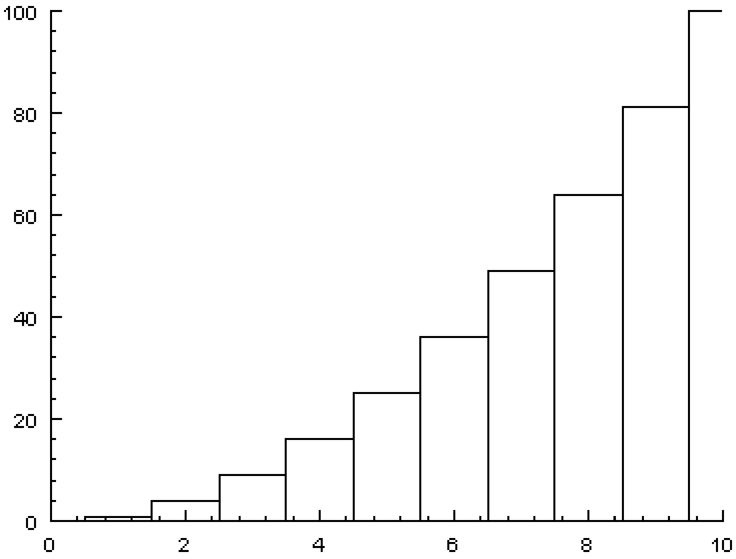
```
PS=[1;-2;-3;4;-5;6;7;-8;-9;10]
GENERATE\RANDOM PSS 2 10 10
PSC=-[1:10]
GENERATE PSA 0,,90 10
SET PLOTSYMBOL PS
SET %PLOTSYMBOSIZE PSS
SET PLOTSYMBOLCOLOR PSC
SET PLOTSYMBOLANGLE PSA
SET %XNUMBERHEIGHT 6
SET %YNUMBERHEIGHT 6
SCALE 0 11 0 10.5
GRAPH [1:10]
```



Histograms

There are four types of 1D histogram available via the HISTOGRAMTYPE characteristic. To draw the standard type of histogram, i.e., horizontal with tails going to *y=0*, the simplest way is to use the \HISTOGRAM qualifier on the GRAPH command.

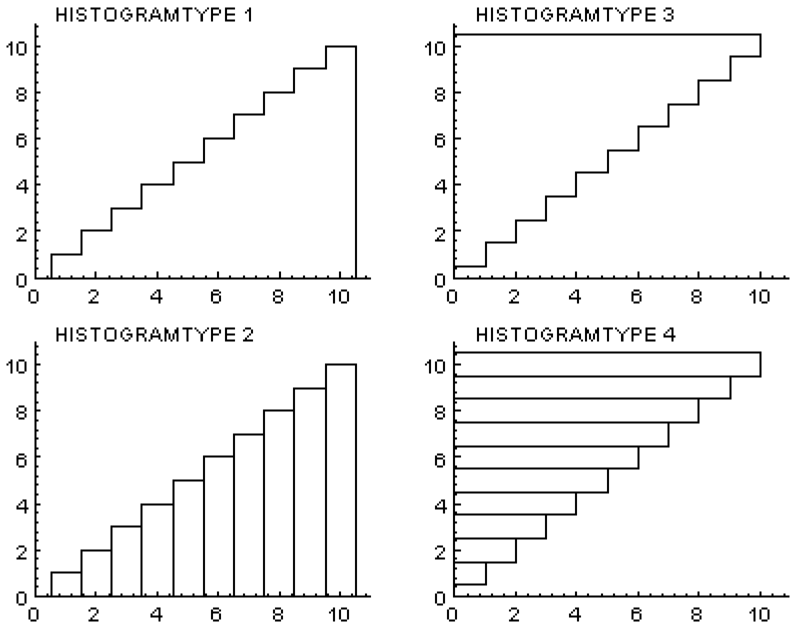
```
X = [1:10]
GRAPH\HISTOGRAM X^2
```



An equivalent way to draw the same graph:

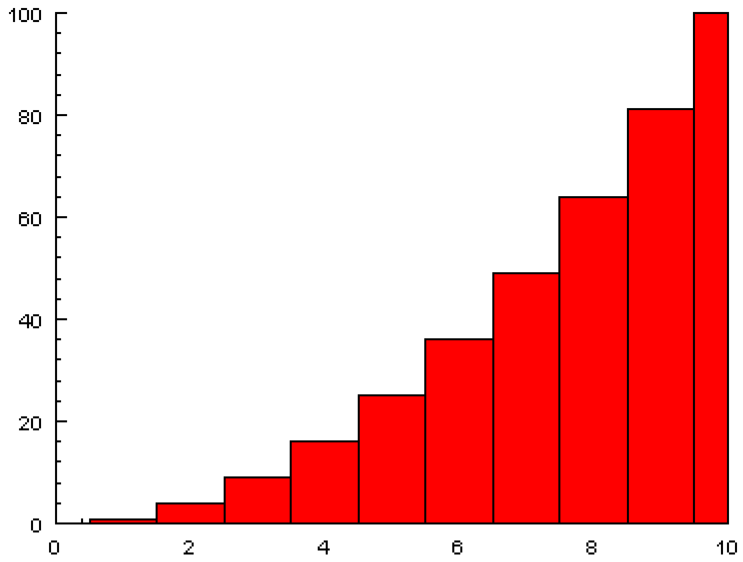
```
X = [1:10]
SET HISTOGRAMTYPE 2
GRAPH X^2
```

The other types of histogram are shown below, each with the appropriate value of HISTOGRAMTYPE.



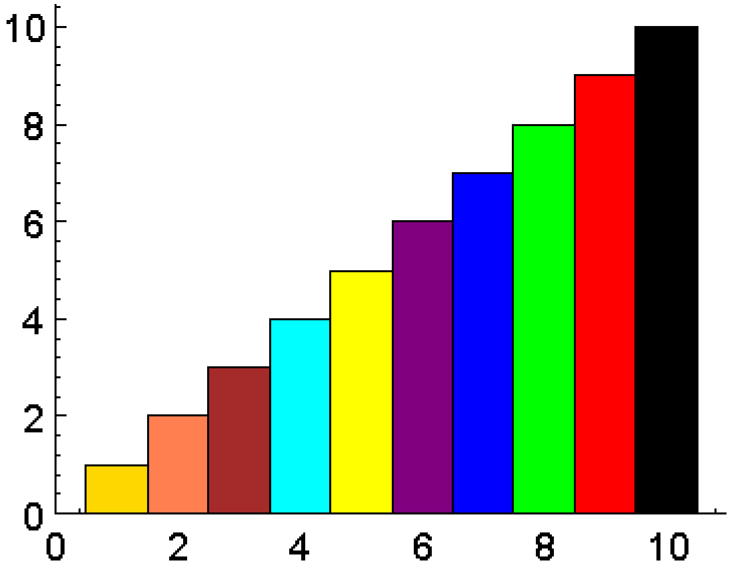
If you want color solid filled histogram bars, use the SET AREAFILLCOLOR command, as in the following set of commands which produce the figure on the right.

```
X = [1:10]
SET AREAFILLCOLOR RED
GRAPH\HISTOGRAM X^2
```



You can also fill the individual bars of types 2 and 4 histograms with different colors by setting the AREAFILLCOLOR characteristic to a vector instead of a scalar.. See the example below.

```
SET AREAFILLCOLOR [-10:-1:1]
SET %XNUMBERHEIGHT 6
SET %YNUMBERHEIGHT 6
SCALES 0 11 0 10.5
GRAPH\HISTOGRAM [1:10]
```

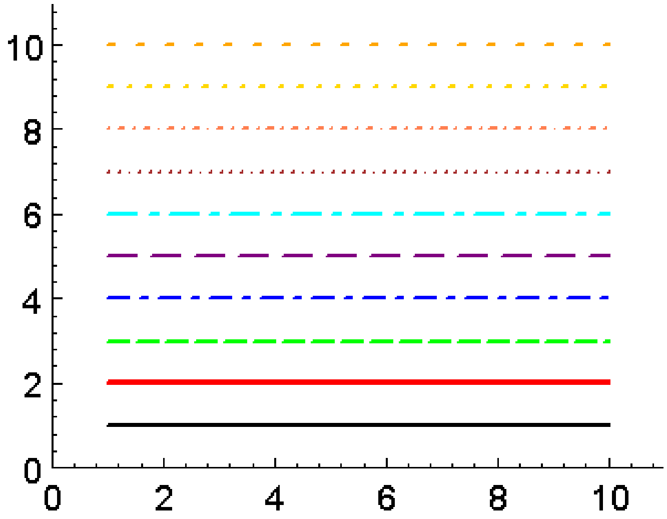


Data curves

We have seen that if the plot symbol drawn at a data point is non-negative, it will be connected to the previous data point by a line segment. The collection of line segments connecting data points is the **data curve**. It is possible to control the color, line type, and line width for the data curve using the CURVECOLOR, CURVELINETYPE, and CURVELINEWIDTH characteristics. The example below shows a script which changes the color, line type, and line width for ten data curves.

```
X[1:10]=1
COLORS=[1;2;3;4;5;7;8;9;10;11]
SET
 PLOTSYMBOL 0
 CURVELINEWIDTH 5

SCALES 0 11 0 11
GRAPH\AXESONLY
DO I = [1:10]
 SET CURVECOLOR -COLORS[I]
 SET CURVELINETYPE I
 GRAPH\OVERLAY X+I-1
ENDDO
```

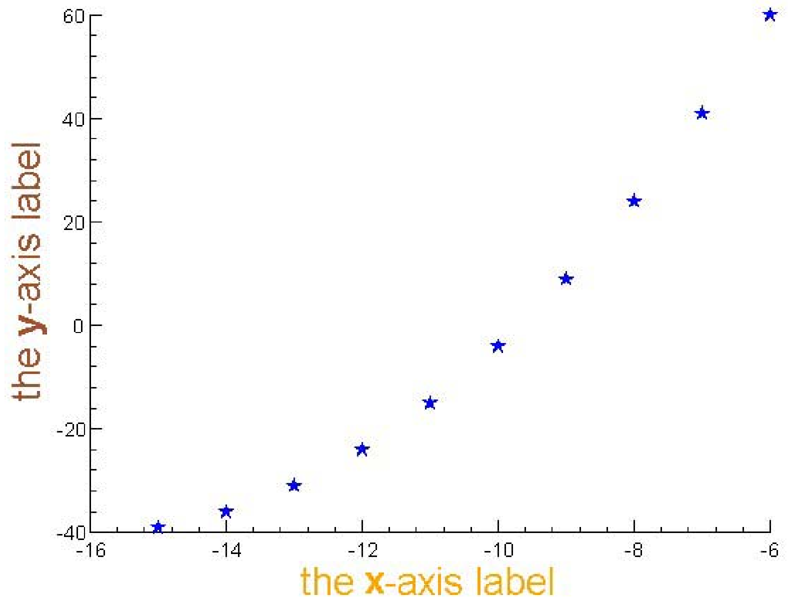


Axis labels

Axis labels are a special case of text strings, since they have a standard placement and orientation. The *x*-axis text label is drawn, centred, below the *x*-axis. The *y*-axis text label is drawn, centred, to the left of the *y*-axis. The axis text labels are drawn only when the axes are drawn. The character string may contain format commands.

The `SET XLABEL` command sets the automatic *x*-axis text label. Use the `SET XLABELON` command to toggle off/on drawing the *x*-axis text label. Change the sizes of the text label with `SET XLABELHEIGHT` or `SET %XLABELHEIGHT`. Change the font of the *x*-axis text label with the `SET XLABELFONT` command and change the color of the *x*-axis text label with the `SET XLABELCOLOR` command.

The `SET YLABEL` command sets the automatic *y*-axis text label. Use the `SET YLABELON` command to toggle off/on drawing the *y*-axis text label. Change the sizes of the text label with `SET YLABELHEIGHT` or `SET %YLABELHEIGHT`. Change the font of the *y*-axis text label with the `SET YLABELFONT` command and change the color of the *y*-axis text label with the `SET YLABELCOLOR` command.



Graph legend

Legends are boxes of descriptive text that describe certain details of the graph. Typically, they are used to label different point types, different line types or colors, contour elevations, fit parameters, and so on.

The `LEGEND` characteristic is changed with the `SET` command and the current value is obtained with the `GET` command. If `LEGEND` ≠ 0, a legend entry is drawn into a legend frame box. A legend entry consists of a short line segment, with optional plotting symbol(s), and a text string. The legend entry is drawn when the `GRAPH` command is entered. The string portion of the legend entry is expected as the first parameter of the `GRAPH` command, for example:

```
GRAPH 'legend entry' x y
```

Note	If `LEGEND` = 0, a string entered as a first parameter with the `GRAPH` command is ignored.

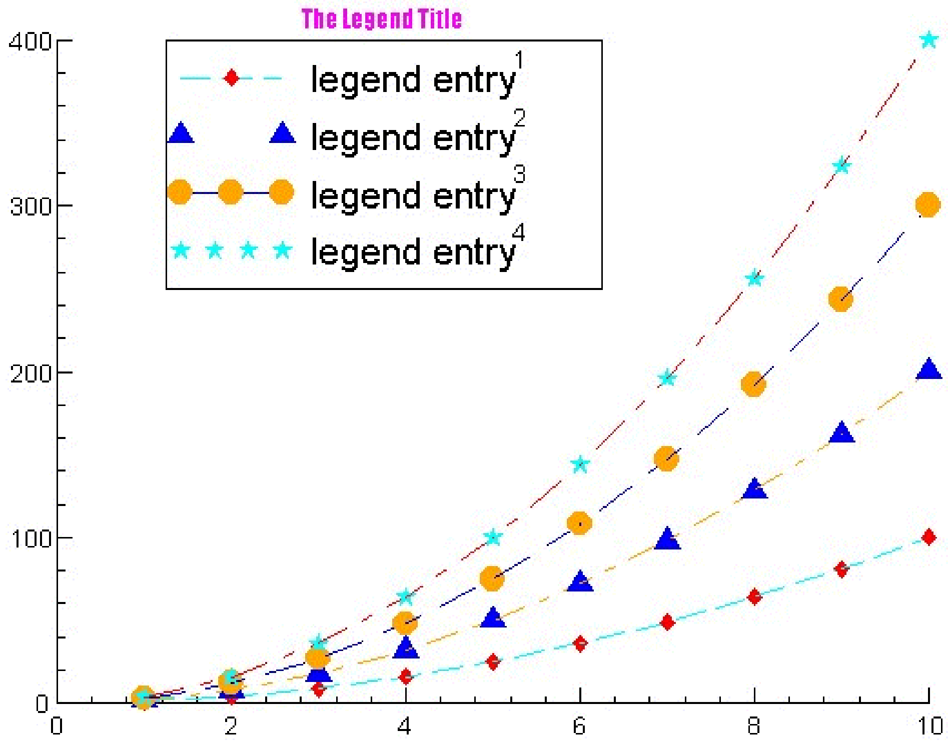
Following is an example script using a graph legend and the picture that it produces.

```
X=[1:10]
SET
 LEGEND 1
 LEGENDTITLECOLOR -16
 LEGENDTITLEFONT 'IMPACT'
 LEGENDTITLE 'THE LEGEND TITLE'
 LEGENDFRAME 20 60 60 90

LINECODES = [1;0;1;0]
SYMBOLSIZES = [2;2.5;3;2]
```

```
SYMBOLS = [15;16;17;18]
COLORS[1] = 'RED'
COLORS[2] = 'BLUE'
COLORS[3] = 'ORANGE'
COLORS[4] = 'CYAN'
WIDTHS = [1;1;1;1]
DO I = [1:4]
  SET
   PLOTSYMBOL SYMBOLS[I]
   PLOTSYMBOLCOLOR COLORS[I]
   %PLOTSYMBOLSIZE SYMBOLSIZES[I]
   LEGENDSYMBOLS I
   LEGENDENTRYLINE LINECODES[I]
   CURVECOLOR COLORS[5-I]
   CURVELINETYPE I+2
   CURVELINEWIDTH WIDTHS[I]

  GRAPH 'LEGEND ENTRY<^>'//RCHAR(I) X I*X^2
ENDDO
REPLOT
```



Graph axes

To graph only the axes for a particular set of data, use:

```
GRAPH\AXESONLY x y
```

To graph a set of data with no axes, use:

```
GRAPH\OVERLAY x y
```

These options are handy if you make multiple drawing passes over the same graph. In the GUI you can simply select the appropriate checkboxes to get the same behaviour.

Scaling

Axes can be manually or automatically scaled.

Auto-scaling is the default, in which the axis will stretch or shrink to accommodate the full range of the plotted data. This is convenient for well-behaved data sets, but maybe not for data with spikes, infinities, or related problems. Autoscaling is also inconvenient when one is overlaying numerous similar graphs, where one requires that the scale be fixed.

Manual axis scaling is done using the `SCALES` command:

```
SCALES xmin xmax ymin ymax
SCALES xmin xmax nxtics ymin ymax nytics
SCALES
```

The first form simply sets axis ranges. The second form also sets the number of large (numbered) tic marks that should be shown for each axis. The last form freezes the axis scales at whatever is their current value.

Tic marks

The parameters controlling *x*-axis tic marks are:

`XTICSON`	controls whether or not tic marks, both large and small, are drawn on the *x*-axis.
`XTICSBOTHSIDES`	controls whether or not tic marks, both large and small, are drawn on both sides of the *x*-axis.
`XTICANGLE`	controls the angle of the tic marks, both large and small, on the *x*-axis.
`XNLINCS`	controls the number of large, labelled, tic marks to be displayed on the *x*-axis
`XNSINCS`	controls the number of small, unlabeled, tic marks to be displayed between the large, labelled, tic marks on the *x*-axis.

XLARGETICLENGTH	controls the length of the large, labelled, tic marks on the *x*-axis.
XSMALLTICLENGTH	controls the length of the optional small tic marks on the *x*-axis. These are the unlabeled tic marks between the large, numbered, tic marks.
XIMAGTICANGLE	controls the angle, in degrees, measured counter clockwise, between the *x*-axis and a line joining the base of each large tic mark on the *x*-axis to the centre of the number labelling that tic mark.
XIMAGTICLENGTH	controls the distance, measured from the base of each large tic mark on the *x*-axis, to the centre of the number labelling that tic mark

The parameters controlling *y*-axis tic marks are:

YTICSON	controls whether or not tic marks, both large and small, are drawn on the *y*-axis.
YTICSBOTHSIDES	controls whether or not tic marks, both large and small, are drawn on both sides of the *y*-axis.
YTICANGLE	controls the angle of the tic marks, both large and small, on the *y*-axis.
YNLINCS	controls the number of large, labelled, tic marks to be displayed on the *y*-axis
YNSINCS	controls the number of small, unlabeled, tic marks to be displayed between the large, labelled, tic marks on the *y* -axis.
YLARGETICLENGTH	controls the length of the large, labelled, tic marks on the *y* -axis.
YSMALLTICLENGTH	controls the length of the optional small tic marks on the *y* -axis. These are the unlabeled tic marks between the large, numbered, tic marks.
YIMAGTICANGLE	controls the angle, in degrees, measured counter clockwise, between the *y* -axis and a line joining the base of each large tic mark on the *y*-axis to the centre of the number labelling that tic mark.

`YIMAGTICLENGTH`	controls the distance, measured from the base of each large tic mark on the *y*-axis, to the centre of the number labelling that tic mark

Logarithmic axes

To get logarithmic scaling on the *x*-axis, use `SET XLOGBASE n`, where:

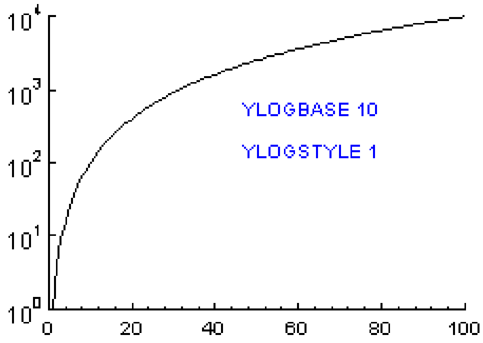
`n` > 1.0	the *x*-axis will have a logarithmic scale. The base will be the integer part of `XLOGBASE`, except for the special case: $1.05*e >$ `XLOGBASE` $> 0.95*e$, where e is the base of the natural logarithms, $e \approx 2.718281828$, in which case the base will be e.
`n` ≤ 1.0	the *x*-axis will have a linear scale

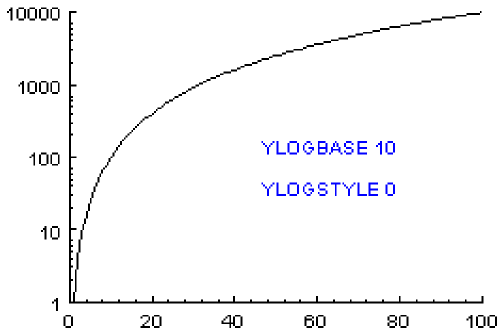
If `XLOGSTYLE` $= 0$, and `XLOGBASE` > 1.0, then the numbers labelling the large tic marks on the *x*-axis are displayed in decimal format. If `XLOGSTYLE` $\neq 0$, and `XLOGBASE` > 1.0, then the numbers labelling the large tic marks on the *x*-axis are displayed in exponential format.

To get logarithmic scaling on the *y*-axis, use `SET YLOGBASE n`, where:

`n` > 1.0	the *y*-axis will have a logarithmic scale. The base will be the integer part of `YLOGBASE`, except for the special case: $1.05*e >$ `YLOGBASE` $> 0.95*e$, where e is the base of the natural logarithms, $e \approx 2.718281828$, in which case the base will be e.
`n` ≤ 1.0	the *y*-axis will have a linear scale

If `YLOGSTYLE` $= 0$, and `YLOGBASE` > 1.0, then the numbers labelling the large tic marks on the *y*-axis are displayed in decimal format. If `YLOGSTYLE` $\neq 0$, and `YLOGBASE` > 1.0, then the numbers labelling the large tic marks on the *y*-axis are displayed in exponential format.





Axis placement

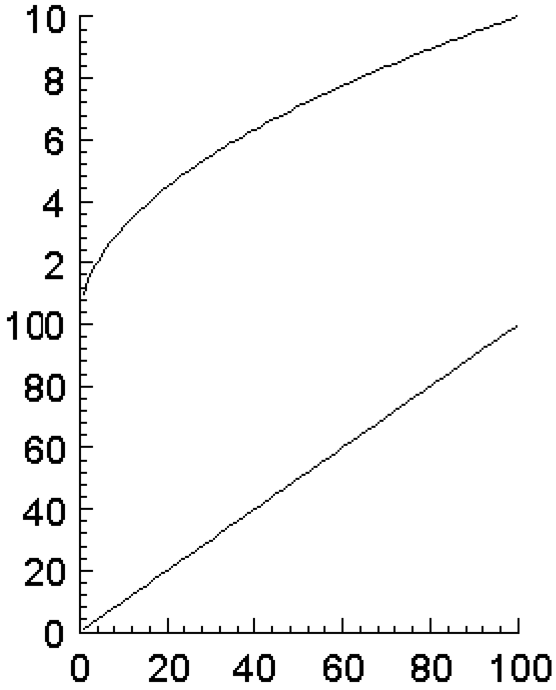
The placement of the axes can be precisely controlled by manipulating the axis location parameters:

```
XLOWERAXIS, XUPPERAXIS, YLOWERAXIS, YUPPERAXIS
```

The percentage versions specify positions as percentages of the current drawing window; otherwise the positions are in the drawing coordinates.

By careful manipulation of these values, you can place one graph at any point on the drawing with respect to another. For instance, to adjoin two graphs along the *x*-axis so that there is an upper graph and a lower graph with a common edge:

1. Set `%YUPPERAXIS` to a reduced value, e.g., 50.
2. Plot the first graph.
3. Set `%YLOWERAXIS` to the value of `%YUPPERAXIS`
4. Set `%YUPPERAXIS` to 90.
5. Turn off drawing of the *x*-axis labels with `SET XAXIS 0`.
6. Plot the second graph.



In practice, there are some other parameters you may need to play with to keep the *y*-axis labelling clean, but the above will suffice in simple cases.

Axis characteristics

There are many characteristics of a graph's *x*- and *y*-axes which can be controlled by the user. These include such properties as the color of each axis, the color of the numbers on each axis, the location and length of each axis, the number of tic marks (both major and minor), the length and angle of the tic marks, and so on.

SUMMARY

Here are the characteristics you can change to achieve the effects you desire in your drawings. These may be looked up in the online help or in the Extrema Command Reference for more details on their use.

Plot symbol characteristics

PLOTSYMBOL, PLOTSYMBOLSIZE, PLOTSYMBOLANGLE, PLOTSYMBOLLINEWIDTH, PLOTSYMBOLCOLOR

Data curve characteristics

HISTOGRAMTYPE, CURVECOLOR, CURVELINETYPE, CURVELINEWIDTH

General characteristics

NHISTORY, TENSION, AUTOSCALE, COLOR, COLORMAP, COLORMAPFILE, COLORMAPNAME, COLORMAPSIZE, AREAFILLCOLOR, GRAPHBOX, ORIENTATION, WINDOWSIZE, LINETYPE, LINEWIDTH, FONT, CONTOURLABELHEIGHT, CONTOURLABELSEPARATION

x-axis characteristics

XAXISCOLOR, XLABELFONT, XLABELCOLOR, XNUMBERSFONT, XNUMBERSCOLOR, XLOWERAXIS, XUPPERAXIS, XNUMBERS, XNUMBEROFDIGITS, XNUMBEROFDECIMALS, XNUMBERHEIGHT, XIMAGTICLENGTH, XIMAGTICANGLE, XNUMBERANGLE, XPOWER, XPOWERAUTO, XLABEL, XLABELHEIGHT, XLABELON, XLARGETICLENGTH, XSMALLTICLENGTH, XTICANGLE, XFORCECROSS, XMIN, XMAX, XVIRTUALMIN, XVIRTUALMAX, XNLINCS, XNSINCS, XTICSON, XTICSBOTHSIDES, XAXIS, XGRID, XAXISANGLE, XLOGBASE, XLOGSTYLE, XZERO, XMOD, XLEADINGZEROS, XOFFSET, XDROPFIRSTNUMBER, XDROPLASTNUMBER

y-axis characteristics

YAXISCOLOR, YLABELFONT, YLABELCOLOR, YNUMBERSFONT, YNUMBERSCOLOR, YLOWERAXIS, YUPPERAXIS, YNUMBERS, YNUMBEROFDIGITS, YNUMBEROFDECIMALS, YNUMBERHEIGHT, YIMAGTICLENGTH, YIMAGTICANGLE, YNUMBERANGLE, YPOWER, YPOWERAUTO, YLABEL, YLABELHEIGHT, YLABELON, YLARGETICLENGTH, YSMALLTICLENGTH, YTICANGLE, YFORCECROSS, YMIN, YMAX, YVIRTUALMIN, YVIRTUALMAX, YNLINCS, YNSINCS, YTICSON, YTICSBOTHSIDES, YAXIS, YGRID, YAXISANGLE, YLOGBASE, YLOGSTYLE, YZERO, YMOD, YLEADINGZEROS, YOFFSET, YDROPFIRSTNUMBER, YDROPLASTNUMBER

Text characteristics

TEXTFONT, TEXTCOLOR, TEXTINTERACTIVE, TEXTALIGN, TEXTHEIGHT, TEXTANGLE, XTEXTLOCATION, YTEXTLOCATION

Graph legend characteristics

LEGEND, LEGENDUNITS, LEGENDENTRYLINE, LEGENDFRAMEON, LEGENDFRAME, LEGENDTRANSPARENCY, LEGENDSYMBOLS, LEGENDAUTOHEIGHT, LEGENDTITLEON, LEGENDTITLE, LEGENDTITLEHEIGHT, LEGENDTITLECOLOR, LEGENDTITLEFONT

File related characteristics

EXTENSION, COMMENT, ERRORFILL